OpenEL®仕様書

1.0-1 版

2014年 03月01日

改版履歴

日付	版数	変更箇所	内容
2012 年 04 月 01 日	0.1	-	ET ロボコン用プロトタイプ
2013 年 05 月 22 日	1.0	-	1.0 版策定
2014 年 03 月 01 日	1.0-1	-	図表番号追記 モータ状態遷移図更新

目次

1. はじめに

本書は、一般社団法人組込みシステム技術協会(以下 JASA という)が提唱する、OpenEL（Open Embedded Library）1.0 版の仕様を説明するものである。

本書の目的は、デバイスを作成するメーカ、デバイスを使用する開発者及びソフトウェア利用者が、OpenEL 仕様に準拠したインタフェースを構築するために、OpenEL 対応デバイスの API 仕様を規定することである。

本書は以下の読者を対象とする。

- OpenEL を用いて、ミドルウェアやソフトウェアを開発したり利用したりするソフトウェア技術者
- OpenEL に準拠したデバイスとコンポーネントを開発し供給するデバイスベンダ及びその技術者
- ロボット及び組込みソフトウェアの開発に興味のある技術者

2. ライセンス

本仕様書の著作権は JASA に帰属する。

OpenEL は、一般社団法人組込みシステム技術協会の日本における登録商標である。

OpenEL のロゴは、一般社団法人組込みシステム技術協会の商標である。

JASA は、本仕様書の全文または一部分を改変することなく複写し、無料または実費程度の費用で再配布することを許諾する。ただし、本仕様書の一部を抜粋して再配布する場合には、OpenEL 1.0 仕様書からの抜粋であること、抜粋した箇所、及び本仕様書の全文を入手する方法を明示(http://www.jasa.or.jp/ など)することを条件とする。

その他、本仕様ならびに本仕様書の利用条件の詳細については次節に記載する。

本仕様ならびに本仕様書に関する問合せ先は、巻末の「8.問合せ先」を参照のこと。

仕様の利用条件

OpenEL 1.0 仕様は、オープンな仕様である。誰でも自由に、OpenEL 1.0 仕様に準拠したソフトウェアを開発・使用・配布・販売することができる。

ただし、JASA は、OpenEL 1.0 仕様に準拠したソフトウェアの製品ドキュメントなどに、以下の文言を入れることを強く推奨する。

> OpenEL は、一般社団法人組込みシステム技術協会の日本における登録商標です。
> OpenEL のロゴは、一般社団法人組込みシステム技術協会の商標です。

また、OpenEL 1.0 仕様に準拠したソフトウェアの製品ドキュメントなどに、以下の文言を入れることを推奨する。

> OpenEL1.0 仕様は、JASA が定めたオープンな（組込みシステム API）仕様です。
> OpenEL1.0 仕様の仕様書は、JASA のホームページから入手することが出来ます。

仕様書を改変して製品ドキュメントを作成する許諾を受けた場合などは、これらの 2 つの推奨に従うことが条件となる。

仕様書の利用条件

OpenEL 1.0 仕様書の著作権は JASA に帰属する。

JASA は、OpenEL 1.0 仕様書の全文または一部分を改変することなく複写し、無料または実費程度の費用で再配布することを許諾する。ただし、OpenEL 1.0 仕様書の一部を再配布する場合には、OpenEL 1.0 仕様書からの抜粋である旨、抜粋した箇所、及び OpenEL 1.0 仕様書の

全文を入手する方法を明示しなければならない。

また、JASA は事前に書面による承諾が無い限り、OpenEL 1.0 仕様書を改変することを堅く禁止する。JASA は、JASA 会員に対してのみ、OpenEL 1.0 仕様書を改変して製品ドキュメントを作成し、それを配布・販売することを許諾している。許諾条件やその申請方法については、JASA に問い合わせること。

無保証

JASA は、OpenEL 1.0 仕様及び仕様書に関して、いかなる保障も行わない。また、OpenEL 1.0 仕様及び仕様書を利用したことによって、直接的または間接的に発生した如何なる損害も補償しない。

また、JASA は、OpenEL 1.0 仕様書を予告無く改定する場合がある。

【今後開始予定】OpenEL 仕様準拠品登録制度

JASA では、OpenEL 仕様の普及と発展を促進するため、OpenEL 仕様準拠製品登録制度を設ける予定である。

この制度は、各社で開発された OpenEL 仕様準拠の製品の一覧を作成・保守し、JASA の広報活動などを通じて、OpenEL 仕様ならびに準拠製品の普及を図ることを目的としたものである。なお、この制度は、いわゆる検定制度とは異なり、登録された製品が OpenEL 仕様に準拠していることを認証するためのものではない。

