

---

# KEIm SoM LED 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
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 .....	9
5.1. 定義式 .....	9
5.2. 変数 .....	9
5.3. 関数 .....	10
5.3.1. Global 関数.....	10
5.3.1.1. main 関数.....	10
5.3.1.2. LedApl_Init 関数 .....	10
5.3.1.3. LedApl_Set_Type 関数 .....	10
5.3.1.4. LedApl_Exec 関数 .....	11
5.3.1.5. LedApl_Irq_Timer 関数 .....	11
5.3.2. static 関数.....	11
5.3.2.1. led_apl_is_onoff_type 関数 .....	11
5.3.2.2. led_apl_set_timer 関数.....	11
5.3.2.3. led_apl_get_pattarn 関数 .....	12
5.3.2.4. led_apl_chg_type 関数 .....	12
6. 更新履歴 .....	13

## 図表目次

表 2.1 操作方法 .....	5
表 3.1 ファイル構成 .....	5
表 4.1 LED 動作設定 .....	8
表 5.1 定義式 .....	9
表 5.2 変数 .....	9
表 5.3 main 関数 .....	10
表 5.4 LedApl_Init 関数 .....	10
表 5.5 LedApl_Set_Type 関数 .....	10
表 5.6 LedApl_Exec 関数 .....	11
表 5.7 LedApl_Irq_Timer 関数 .....	11
表 5.8 led_apl_is_onoff_type 関数 .....	11
表 5.9 led_apl_set_timer 関数 .....	11
表 5.10 led_apl_get_patturn 関数 .....	12
表 5.11 led_apl_chg_type 関数 .....	12
図 4.1 メイン フローチャート 1/2 .....	6
図 4.2 メイン フローチャート 2/2 .....	7
図 4.3 タイマー割り込み フローチャート .....	8

## 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. LED 点灯/消灯パターンを生成します。
2. I2C 経由で IO Extender に上記パターンの設定を行います。

フローチャートについては以下をご参照ください。

## 2. 操作方法

ベースボード上の SW1 を押すことにより、LED が変化する 8 回目以降は 0 回目に戻りループします。

表 2.1 操作方法

SW1 の押した回数	LED の変化
0 回目	全 LED が点滅
1 回目	交互に点滅
2 回目	シフト点滅
3 回目	1bit 加算点灯
4 回目	1bit 減算点灯
5 回目	任意点灯
6 回目	任意点灯
7 回目	全消灯
8 回目	全点灯

