

---

# KEIm SoM

## SteppingMotor サンプル アプリケーションマニュアル

Ver.1.0.0



株式会社近藤電子工業



## はじめに

この度は、KEIm 製品をお買い上げいただき誠にありがとうございます。

本製品をご使用になる前に、本マニュアル及び関連資料を十分ご確認ください、使用上の注意を守って正しくご使用ください。



### 取扱い上の注意

- 本書に記載されている内容は、将来予告なく変更されることがあります。本製品のご使用にあたっては、弊社窓口又は弊社ホームページなどで最新の情報をご確認ください。
- 本製品には一般電子機器用部品が使用されています。極めて高い信頼性を要求する装置（航空、宇宙機器、原子力制御機器、生命維持のための医療機器等）には使用しないでください。
- 本製品は国内使用を前提として開発及び製造を行っています。本製品又は本製品を組み込んだ製品を輸出される場合は、お客様の責任において「外国為替及び外国貿易法」及びその他輸出関連法令等を順守し、必要な手続きを行ってください。
- LAN、USB 以外のコネクタへのケーブルの抜き差しは、必ず電源を OFF にした状態で行ってください。
- 水、湿気、ほこり、油煙等の多い場所では使用しないでください。
- 本製品の関連資料の全部又は一部を弊社に無断で使用または複製することを禁止します。
- 本書及び関連資料で取り上げる会社名及び製品名等は、各メーカーの商標または登録商標です。

## お問い合わせ先

- 製品に関するお問い合わせは、下記のメールアドレスよりお願いいたします。

[keim-support@kd-group.co.jp](mailto:keim-support@kd-group.co.jp)

## 目次

1. 概要.....	5
1.1. 使用部品の詳細.....	5
2. 操作方法 .....	5
3. ファイル構成 .....	5
4. フローチャート .....	6
5. 各定義値、関数の説明 .....	10
5.1. 定義式 .....	10
5.2. 変数.....	10
5.3. 関数.....	11
5.3.1. Global 関数.....	11
5.3.1.1. main 関数.....	11
5.3.1.2. SteppingApl_Init 関数 .....	11
5.3.1.3. SteppingApl_Exec 関数.....	11
5.3.1.4. SteppingApl_Irq_Timer 関数 .....	12
5.3.2. static 関数.....	12
5.3.2.1. stepping_apl_is_input 関数 .....	12
5.3.2.2. stepping_apl_is_release 関数 .....	12
5.3.2.3. stepping_apl_start 関数.....	13
5.3.2.4. stepping_apl_stop 関数.....	13
5.3.2.5. stepping_apl_set_next_state 関数 .....	13
5.3.2.6. stepping_apl_set_timer 関数.....	13
5.3.2.7. stepping_apl_set_timer_value 関数 .....	14
6. 更新履歴 .....	15

## 図表目次

表 1.1 使用部品詳細.....	5
表 2.1 操作方法.....	5
表 3.1 ファイル構成.....	5
表 5.1 定義式.....	10
表 5.2 変数 .....	10
表 5.3 main 関数 .....	11
表 5.4 SteppingApl_Init 関数.....	11
表 5.5 SteppingApl_Exec 関数.....	11
表 5.6 SteppingApl_Irq_Timer 関数 .....	12
表 5.7 stepping_apl_is_input 関数 .....	12
表 5.8 stepping_apl_is_release 関数.....	12
表 5.9 stepping_apl_start 関数.....	13
表 5.10 stepping_apl_stop 関数.....	13
表 5.11 stepping_apl_set_next_state 関数 .....	13
表 5.12 stepping_apl_set_timer 関数.....	13
表 5.13 stepping_apl_set_timer_value 関数.....	14
図 4.1 メイン フローチャート 1/3 .....	6
図 4.2 メイン フローチャート 2/3 .....	7
図 4.3 メイン フローチャート 3/3 .....	8
図 4.4 タイマー割り込み フローチャート.....	9