この制度に登録されている製品の一覧は、JASA のホームページ(http://www.jasa.or.jp/)で閲覧可能となる予定である。

この制度は開始の準備が整い次第、JASA のホームページで制度開始のアナウンスと利用・登録方法を一般に公開する。

3. 用語

表 3-1 用語一覧

No	用語	意味・用途
1	OpenEL	Open Embedded Library。制御システムのソフトウェアの実装仕様を標準化する組込みシステム向けのオープンなプラットフォーム。
3	JASA	一般社団法人 組込みシステム技術協会の略称。 組込みシステム（組込みソフトウェアを含めた組込みシステム技術をいう。以下同じ。）における応用技術に関する調査研究、標準化の推進、普及及び啓発等を行うことにより、組込みシステム技術の高度化及び効率化を図り、もって我が国の産業の健全な発展と国民生活の向上に寄与することを目的とする、業界団体である。OpenEL 仕様の策定・管理も本団体にて行っている。

4. OpenEL とは

OpenEL (Open Embedded Library)とは、ロボットや制御システムなどのソフトウェアの実装仕様を標準化する組込みシステム向けのオープンなプラットフォームである。具体的には、センサ入力やモータへの出力等、機器制御のための API(Application Program Interface)をミドルウェアよりも低いレイヤで標準化し、デバイスドライバ等のソフトウェアの移植性、再利用性、生産性を向上するための仕組みである。

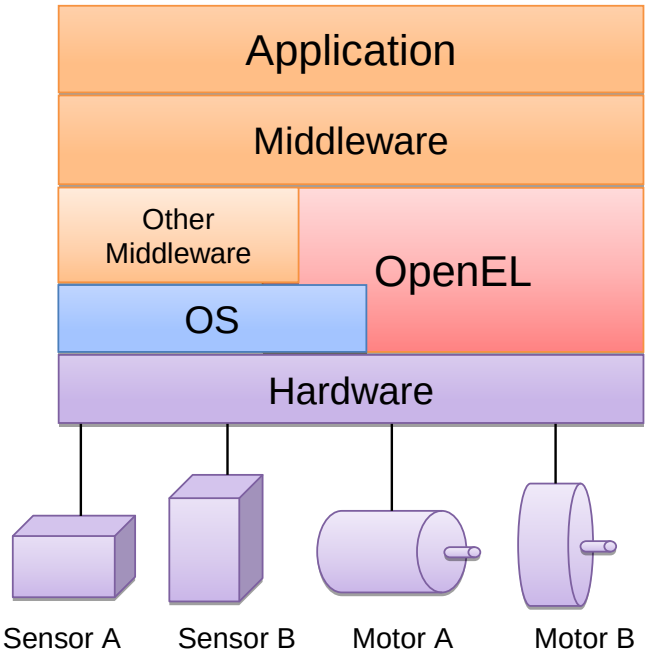

図 4-1 OpenEL の位置づけ

組込みシステムの開発においてデバイスドライバをはじめ既存のソフトウェアを異なるシステムに移植するにはかなりの工数を必要とする。例えば、異なるハードウェア上の LED を点灯させたり、モータを動作させたりするだけで、何日も費やすこともあり得る。これは、センサの入出力やモータの制御などを行うアプリケーションプログラムのインタフェースが、各デバイスメーカにより独自に定義、実装されてきたためである。

OpenEL では、ロボットや制御システムなどのソフトウェアのベースとなる部分をプラットフォーム化することにより、異なるハードウェアでもすぐにアプリケーションを動作可能とすることを目的とする。これにより、ソフトウェアの移植性や再利用性が高まり、その結果、品質向上、コスト削減、生産性向上につながるなど、開発者および利用者の利便性が向上する。

5. OpenEL の実装

OpenEL 1.0 版は ISO/IEC 9899:1999 にて定義された C 言語(一般に C99 と呼ばれる)にて実装されることを前提としている。OpenEL 1.0 仕様の API は、C 言語の関数として定義される。