## 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
LedSampleApl.h	Led サンプル関数定義
LedSampleApl.c	Led サンプル関数

## 4. フローチャート

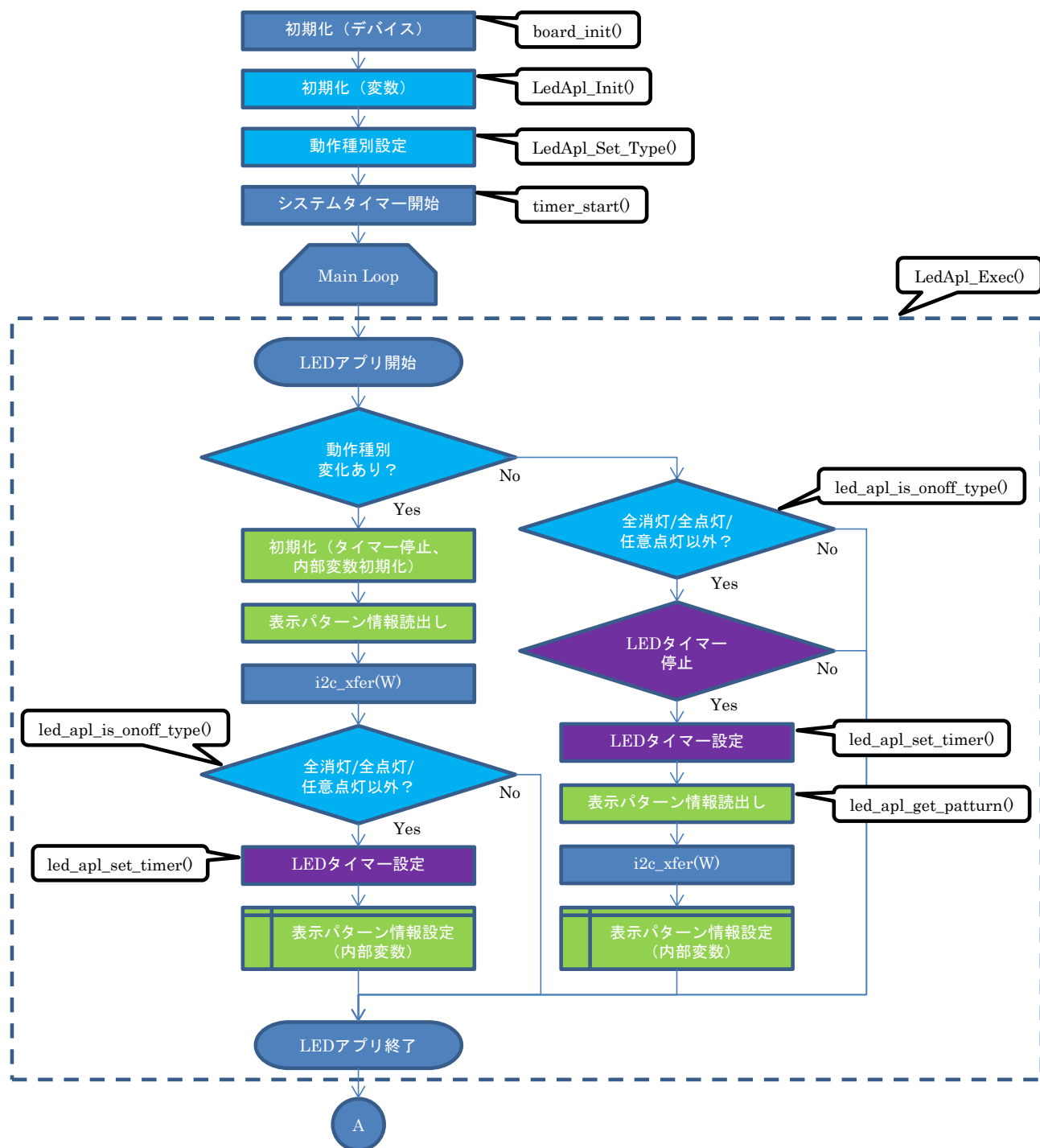


図 4.1 メイン フローチャート 1/2

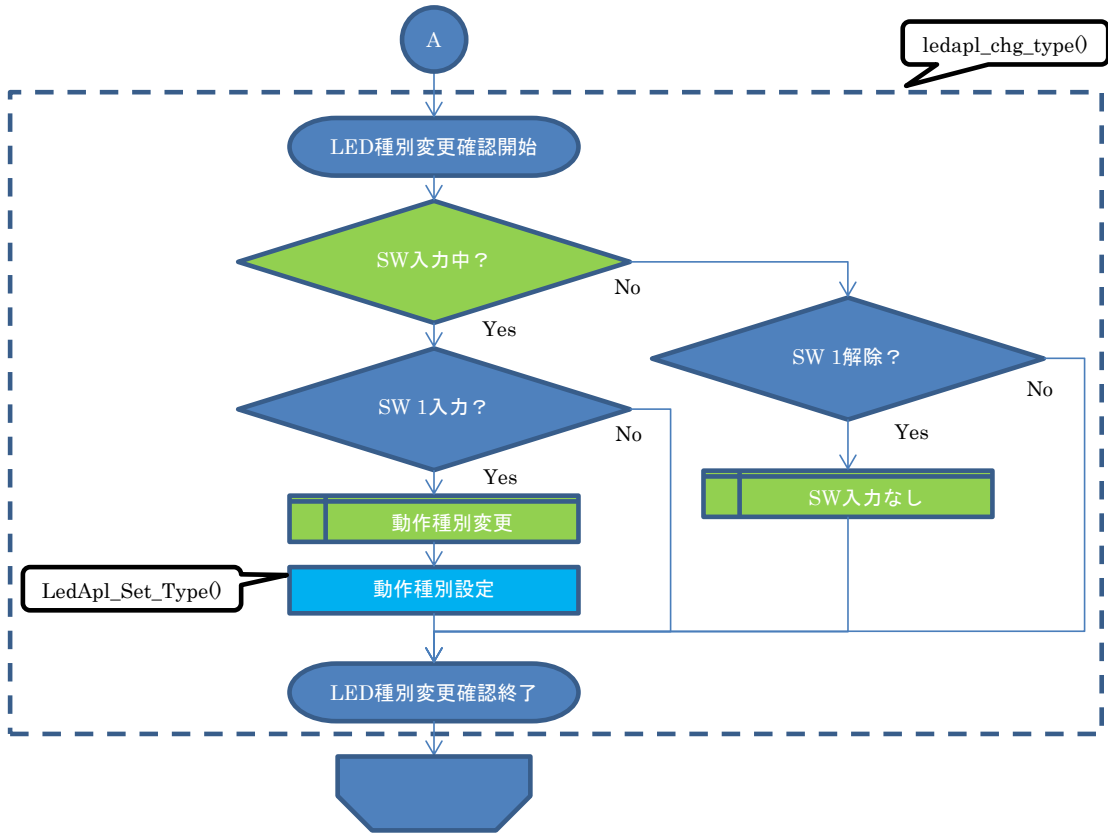


図 4.2 メイン フローチャート 2/2

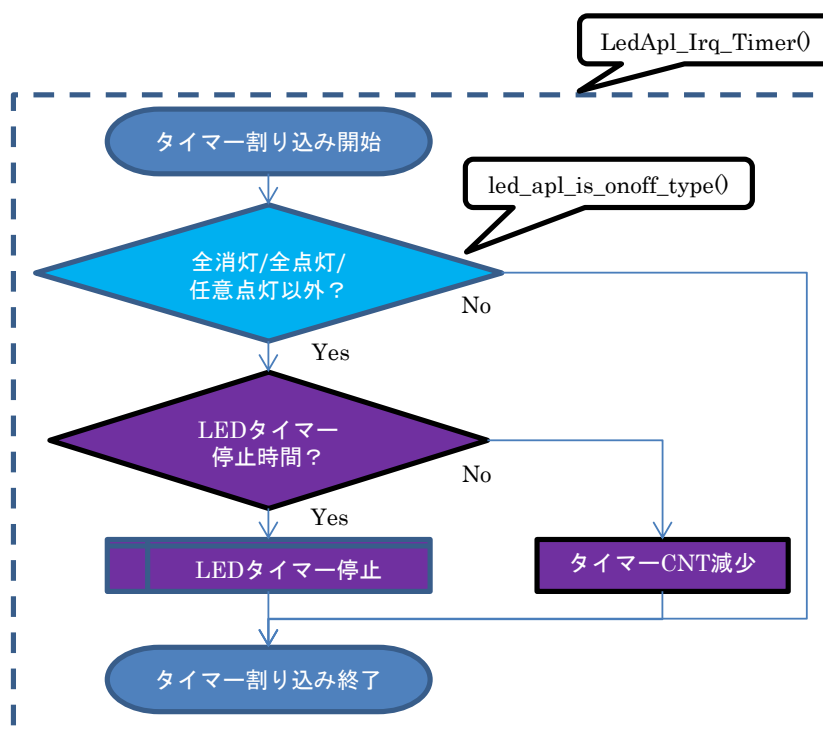


図 4.3 タイマー割り込み フローチャート

表 4.1 LED 動作設定

※動作種別としては以下を定義します。

動作	設定値
全消灯	0x00
全点灯	0xFF
全点減	0xFF⇔0x00
交互点減	0xAA⇔0x55
シフト	0x01→0x02→0x04…0x40→0x80 の繰り返し
インクリメント	0x00 から+1 ずつ、0xFF まで
デクリメント	0xFF から-1 ずつ、0x00 まで
任意点灯	0x00～0xFF まで任意の指定値

## 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
LED_TIMER_STATE	LED_TIMER_STOP	0x00	タイマー停止状態
LED_TIMER_STATE	LED_TIMER_DOING	0x01	タイマー動作状態