## 1. 概要

本サンプルアプリでは以下の制御を行っています。

1. SW1 を入力で右回転します。
2. SW2 を入力で左回転します。
3. 入力中は回転し続けます。
4. SW を解除で停止します。
5. SW3 を入力で回転速度を上げる設定をします。
6. SW4 を入力で回転速度を下げる設定をします。

フローチャートについては以下をご参照ください。

### 1.1. 使用部品の詳細

SteppingMotor サンプルアプリは以下の部品を使用し作成しております。

本商品に付属されておりませんので、別途ご用意ください。

表 1.1 使用部品詳細

品名	品番	メーカー
ステッピングモータ	ST-42BYG0506H	MERCURY MOTOR.
<a href="http://akizukidenshi.com/catalog/g/gP-05374/">http://akizukidenshi.com/catalog/g/gP-05374/</a>		

## 2. 操作方法

ベースボード上の SW1～4 をそれぞれ押すことにより、ステッピングモータが駆動し回転します。

表 2.1 操作方法

スイッチ	モータの動き
SW1	右に回転
SW2	左に回転
SW3	回転速度を上げる
SW4	回転速度を下げる

## 3. ファイル構成

表 3.1 ファイル構成

ファイル名	内容
main.c	メイン関数
SteppingSampleApl.h	SteppingMotor サンプル関数定義
SteppingSampleApl.c	SteppingMotor サンプル関数