6. OpenEL の構成

OpenEL は、「Surface」と「Component」のレイヤから構成される。

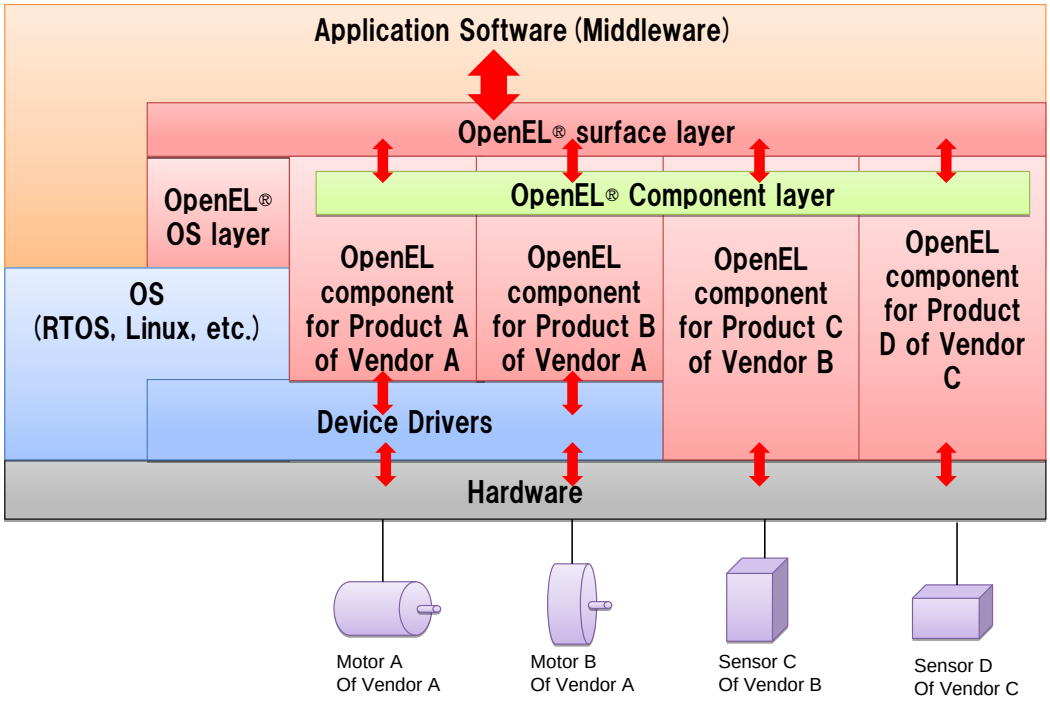

図 6-1 OpenEL の構成図

6.1 Surface(サーフェース) レイヤ

OpenEL 利用者向けに提供する API(Application Program Interface)を定義する。ミドルウェアを含むアプリケーションソフトウェアの開発者は、Surface レイヤに定義された API のみを使用することによって、各デバイスハードウェアの差異を意識することなくソフトウェアの構築が可能となる。

Surface レイヤは通常 JASA より提供と公開がなされ、改版・追加・変更・削除は本仕様書に基づき JASA のみが実施する。

6.2 Component (コンポーネント)レイヤ

ミドルウェアを含むアプリケーションソフトウェアの開発者は Component レイヤの存在を意識する必要はない。

Component レイヤは、モータやセンサなどのハードウェアと共に、使用するデバイス毎にそのデバイスの製造者又は供給者により提供される。OpenEL 仕様に準拠した実装の Component が提供されることにより、Surface レイヤの API から実際のデバイス操作への橋渡しを行う。Component レイヤはデバイスドライバソフトウェアを含む処理の実態であることも出来るし、デバ

イスドライバソフトウェアのラッパー関数であることも可能である。

① OpenEL コンポーネント名

Component は固有の名前「OpenEL コンポーネント名」を割り当てられる。コンポーネント名は以下の規則にて決定し、割り当てられたコンポーネント名は JASA にて承認し管理される。

el ＋(デバイス名) ＋(ベンダ名) ＋(シリーズ名)

表 6-1 コンポーネント命名規約

No	名称	決定者	概要
1	el	固定	OpenEL コンポーネントであることを表す prefix。
2	(デバイス名)	OpenEL 仕様	デバイスの種類を表す名称。 (例:Motor, Sensor 等)
3	(ベンダ名)	JASA	大文字から始まる 2～16 文字の英字。JASA が提案する単語からベンダが選定し申請する。
4	(シリーズ名)	ベンダ	製品名や型番を表すベンダ固有の名称。

例:ベンダ「OM」、製品名「ABC」のモータ :「elMotorOMABC」

② 物理 ID

OpenEL に準拠するデバイスは、Component を介して全てのデバイスに必ずユニークな物理 ID が割り当てられる。

物理 ID は 32bit 符号無し整数型で定義され、0x00000000~0xFFFFFFFF の値を持ち、16bit ずつ上位と下位に分けて決定される。上位 16bit は JASA にて規定し管理されコンポーネント製造ベンダに割り当てられる。

表 6-2 物理 ID の構成

No	開始番号	終了番号	用途
1	－	0x0000	開発用
2	0x0001	0x000F	予約領域
3	0x0010	0xFFFE	ベンダ割り当て領域(JASA にて管理)
4	0xFFFF	－	開発用

下位 16bit は、上位 16bit を割り当てられたデバイス提供者が自由に割り振ることが可能である。

7. API 仕様

OpenEL 1.0 版にて定義する API を示す。OpenEL 1.0 版に準拠する製品は、該当するデバイス毎に全 API を実装しなければならない。製品の特性上機能しない API(速度制御のみに対応するモータで、トルク制御 API・位置制御 API をサポートできない場合など)は、全 API を実装し非サポート API は API の戻り値として(-1)を返却する様実装すること。

7.1 型定義

OpenEL 1.0 版では以下の型定義を使用する。

表 7-1 OpenEL 型定義一覧

No	型定義	説明	備考
1	ELBool	真偽型	true またはfalseをとる
2	ELChar	符号付き8bit整数	
3	ELUChar	符号無し8bit整数	
4	ELInt16	符号付き16bit整数	
5	ELUInt16	符号無し16bit整数	
6	ELInt32	符号付き32bit整数	
7	ELUInt32	符号無し32bit整数	
8	ELFloat64	符号付き64bit浮動小数	
9	ELUFloat64	符号無し64bit浮動小数	
101	EL_CMN_FNC_TBL_T	構造体	コンポーネントテーブル型
102	EL_VENDAR_CONF_T	構造体	ベンダー固有設定情報型(void*)