LED_APL_TYPE	LED_TYPE_ALLOFF	0x00	LED 動作種別 全消灯
LED_APL_TYPE	LED_TYPE_ALLON	0x01	LED 動作種別 全点灯
LED_APL_TYPE	LED_TYPE_ALLBLINK	0x02	LED 動作種別 全点滅
LED_APL_TYPE	LED_TYPE_BLINK	0x03	LED 動作種別 交互点滅
LED_APL_TYPE	LED_TYPE_SHIFT	0x04	LED 動作種別 シフト
LED_APL_TYPE	LED_TYPE_INC	0x05	LED 動作種別 インクリメント
LED_APL_TYPE	LED_TYPE_DEC	0x06	LED 動作種別 デクリメント
LED_APL_TYPE	LED_TYPE_FREE	0x07	LED 動作種別 任意点灯
—	LED_ALLOFF_INIT_DAT	0x00	全消灯 初期設定値
—	LED_ALLON_INIT_DAT	0xff	全点灯 初期設定値
—	LED_ALLBLINK_INIT_DAT	0xff	全点滅 初期設定値
—	LED_BLINK_INIT_DAT	0xAA	交互点滅 初期設定値
—	LED_SHIFT_INIT_DAT	0x01	シフト 初期設定値
—	LED_INC_INIT_DAT	0x00	インクリメント 初期設定値
—	LED_DEC_INIT_DAT	0xff	デクリメント 初期設定値
—	LED_TIME	250	点灯/消灯時間(250ms)
—	LED_INTVAL_TIME_MS	10	ms 基準値