## 4. フローチャート

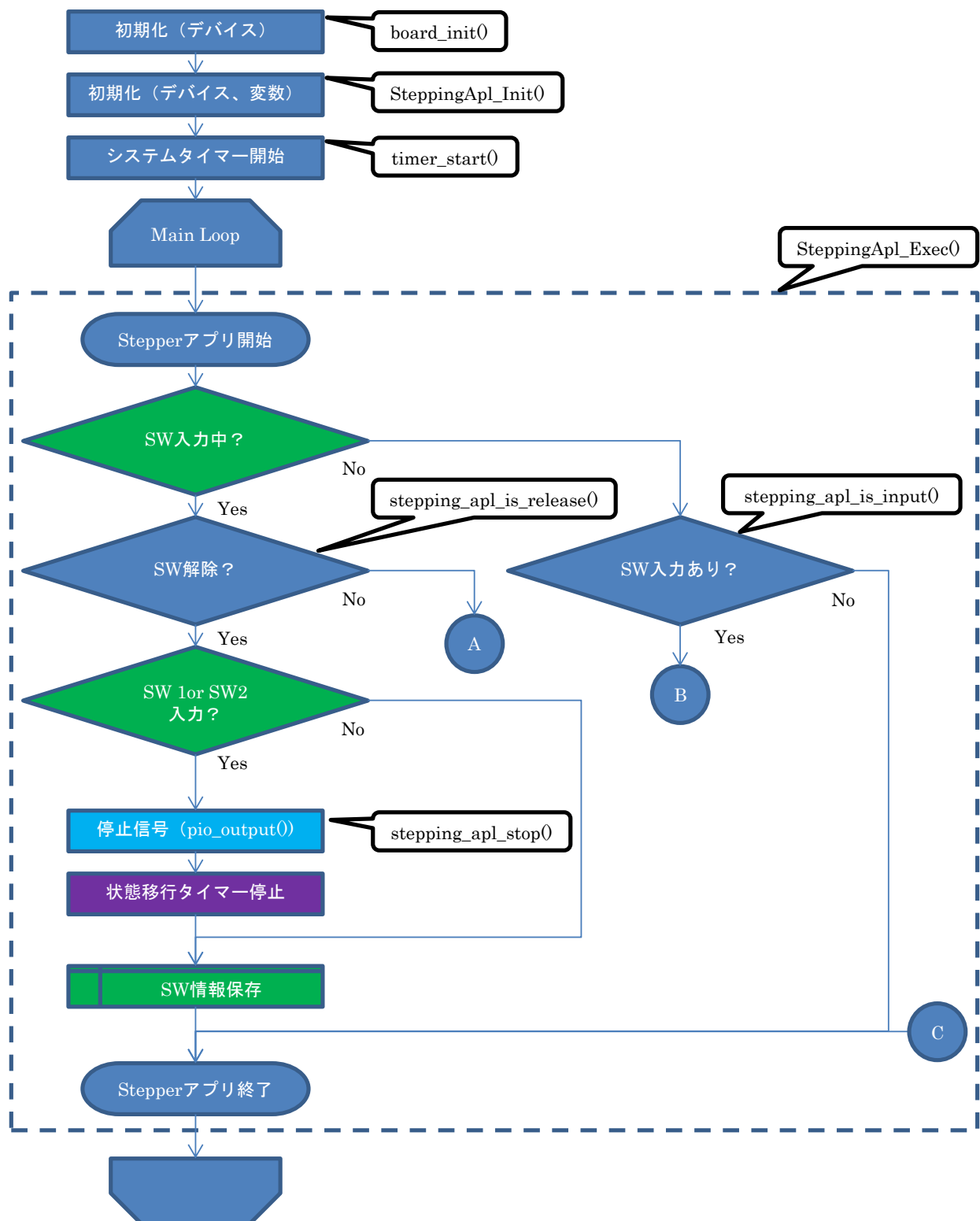


図 4.1 メイン フローチャート 1/3

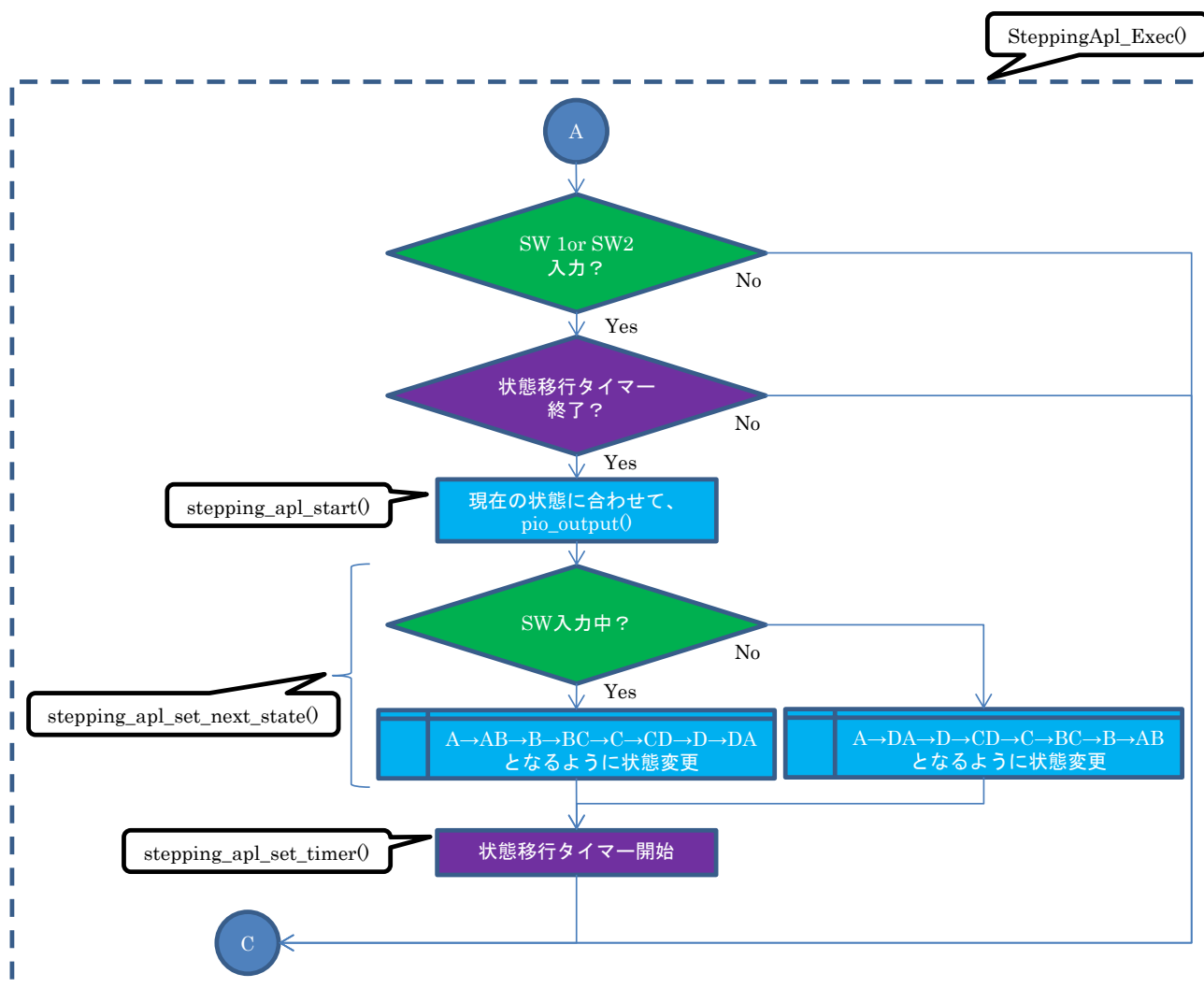


図 4.2 メイン フローチャート 2/3

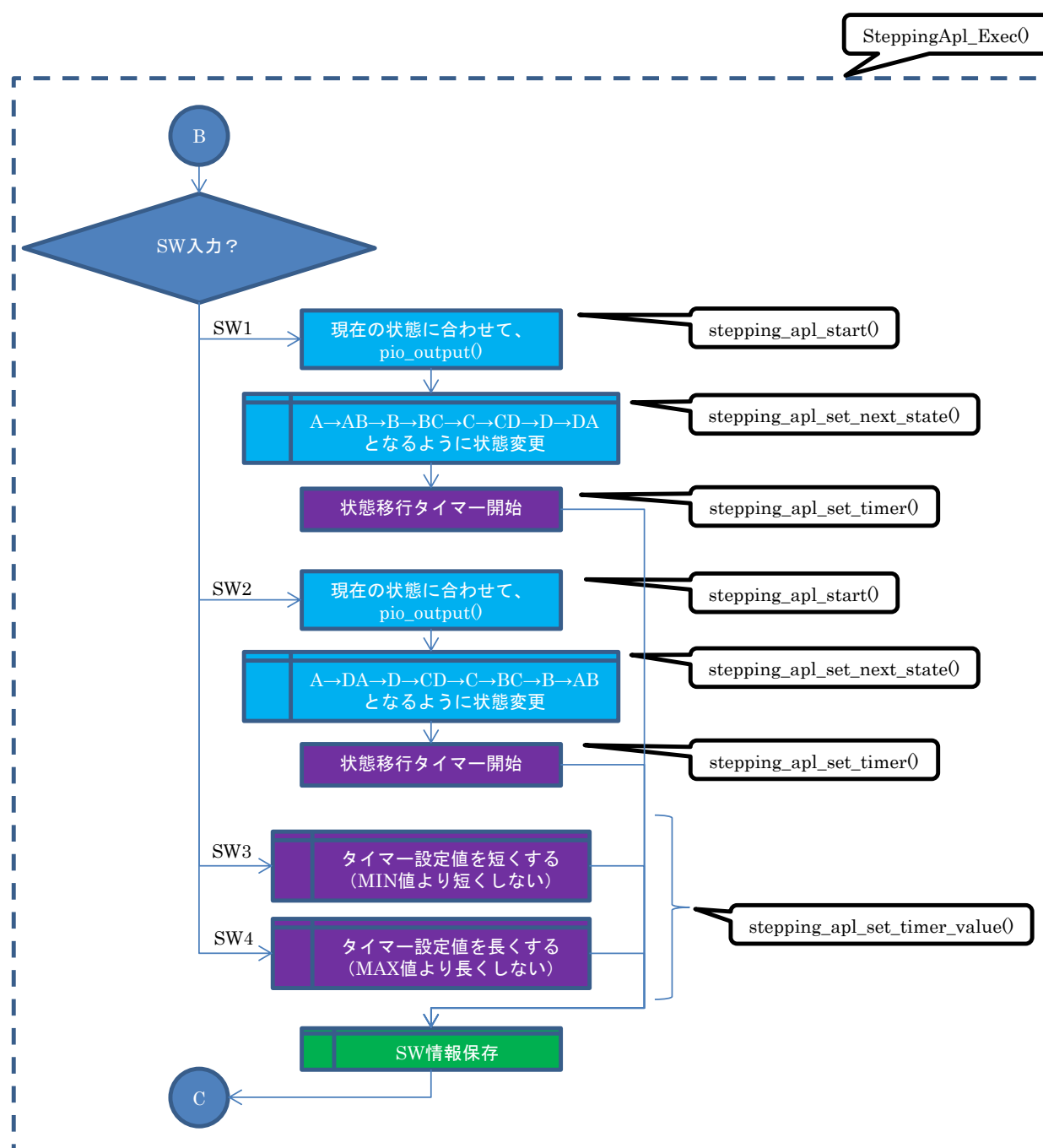


図 4.3 メイン フローチャート 3/3

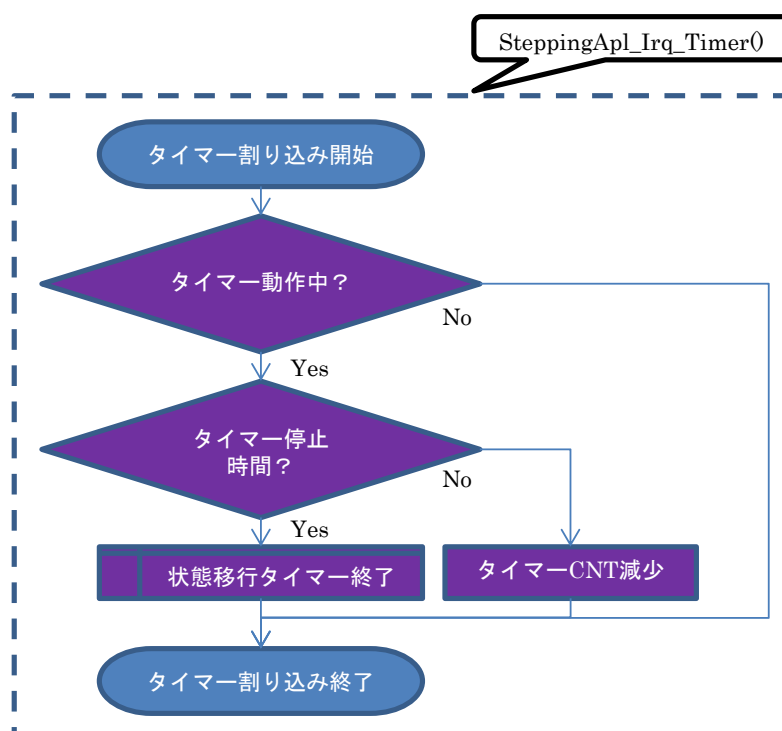


図 4.4 タイマー割り込み フローチャート

## 5. 各定義値、関数の説明

### 5.1. 定義式

表 5.1 定義式

型	名前	値	説明
STEPPING_TIMER_STATE	STEPPING_TIMER_STOP	0x00	タイマー停止状態
STEPPING_TIMER_STATE	STEPPING_TIMER_DOING	0x01	タイマー動作状態
STEPPING_KEY_IN_TYPE	STEPPING_KEY_IN_NONE	0x00	SW 入力なし
STEPPING_KEY_IN_TYPE	STEPPING_KEY_IN_A	0x01	SW1 入力中
STEPPING_KEY_IN_TYPE	STEPPING_KEY_IN_B	0x02	SW2 入力中
STEPPING_KEY_IN_TYPE	STEPPING_KEY_IN_C	0x03	SW3 入力中