7.2 共通 API

■デバイス初期化

表 7-2 デバイス初期化 API

関数名	**elInit**	実装版数	1.0以降
ELInt32 **elInit**(ELUInt32 portID, EL_CMN_FNC_TBL_T fncTbl, ELUInt32 physicalPortID, EL_VENDAR_CONF_T *pConfig) □			
OpenELデバイスの初期化を行う。			
コンポーネントテーブル名		pFuncInit	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	EL_CMN_FNC_TBL_T	fncTbl	ベンダーで準備された関数テーブル
	ELUInt32	physicalPortID	ベンダーによる接続用portID
	EL_VENDAR_CONF_T *	pCnfig	ベンダー準拠のコンフィグレーション情報
戻り値	ELInt32	0:エラーなし / 0:以外 設定できない	

■デバイス終了

表 7-3 デバイス終了 API

関数名	**elExit**	実装版数	1.0以降
ELInt32 **elExit**(ELUInt32 portID) □			
OpenELデバイスを終了させる。 再度このデバイスを使用するためにはもう一度初期化する必要がある。			
コンポーネントテーブル名		pFuncExit	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
戻り値	ELInt32	0:エラーなし / 0:以外 異常終了	

7.3 モータ デバイス

■制御状態

表 7-4 elMotorSetControlMode API

関数名	elMotorSetControlMode	実装版数	1.0以降
ELInt32 elMotorSetControlMode(ELUInt32 portID, ELInt32 mode)			
モータに制御モードを設定する。			
コンポーネントテーブル名		pFncSetCtlMode	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELInt32	mode	制御モード 0:制御無し 1:位置制御 2:速度制御 3:トルク制御
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

表 7-5 elMotorGetControlMode API

関数名	elMotorGetControlMode	実装版数	1.0以降
ELInt32 elMotorGetControlMode(ELUInt32 portID, ELInt32* pMode)			
モータの現在の制御モードを取得する。			
コンポーネントテーブル名		pFncGetCtlMode	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELInt32 *	pMode	[outパラメータ]制御モード 0:制御無し 1:位置制御 2:速度制御 3:トルク制御
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

モータは以下のような制御状態遷移を行う。ただし、全ての制御モードを実装することは必須ではなく、製品仕様により実装されるモードは一部であることが許される。

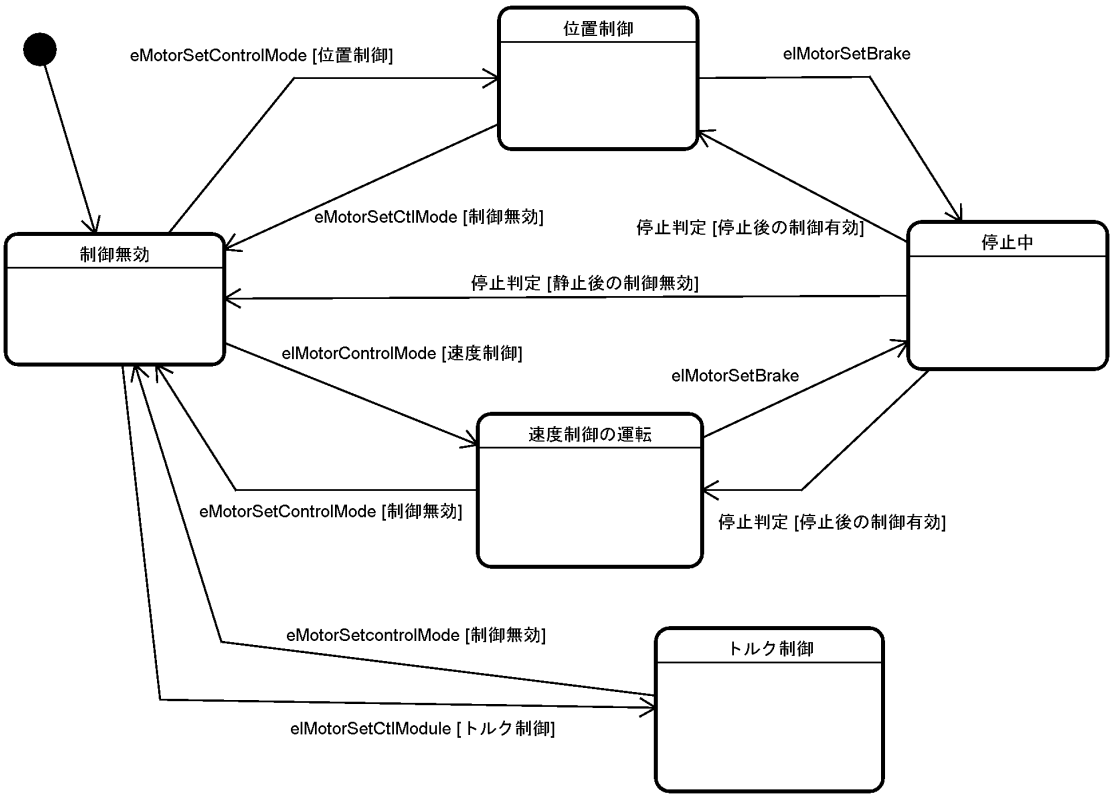

図 7-1 モータ制御モードによる状態遷移

■位置制御

表 7-6 elMotorSetPosition API(整数型)

