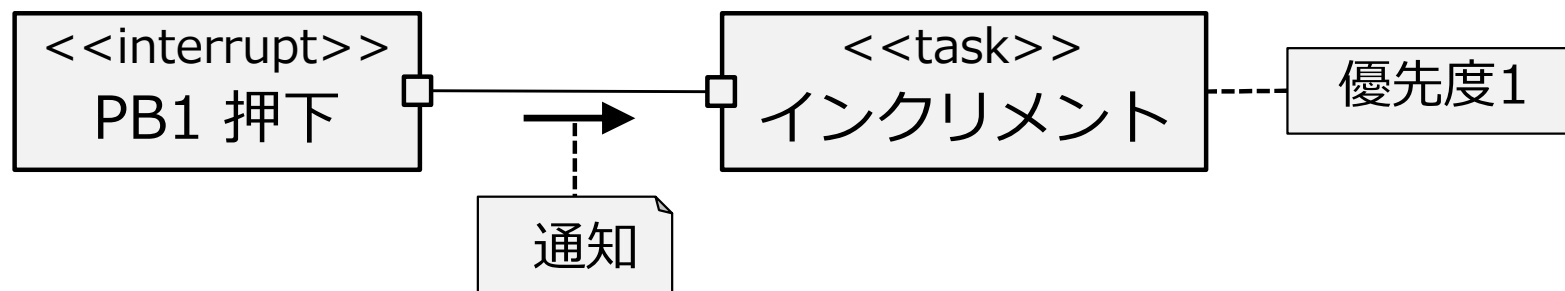
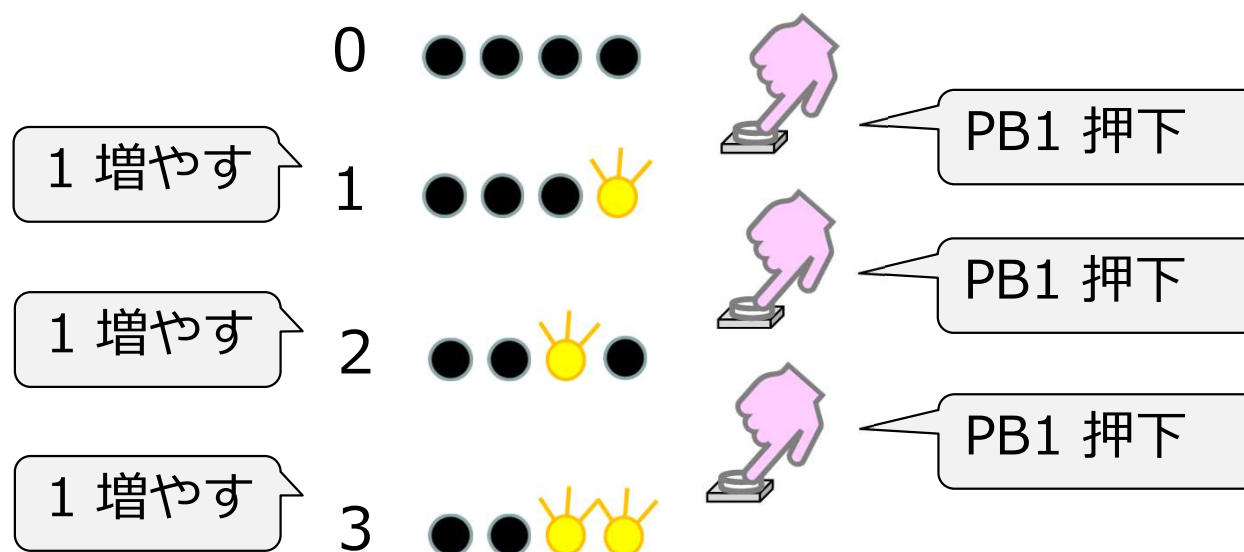