STEPPING_KEY_IN_TYPE	STEPPING_KEY_IN_D	0x04	SW4 入力中
STEPPING_STATE	STEPPING_STATE_A	0x00	Stepper 位置 : A
STEPPING_STATE	STEPPING_STATE_AB	0x01	Stepper 位置 : AB
STEPPING_STATE	STEPPING_STATE_B	0x02	Stepper 位置 : B
STEPPING_STATE	STEPPING_STATE_BC	0x03	Stepper 位置 : BC
STEPPING_STATE	STEPPING_STATE_C	0x04	Stepper 位置 : C
STEPPING_STATE	STEPPING_STATE_CD	0x05	Stepper 位置 : CD
STEPPING_STATE	STEPPING_STATE_D	0x06	Stepper 位置 : D
STEPPING_STATE	STEPPING_STATE_DA	0x07	Stepper 位置 : DA
STEPPING_SPEED	STEPPING_SPEED_HIGH	0x00	高速モード
STEPPING_SPEED	STEPPING_SPEED_MEDIUM_HIGH	0x01	中高速モード
STEPPING_SPEED	STEPPING_SPEED_MEDIUM	0x02	中速モード
STEPPING_SPEED	STEPPING_SPEED_MEDIUM_LOW	0x03	中低速モード
STEPPING_SPEED	STEPPING_SPEED_LOW	0x04	低速モード