関数名	elMotorSetPosition_I32	実装版数	1.0以降
ELInt32 elMotorSetPosition_I32(ELUInt32 portID, ELInt32 t position, ELInt32 speed, ELInt32 tmAcc)			
OpenELモーター 位置決め運転を行う。			
コンポーネントテーブル名		pFncSetPosition	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELInt32	position	絶対位置指令 [ユーザー単位]
	ELInt32	speed	速度 [ユーザー単位/s] 範囲 0より大きい～製品仕様
	ELInt32	tmAcc	加減速時間 [ms](I32),[s](F64) 0以上～製品仕様
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

表 7-7 elMotorSetPosition API(浮動小数点型)

関数名	elMotorSetPosition_F64	実装版数	1.0以降
ELInt32 elMotorSetPosition_F64(ELUInt32 portID, ELFloat64 position, ELInt32 speed, ELFloat64 tmAcc)			
OpenELモーター 位置決め運転を行う。			
コンポーネントテーブル名		pFncSetPosition	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELFloat64	position	絶対位置指令 単位 直動系 [m] , 回転系[rad]
	ELFloat64	speed	速度 範囲 0.0より大きい～製品仕様 単位 直動系 [m/s] , 回転系[rad/s]
	ELFloat64	tmAcc	加減速時間 [ms](I32),[s](F64) 0以上～製品仕様
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

表 7-8 elMotorGetPosition API(整数型)

関数名	elMotorGetPosition_I32	実装版数	1.0以降
ELInt32 elMotorGetPosition_I32(ELUInt32 portID , ELUInt32 sw , ELInt32 * pAngle)			
モータの現在位置を取得する。			
コンポーネントテーブル名		pFncGetPosition	
パラメータ	型	名前	説明
	ELUInt32	portID	モータのポートID
	ELUInt32	sw	取得する値 0:現在の検出位置 1:指定された目標位置 2:現在の指令位置
	ELInt32 *	pPosition	[outパラメータ]位置
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

表 7-9 elMotorGetPosition API(浮動小数点型)

関数名	**elMotorGetPosition_F64**	実装版数	1.0以降
ELInt32 **elMotorGetPosition_F64**(ELUInt32 portID , ELUInt32 sw , ELFloat64 * pAngle)			
モータの現在位置を取得する。			
コンポーネントテーブル名		pFncGetPosition	
パラメータ	型	名前	説明
	ELUInt32	portID	モータのポートID
	ELUInt32	sw	取得する値 0:現在の検出位置 1:指定された目標位置 2:現在の指令位置
	ELFloat64 *	pPosition	[outパラメータ]位置
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

表 7-10 elMotorPreSetPosition API(整数型)

関数名	**elMotorPreSetPosition_I32**	実装版数	1.0以降
ELInt32 **elMotorPreSetPosition_I32**(ELUInt32 portID, ELInt32 angle)			
OpenELモーター 位置座標の設定を行う。			
コンポーネントテーブル名		pFncPreSetPosition	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELInt32	angle	絶対位置 [ユーザー単位]
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

表 7-11 elMotorPreSetPosition API(浮動小数点型)

関数名	**elMotorPreSetPosition_F64**	実装版数	1.0以降
ELInt32 **elMotorPreSetPosition_F64**(ELUInt32 portID, ELFloat64 angle)			
OpenELモーター 位置座標の設定を行う。			
コンポーネントテーブル名		pFncPreSetPosition	
パラメータ	型	名前	説明
	ELUInt32	portID	[in] ポートID
	ELFloat64	angle	[in] 絶対位置 [ユーザー単位]
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

■速度制御

表 7-12　elMotorSetSpeed API(整数型)

関数名	**elMotorSetSpeed_I32**	実装版数	1.0以降
ELInt32 **elMotorSetSpeed_I32**(ELUInt32 portID, ELInt32 speed)			
OpenELモーター 速度制御運転を行う。			
コンポーネントテーブル名		pFncSetSpeed	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELInt32	speed	速度 [ユーザー単位/s] 範囲 製品仕様(正転/逆転は符号で指定)
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

表 7-13　elMotorSetSpeed API(浮動小数点型)

関数名	**elMotorSetSpeed_F64**	実装版数	1.0以降
ELInt32 **elMotorSetSpeed_F64**(ELUInt32 portID, ELFloat64 speed)			
OpenELモーター 速度制御運転を行う。			
コンポーネントテーブル名		pFncSetSpeed	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELFloat64	speed	速度 範囲 製品仕様(正転/逆転は符号で指定) 単位　直動系 [m/s] , 回転系[rad/s]
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

表 7-14　elMotorGetSpeed API(整数型)

関数名	**elMotorGetSpeed_I32**	実装版数	1.0以降
ELInt32 **elMotorGetSpeed_I32**(ELUInt32 portID, ELInt32 idx, ELInt32 *pOutSpeed)			
OpenELモーター 速度の取得を行う。			
コンポーネントテーブル名		pFncGetSpeed	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELInt32	idx	0:現在の検出速度 1:指定された目標速度 2:現在の指令速度
	ELInt32 *	pOutSpeed	[outパラメータ]速度
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