### 5.2. 変数

表 5.2 変数

型	名前	初期値	説明
LED_TIMER_STATE	led_timer_state	LED_TIMER_STOP	タイマー動作状態を表す変数
alt_u8	led_apl_data	LED_ALLOFF_INIT_DAT	現在設定している LED パターンデータ
LED_APL_TYPE	led_apl_type	LED_TYPE_MAX	現在の LED 動作種別
alt_u8	led_apl_req_data	LED_ALLOFF_INIT_DAT	要求された LED パターンデータ
LED_APL_TYPE	led_apl_req_type	LED_TYPE_MAX	要求された LED 動作種別
Int	led_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 LED アプリの初期化 初期動作設定 タイマー開始処理 メインループ ・LED メイン処理実行 ・LED 動作種別変更設定

## 5.3.1.2. LedApl\_Init 関数

表 5.4 LedApl\_Init 関数

関数名	LedApl_Init
引数	なし
戻り値	なし
概要	LED アプリの初期化
処理詳細	変数の初期化 LED の出力設定

## 5.3.1.3. LedApl\_Set\_Type 関数

表 5.5 LedApl\_Set\_Type 関数

関数名	LedApl_Set_Type
引数	LED_APL_TYPE type : 動作種別 alt_u8 data : 初期点灯データ
戻り値	なし
概要	LED 動作種別設定
処理詳細	動作種別と初期値の設定

## 5.3.1.4. LedApl\_Exec 関数

表 5.6 LedApl\_Exec 関数

関数名	LedApl_Exec
引数	なし
戻り値	なし
概要	LED アプリの実行
処理詳細	動作種別、もしくは任意点灯の値変化時に以下を実行 <ul style="list-style-type: none"> <li>・現在の設定値で LED 設定</li> <li>・全点灯/全消灯/任意点灯以外のときタイマー設定</li> <li>・現在の設定値に設定データを保存</li> <li>・動作種別に変化がない場合、タイマー動作停止</li> <li>・次の表示パターンの取得</li> <li>・取得データで LED 設定</li> <li>・タイマー設定</li> </ul>

## 5.3.1.5. LedApl\_Irq\_Timer 関数

表 5.7 LedApl\_Irq\_Timer 関数

関数名	LedApl_Irq_Timer
引数	なし
戻り値	なし
概要	LED アプリのタイマー割り込み
処理詳細	全点灯/全消灯/任意点灯以外で、タイマー動作中状態で以下を実行 <ul style="list-style-type: none"> <li>・タイマーCNT が 0 の場合にタイマー停止状態の設定</li> </ul> 上記以外で、タイマーCNT 値の減算

## 5.3.2. static 関数

## 5.3.2.1. led\_apl\_is\_onoff\_type 関数

表 5.8 led\_apl\_is\_onoff\_type 関数

関数名	led_apl_is_onoff_type
引数	LED_APL_TYPE type : 動作種別
戻り値	int true : あり / false : なし
概要	LED アプリのタイマー処理判定
処理詳細	動作種別によりタイマー処理が必要な場合 true、 そうでない場合は false を戻り値に設定

## 5.3.2.2. led\_apl\_set\_timer 関数

表 5.9 led\_apl\_set\_timer 関数

関数名	led_apl_set_timer
引数	なし
戻り値	なし
概要	LED アプリのタイマー値設定
処理詳細	タイマーCNT 値を計算

## 5.3.2.3. led\_apl\_get\_pattarn 関数

表 5.10 led\_apl\_get\_pattarn 関数

関数名	led_apl_get_pattarn
引数	LED_APL_TYPE type : 動作種別 alt_u8 now_led_data : 現在出力中のパターン
戻り値	alt_u8 : 新規出力パターン
概要	LED アプリの表示情報取得
処理詳細	種別によって出力パターンを取得 全点滅/交互点滅 :現在のパターンを全反転 シフト :現在のパターンから1Bit 左シフト インクリメント :現在のパターンから加算 デクリメント :現在のパターンから減算 その他 :現在のパターンをそのまま取得

## 5.3.2.4. led\_apl\_chg\_type 関数

表 5.11 led\_apl\_chg\_type 関数

関数名	led_apl_chg_type
引数	なし
戻り値	なし
概要	SW 入力により LED 動作種別を変更
処理詳細	SW1 が入力されたら内部テーブルに従い動作種別を変更

## 6. 更新履歴

Ver.	更新日付	内容
1.0.0	2015/09/14	新規作成