-	STEPPING_A_PORT	PD10	Stepper A のポート
-	STEPPING_B_PORT	PD11	Stepper B のポート
-	STEPPING_C_PORT	PD12	Stepper C のポート
-	STEPPING_D_PORT	PD13	Stepper D のポート

### 5.2. 変数

表 5.2 変数

型	名前	初期値	説明
STEPPING_TIMER_STATE	stepping_timer_state	STEPPING_TIMER_STOP	タイマー動作状態を表す変数
STEPPING_KEY_IN_TYPE	stepping_key_in	STEPPING_KEY_IN_NONE	現在の SW 入力状態
STEPPING_STATE	stepping_apl_state	STEPPING_STATE_A	現在の Stepper 位置状態
STEPPING_SPEED	stepping_speed_state	STEPPING_SPEED_MEDIUM	現在の回転速度モード
int	stepping_speed_value[5]	50, 100, 250, 500, 750	各回転速度モード用のタイマー値
int	stepping_timer_cnt	0	タイマーCNT 値: この値が0でない間、 タイマー動作中

## 5.3. 関数

## 5.3.1. Global 関数

## 5.3.1.1. main 関数

表 5.3 main 関数

関数名	main
引数	なし
戻り値	int
概要	メイン関数
処理詳細	ボードの初期化 Stepping アプリの初期化 タイマー動作開始 メインループ ・Stepping メイン処理実行

## 5.3.1.2. SteppingApl\_Init 関数

表 5.4 SteppingApl\_Init 関数

関数名	SteppingApl_Init
引数	なし
戻り値	なし
概要	Stepping アプリの初期化
処理詳細	変数の初期化 Stepping 用の各ポートを出力設定

## 5.3.1.3. SteppingApl\_Exec 関数

表 5.5 SteppingApl\_Exec 関数

関数名	SteppingApl_Exec
引数	なし
戻り値	なし
概要	Stepping アプリの実行
処理詳細	SW 入力中に以下を実行 <ul style="list-style-type: none"> <li>・SW 入力がない場合、PIO に停止設定して回転を停止</li> <li>・SW が入力中であり、タイマーが停止している場合、</li> <li>・現在の状態で出力した後、次の状態へ移行</li> </ul> また、次の確認用のタイマーを設定 SW が入力中でない場合に以下を実行 <ul style="list-style-type: none"> <li>・SW 入力を確認、入力により現在の設定で回転開始</li> <li>・次の状態へ移行</li> <li>・また、確認用のタイマーを設定</li> </ul>