表 7-15　elMotorGetSpeed API(浮動小数点型)

関数名	**elMotorGetSpeed_F64**	実装版数	1.0以降
ELInt32 **elMotorGetSpeed_F64**(ELUInt32 portID, ELInt32 idx, ELFloat64 *pOutSpeed)			
OpenELモーター 速度の取得を行う。			
コンポーネントテーブル名		pFncGetSpeed	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELInt32	idx	0:現在の検出速度 1:指定された目標速度 2:現在の指令速度
	ELFloat64 *	pOutSpeed	[outパラメータ] 速度
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

■トルク制御

表 7-16　elMotorSetTorque API(整数型)

関数名	**elMotorSetTorque_I32**	実装版数	1.0以降
ELInt32 **elMotorSetTorque_I32**(ELUInt32 portID, ELInt32 torque)			
OpenELモーター トルク制御運転を行う。			
コンポーネントテーブル名		pFncSetTorque	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELInt32	torque	トルク 単位は 0.1[%] 範囲は製品使用
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

表 7-17 elMotorSetTorque API(浮動小数点型)

関数名	**elMotorSetTorque_F64**	実装版数	1.0以降
ELInt32 **elMotorSetTorque_F64**(ELUInt32 portID, ELFloat64 torque)			
OpenELモーター トルク制御運転を行う。			
コンポーネントテーブル名		pFncSetTorque	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELFloat64	torque	[Nm](revolution),[N](Liner) 範囲は製品使用
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

表 7-18　elMotorGetTorque API(整数型)

関数名	**elMotorGetTorque_I32**	実装版数	1.0以降
ELInt32 **elMotorGetTorque_I32**(ELUInt32 portID, ELint32 idx, ELInt32* pTorque)			
現在設定されたトルク値を取得する			
コンポーネントテーブル名		pFncGetTorque	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELInt32	idx	0:現在の発生トルク 1:指定されたトルク
	ELInt32 *	pTorque	[outパラメータ] トルク 単位は 0.1[%] 範囲は製品使用
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

表 7-19 elMotorGetTorque API(浮動小数点型)

関数名	**elMotorGetTorque_F64**	実装版数	1.0以降
ELInt32 **elMotorGetTorque_F64**(ELUInt32 portID, ELInt32 idx, ELFloat64* pTorque)			
現在設定されたトルク値を取得する			
コンポーネントテーブル名		pFncGetTorque	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELInt32	idx	0:現在の発生トルク 1:指定されたトルク
	ELFloat64 *	pTorque	[outパラメータ] [Nm](revolution),[N](Liner) 範囲は製品使用
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

■ブレーキ

表 7-20　elMotorSetBrake API

関数名	**elMotorSetBrake**	実装版数	0.1以降
ELInt32 **elMotorSetBrake**(ELUInt32 portID, ELBool flagBrake)			
モータのブレーキを有効・無効にする。			
コンポーネントテーブル名		pFncSetBreake	
パラメータ	型	名前	説明
	ELUInt32	portID	モータのポートID
	ELBool	flagBrake	ブレーキを有効にする場合はtrue、無効にする場合はfalse
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

表 7-21 elMotorGetBrake API

関数名	**elMotorGetBrake**	実装版数	0.1以降
ELInt32 **elMotorGetBrake**(ELUInt32 portID, ELBool *pFlagBrake)			
OpenELモーター ブレーキ状態の取得する。			
コンポーネントテーブル名		pFncGetBreake	
パラメータ	型	名前	説明
	ELUInt32	portID	モータのポートID
	ELBool *	pFlagBrake	[outパラメータ] true:制動中 / false:停止制御に入っていない
戻り値	ELInt32	エラーコード(0:正常 / 0以外:異常(運転を受け付けられない状態))	

7.4 Bluetooth デバイス

■初期化・終了

表 7-22 elBluetoothInitializeMaster API

関数名	**elBluetoothInitializeMaster**	実装版数	0.1以降
ELInt32 **elBluetoothInitializeMaster**(ELUInt32 portID, const ELUChar *pAddr, const ELChar *pPin) □			
Bluetoothをマスタデバイスとして初期化する。			
コンポーネントテーブル名		–	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	const ELUChar *	pAddr	スレーブデバイスのBluetoothデバイスアドレスの先頭アドレス
	const ELChar *	pPin	パスキー交換用ピンコードの先頭アドレス
戻り値	ELInt32	0:エラーなし / 0:以外 設定できない	

表 7-23 elBluetoothInitializeSlave API

関数名	**elBluetoothInitializeSlave**	実装版数	0.1以降
ELInt32 **elBluetoothInitializeSlave**(ELUInt32 portID, const ELChar *pPin) □			
Bluetoothをスレーブデバイスとして初期化する。			
コンポーネントテーブル名		–	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	const ELChar *	pPin	パスキー交換用ピンコードの先頭アドレス
戻り値	ELInt32	0:エラーなし / 0:以外 設定できない	

