

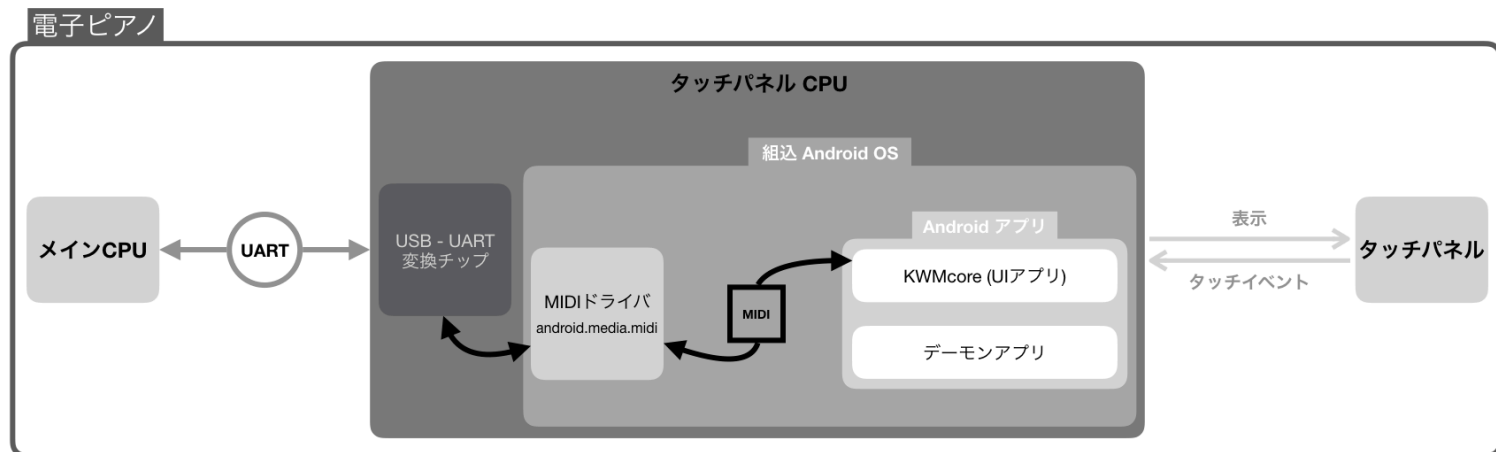
電子ピアノ用 組込AndroidOSシステム 要求仕様書

2019.2.4

河合楽器製作所
電子楽器事業部 開発課
新技術開発グループ

電子ピアノ用のタッチパネルを制御する組込AndroidOS基板に弊社が別途開発したAndroidアプリ「KWMcore」を導入し、アプリを電子ピアノ用のユーザーインターフェースとする。その為に必要なOSカスタマイズとドライバ、ユーティリティアプリの開発を行う。

システム概略図



開発委託内容

① 組込AndroidOSセットアップ

- ・ AndroidOSのバージョンを6.0にする
- ・ LCDやUSB端子、USB-UART変換チップ等のI/Oのポーティングを行う
- ・ Android標準MIDIドライバ android.media.midiを導入する
- ・ システムのWebViewをversion71以降にアップデートする

以上のセットアップを行い、UIアプリ「KWMcore」から電子ピアノメインCPUとタッチパネルLCDの制御を行えるOS環境を構築する。

※UART通信の要求仕様については「UART通信 要求仕様」ページを参照

② 組込AndroidOS モディファイ

- ・ UIアプリ「KWMcore」から制御ができる事
- ・ 動作パフォーマンス向上
- ・ 電源投入時からの起動時間の極力短縮
- ・ エンドユーザーが本製品を使用してAndroidOSベースと認知されないようにする事

を目的とし、OSのモディファイを行う。

詳細は「OSモディファイ要件」ページと別紙「制御仕様(案)」を参照

③ デーモンアプリ

UIアプリ「KWMcore」を監視し、クラッシュした際に自動的に再起動を行うアプリを開発する。

アプリ対応

- ・ WebViewを最新に
- ・ apkからファイルストレージアクセス（パーミッションも含む）
- ・ アプリ制御用インターフェース(JNI)実装
（別紙「電子ピアノ用 組込AndroidOSシステム 制御仕様」参照）