## 5.3.1.4. SteppingApl\_Irq\_Timer 関数

表 5.6 SteppingApl\_Irq\_Timer 関数

関数名	SteppingApl_Irq_Timer
引数	なし
戻り値	なし
概要	Stepping アプリのタイマー割り込み
処理詳細	タイマー動作中状態で以下を実行 ・タイマーCNT が 0 の場合にタイマー停止状態の設定 上記以外で、タイマーCNT 値の減算

## 5.3.2. static 関数

## 5.3.2.1. stepping\_apl\_is\_input 関数

表 5.7 stepping\_apl\_is\_input 関数

関数名	stepping_apl_is_input
引数	STEPPING_KEY_IN_TYPE *input_key : 有効 SW 格納先
戻り値	Int        true : あり / false : なし
概要	有効 Input の判定
処理詳細	有効 SW1～4 入力がある場合に以下を実行 ・SW 情報を格納し、true を戻り値に設定 上記以外で、false を戻り値に設定

## 5.3.2.2. stepping\_apl\_is\_release 関数

表 5.8 stepping\_apl\_is\_release 関数

関数名	stepping_apl_is_release
引数	STEPPING_KEY_IN_TYPE input_key : 有効 SW
戻り値	Int        true : 離された / false : なし
概要	SW 解除の判定
処理詳細	指定 SW が離された場合に以下を実行 ・true を戻り値に設定 離されていない場合 false を戻り値に設定

## 5.3.2.3. stepping\_apl\_start 関数

表 5.9 stepping\_apl\_start 関数

関数名	stepping_apl_start
引数	STEPPING_STATE state : 出力する状態
戻り値	なし
概要	Stepping アプリの出力処理
処理詳細	各状態で次のように PORT 出力する (PORT A/B/C/D の状態を表す) STATE A : 1/0/0/0 STATE AB : 1/1/0/0 STATE B : 0/1/0/0 STATE BC : 0/1/1/0 STATE C : 0/0/1/0 STATE CD : 0/0/1/1 STATE D : 0/0/0/1 STATE DA : 1/0/0/1

## 5.3.2.4. stepping\_apl\_stop 関数

表 5.10 stepping\_apl\_stop 関数

関数名	stepping_apl_stop
引数	なし
戻り値	なし
概要	Stepping アプリの出力停止処理
処理詳細	全 PORT 0 出力

## 5.3.2.5. stepping\_apl\_set\_next\_state 関数

表 5.11 stepping\_apl\_set\_next\_state 関数

関数名	stepping_apl_set_next_state
引数	STEPPING_KEY_IN_TYPE in_key : 入力 SW 状態
戻り値	なし
概要	Stepping アプリの状態移行処理
処理詳細	各 SW 入力で次のように位置状態を変更 SW1 : 現在の状態が STATE DA の場合は A に設定 上記以外で、状態を加算 SW2 : 現在の状態が STATE A の場合は DA に設定 上記以外で、状態を減算

## 5.3.2.6. stepping\_apl\_set\_timer 関数

表 5.12 stepping\_apl\_set\_timer 関数

関数名	stepping_apl_set_timer
引数	なし
戻り値	なし
概要	Stepping アプリのタイマー設定
処理詳細	タイマーCNT 値を計算

## 5.3.2.7. stepping\_apl\_set\_timer\_value 関数

表 5.13 stepping\_apl\_set\_timer\_value 関数

関数名	stepping_apl_set_timer_value
引数	STEPPING_KEY_IN_TYPE in_key : 入力 SW 状態
戻り値	なし
概要	Stepping アプリのタイマー設定値変更
処理詳細	<p>各 SW 入力で次のようにタイマー値を変更</p> <p>SW3 : 現在の状態が STEPPING_SPEED_HIGH でない場合、 stepping_speed_state を減算</p> <p>SW4 : 現在の状態が STEPPING_SPEED_LOW でない場合、 stepping_speed_state を加算</p> <p>※上記で設定した speed_state 値から stepping_speed_value[] の Tbl 値を設定</p>

## 6. 更新履歴

Ver.	更新日付	内容
1.0.0	2015/09/14	新規作成