表 7-24 elBluetoothTerminate API

関数名	**elBluetoothTerminate**	実装版数	0.1以降
ELInt32 **elBluetoothTerminate**(ELUInt32 portID) □			
Bluetoothの終了処理を行う。 この関数はマスタデバイス・スレイブデバイスの両方で使用可能である。			
コンポーネントテーブル名		–	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
戻り値	ELInt32	0:エラーなし / 0:以外 設定できない	

■データ送受信

表 7-25 elBluetoothSendData API

関数名	elBluetoothSendData	実装版数	0.1以降
ELInt32 **elBluetoothSendData**(ELUInt32 portID, const ELUInt8 *buf, ELUInt32 offset, ELUInt32 len,			
Bluetooth経由でバッファのデータを送信する。			
コンポーネントテーブル名		－	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	const ELUInt8 *	buf	送信データバッファの先頭アドレス
	ELUInt32	offset	送信データバッファ内のオフセット値
	ELUInt32	len	送信データバッファのバイトサイズ
	ELUInt32 *	outsize	[outパラメータ]送信されたデータのバイト数
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

表 7-26 elBluetoothReceiveData API

関数名	elBluetoothReceiveData	実装版数	0.1以降
ELInt32 **elBluetoothReceiveData**(ELUInt32 portID, ELUInt8 *buf, ELUInt32 offset, ELUInt32 len, ELUInt32 * outsize) □			
Bluetooth経由でバッファにデータを受信する。			
コンポーネントテーブル名		－	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELUInt8 *	buf	受信データバッファの先頭アドレス
	ELUInt32	offset	受信データバッファ内のオフセット値
	ELUInt32	len	受信データバッファのバイトサイズ
	ELUInt32 *	outsize	[outパラメータ]受信したデータのバイト数
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

■状態取得

表 7-27 elBluetoothGetDeviceName API

関数名	elBluetoothGetDeviceName	実装版数	0.1以降
ELInt32 elBluetoothGetDeviceName(ELUInt32 portID, ELChar *pName)			
Bluetoothのデバイス名を取得する。			
コンポーネントテーブル名		－	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELChar *	pName	[outパラメータ]デバイス名を格納するバッファの先頭アドレス
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

表 7-28 elBluetoothSetDeviceName API

関数名	elBluetoothSetDeviceName	実装版数	0.1以降
ELInt32 elBluetoothSetDeviceName(ELUInt32 portID, const ELChar *pName)			
Bluetoothのデバイス名を設定する。			
コンポーネントテーブル名		－	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	const ELChar *	pName	設定するデバイス名
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

表 7-29 elBluetoothGetStatus API

関数名	elBluetoothGetStatus	実装版数	0.1以降
ELInt32 elBluetoothGetStatus(ELUInt32 portID, ELInt32 * pStatus)			
Bluetoothの接続状態を取得する。			
コンポーネントテーブル名		－	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELInt32 *	pStatus	[outパラメータ]接続状態定数一覧（仮）: ・4:EL_BT_NO_INIT(未初期化状態) ・5:EL_BT_INITIALIZED(初期化状態) ・6:EL_BT_CONNECTED(接続確立状態) ・7:EL_BT_STREAM(データ送受信可能状態
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

表 7-30 elBluetoothGetSignalStrength API

関数名	**elBluetoothGetSignalStrength**	実装版数	0.1以降
ELInt32 **elBluetoothGetSignalStrength**(ELUInt32 portID, ELInt16 * pStrength)			
Bluetoothの電波強度を取得する。			
コンポーネントテーブル名		-	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELInt16 *	pStrength	[outパラメータ]Bluetoothの電波強度(範囲:[0,100]) 未接続状態の場合は-1が返る。
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

7.5　光センサデバイス

■状態取得

表 7-31 elLightSensorGetValue API

関数名	elLightSensorGetValue	実装版数	0.1以降
ELInt32 elLightSensorGetValue(ELUInt32 portID, ELUInt16 * pValue) □			
光センサ値を取得する。			
コンポーネントテーブル名		－	
パラメータ	型	名前	説明
	ELUInt32	portID	光センサのポートID
	ELUInt16 *	pValue	[outパラメータ]光センサ値
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

表 7-32 elLightSensorGetLED API

関数名	elLightSensorGetLED	実装版数	0.1以降
ELInt32 elLightSensorGetLED(ELUInt32 portID , ELBool * pStatus) □			
光センサのLEDの点灯・消灯状態を取得する。			
コンポーネントテーブル名		－	
パラメータ	型	名前	説明
	ELUInt32	portID	光センサのポートID
	ELBool *	pStatus	[outパラメータ]LEDが点灯状態:true、消灯状態:false
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

■設定

表 7-33 elLightSensorSetLED API

関数名	elLightSensorSetLED	実装版数	0.1以降
ELInt32 elLightSensorSetLED(ELUInt32 portID, ELBool light) □			
光センサのLEDの点灯・消灯状態を設定する。			
コンポーネントテーブル名		－	
パラメータ	型	名前	説明
	ELUInt32	portID	光センサのポートID
	ELBool	light	LEDを点灯状態にする場合はtrue、消灯状態にする場合はfalse
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