タッチパネルLCD対応

- ・ 画面解像度の変更
- ・ 画面表示方向の固定

エンドユーザー向け最適化

- ・ OS起動時間短縮 [OS立ち上がりまで10秒]
- ・ 起動画面アニメーションの変更
- ・ ロック画面機能の削除
- ・ ナビゲーションバーの削除
- ・ 画面下部（ホームボタン、戻るボタン等）の削除
- ・ OSレベル通知の廃止
- ・ キーボード変更（現在検討中のため、別途ご案内）
- ・ UIアプリ「KWMcore」の埋め込み&自動起動
- ・ デモンアプリによるUIアプリ「KWMcore」の自動再起動

パフォーマンス向上のための不要要素削除

- ・ uboot最適化
- ・ カーネル最適化（不要ドライバの削除）
- ・ 余分なapkファイルの削除
- ・ スタートアップサービスのカット
- ・ GPU有効化
- ・ UIアプリ向けヒープサイズの設定

動作検証用

- ・ Android標準の設定アプリを起動できるようにする
- ・ 設定アプリ内の開発者モードを有効にし、設定を保持させる

・ android.media.midiの定義

android.media.midiが導入されていない組込Androidデバイスの場合、導入を行う。

android.media.midi及びMIDI通信機能の適合の定義は以下、Google Androidの公式ドキュメントに従う。

Android 6.0 Compatibility Definition Document

https://source.android.com/compatibility/6.0/android-6.0-cdd#5_9_midi

MIDI Claim Feature

<https://source.android.com/devices/audio/midi.html#claim-feature>

・ android.media.midi適合テスト

android.media.midiの適合確認のテストは以下、Google Androidの公式ドキュメントに従う。

MIDI Test Procedure

https://source.android.com/devices/audio/midi_test

・ 参考情報

参考情報として、android.media.midiの公式ドキュメントを以下に記す。

android.media.midi

<https://developer.android.com/reference/android/media/midi/package-summary>

・通信プロトコル

MIDI1.0規格 (<http://amei.or.jp/midistandardcommittee/MIDIspcj.html>)

・通信ボーレート

115.20kbps / 31.25kbps

許容誤差 : $\pm 1\%$

・要求仕様詳細

MIDI送受信のメソッドに応じて、通信スピード優先の「パラメーター通信」、時間精度優先の「楽曲再生通信」の2種類の要求が存在する。

① パラメーター通信

対象メソッド : MidiReceiverクラス

- ・ send(byte[] msg, int offset, int count)
- ・ onSend(byte[] msg, int offset, int count, long timestamp)

最大送信データ量 : 4096byte

許容レイテンシー : 10msec以下

② 楽曲再生通信

対象メソッド : MidiReceiverクラス

- ・ send(byte[] msg, int offset, int count, long timestamp)

ストリーミング通信 : バッファを設けて、時間精度の高い通信をする

最大送信データ量 : 楽曲の長さにより可変

バッファ時間 : 100msec以下

許容時間ズレ : 1msec以下

・テスト

弊社提供テストアプリにて、測定・テストを実施

①UART入力端子を接続

②テストアプリを起動し、スタートボタンを押す

③アプリに測定結果が表示されるので上記要求仕様を満たしているか確認する

・ソースコード

OSソースファイル

JNIを含むApplication Frameworkソースコード一式

デーモンアプリソースファイル (起動用 initスクリプトを含む)

・ドキュメント

設計書

制御仕様書

- ハードウェア制御仕様書

- JNI制御仕様書

テスト結果報告書

- Androidカスタマイズテスト

- android.media.midi適合テスト

- UART通信テスト (弊社テストアプリ利用)

- デーモンアプリテスト

・ビルド情報

OS・ApplicationFramework・JNI・デーモンアプリの開発時ビルド情報

- ビルド手順書

- ビルド環境構築手順書

- ビルドに必要なPCのスペック情報

任意のAPKファイルを含め、イメージファイルを構築するためのビルド情報

- ビルド手順書

- ビルド環境構築手順書

ビルドマシン (※日本語OS、費用は別途見積もり)

- 上記のビルド環境を再現したビルドマシン一式を納入頂きます

※提供いただく日本語ドキュメントについて翻訳品質が低く弊社で理解出来ない場合は、中国語原版の仕様書や英語による回答を要求します。

ファイル共有について

ビルドマシンを除く全ての納品物はGithubでバージョン管理を行い、GitHubを通じてファイル共有、納品をお願いします。別途GitHubのプライベートリポジトリを作成、アクセス方法をご連絡します。開発中は逐次コミットしていただき、バージョン管理とともに開発状況の把握にも努めてください。前回開発時の失敗・反省点を踏まえた方針のため、ご協力をお願いします。

改變歷

[illegible]