7.6　タッチセンサデバイス

■状態取得

表 7-34 elTouchSensorGetState API

関数名	elTouchSensorGetState	実装版数	0.1以降
ELInt32 elTouchSensorGetState(ELUInt32 portID , ELBool *pState) □			
タッチセンサのON・OFFを取得する。			
コンポーネントテーブル名		–	
パラメータ	型	名前	説明
	ELUInt32	portID	タッチセンサのポートID
	ELBool *	pState	[outパラメータ]ONの場合はtrue、OFFの場合はfalse
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

7.7 スピーカデバイス

■出力

表 7-35 elSpeakerOutput API

関数名	elSpeakerOutput	実装版数	0.1以降
ELInt32 **elSpeakerOutput**(ELUInt32 portID, ELUInt32 freq, ELUInt32 ms, ELUInt32 vol)			
スピーカからトーンを出力する。			
コンポーネントテーブル名		–	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELUInt32	freq	周波数(単位:Hz)
	ELUInt32	ms	出力継続時間(単位:10ms)
	ELUInt32	vol	ボリューム(単位:%)
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

7.8　バッテリーデバイス

■状態取得

表 7-36 elBatteryGetVoltage API

関数名	elBatteryGetVoltage	実装版数	0.1以降
ELInt32 **elBatteryGetVoltage**(ELUInt32 portID, ELUInt16 *pVoltage) □			
バッテリの電圧値を取得する。			
コンポーネントテーブル名		–	
パラメータ	型	名前	説明
	ELUInt32	portID	ポートID
	ELUInt16 *	pVoltage	[outパラメータ]現在のバッテリの電圧値
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

7.9　超音波センサデバイス

■初期化・終了

表 7-37 elSonarSensorInitialize API

関数名	elSonarSensorInitialize	実装版数	0.1以降
ELInt32 elSonarSensorInitialize(ELUInt32 portID) □			
超音波センサの初期化処理を行う。 この関数は超音波センサを使用する前に一度だけ呼び出す必要がある。			
コンポーネントテーブル名		－	
パラメータ	型	名前	説明
	ELUInt32	portID	超音波センサのポートID
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

表 7-38 elSonarSensorTerminate API

関数名	elSonarSensorTerminate	実装版数	0.1以降
ELInt32 elSonarSensorTerminate(ELUInt32 portID) □			
超音波センサの終了処理を行う。 この関数は超音波センサの使用を終了する前に一度だけ呼び出す必要がある。			
コンポーネントテーブル名		－	
パラメータ	型	名前	説明
	ELUInt32	portID	超音波センサのポートID
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

■状態取得

表 7-39 elSonarSensorGetValue API

関数名	elSonarrSensorGetValue	実装版数	0.1以降
ELInt32 elSonarSensorGetValue(ELUInt32 portID, ELInt32 * pValue) □			
超音波センサ値を取得する。 初期化が行われていない場合、-1が返る。			
コンポーネントテーブル名		－	
パラメータ	型	名前	説明
	ELUInt32	portID	超音波センサのポートID
	ELInt32 *	pValue	[outパラメータ]計測距離(単位：cm)
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

7.10 ジャイロセンサデバイス

■状態取得

表 7-40 elGyroSensorGetValue API

関数名	elGyroSensorGetValue	実装版数	0.1以降
ELInt32 elGyroSensorGetValue(ELUInt32 portID, ELUInt16 * pValue) □			
ジャイロセンサ値を取得する。			
コンポーネントテーブル名		－	
パラメータ	型	名前	説明
	ELUInt32	portID	ジャイロセンサのポートID
	ELUInt16 *	pValue	[outパラメータ]ジャイロセンサ値
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

表 7-41 elGyroSensorGetOffset API

関数名	elGyroSensorGetOffset	実装版数	0.1以降
ELInt32 elGyroSensorGetOffset(ELUInt32 portID, ELUInt16 * pOffset) □			
ジャイロセンサのオフセット値を取得する。			
コンポーネントテーブル名		－	
パラメータ	型	名前	説明
	ELUInt32	portID	ジャイロセンサのポートID
	ELUInt16 *	pOffset	[outパラメータ]ジャイロセンサのオフセット値
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

■設定

表 7-42 elGyroSensorSetOffset API

関数名	elGyroSensorSetOffset	実装版数	0.1以降
ELInt32 elGyroSensorSetOffset(ELUInt32 portID, ELUInt16 offset) □			
ジャイロセンサのオフセット値を設定する。			
コンポーネントテーブル名		－	
パラメータ	型	名前	説明
	ELUInt32	portID	ジャイロセンサのポートID
	ELUInt16	offset	設定するオフセット値
戻り値	ELInt32	0:エラーなし / 0:以外 異常発生	

8. お問合せ

「OpenEL仕様書 1.0-1版」

2013/5/22 発行

2014/3/01 改定

発行者　一般社団法人 組込みシステム技術協会

東京都中央区日本橋浜町1丁目8-1

TEL: 03(5821)7973 FAX: 03(5821)0444

URL: http://www.jasa.or.jp