例題6 割り込みの通知

- プッシュボタン PB1 が押されたとき数を 1 増やし、二進数のパターンで LED を点灯する



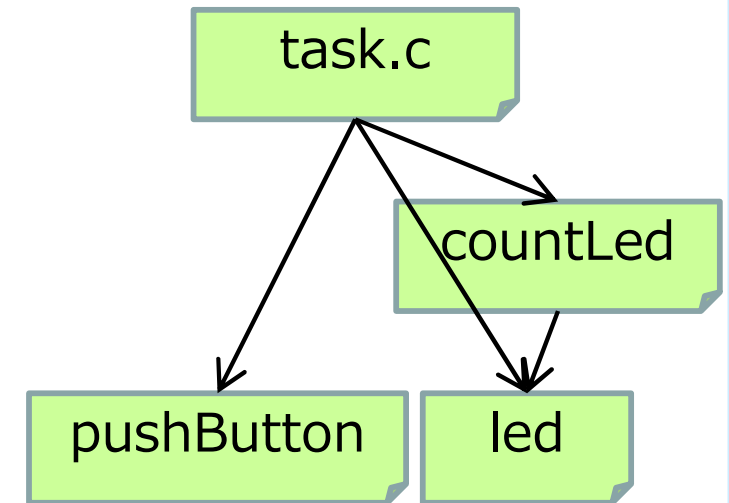
ファイルの構造

■ ファイル countLed

- 数を数え、二進数のパターンで LED を点灯する

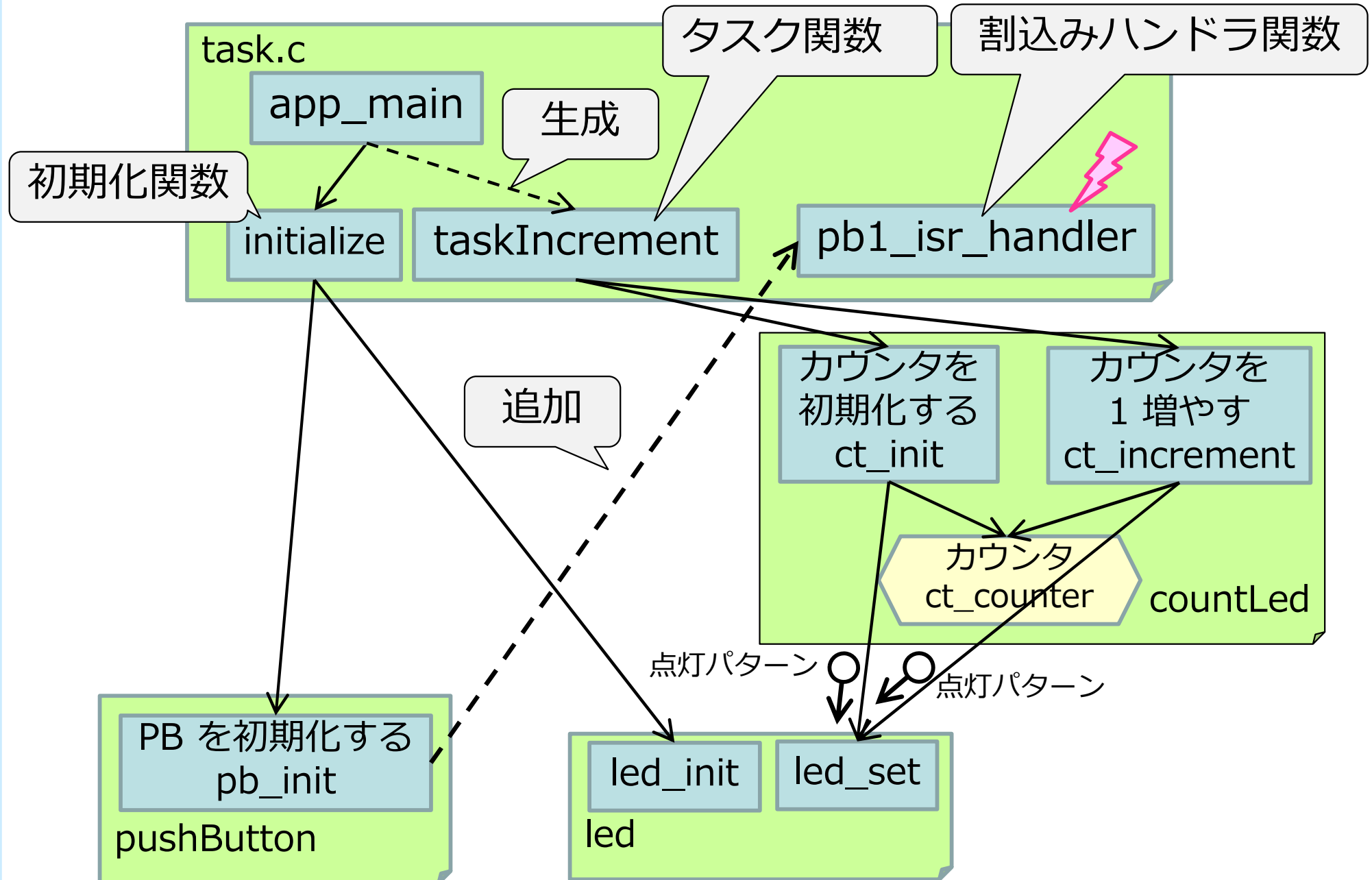
■ ファイル pushButton

- プッシュボタンを初期化し、割り込みハンドラ関数を追加する



ファイル	責務
task.c	システム動作 <ul style="list-style-type: none">・タスクの生成 (app_main 関数)・タスク関数、割り込みハンドラ関数・初期化関数
countLed	数を数えて LED に表示
led	LED 出力
pushButton	プッシュボタン

ファイルと関数の構造



使用する API

- タスクの生成（省略）
- タスクへの通知
 - xTaskNotifyWait（既出）
自タスクへの通知を待つ（ブロック状態に遷移）
 - **xTaskNotifyFromISR**
割り込み処理からタスクに通知する

割り込み処理（割り込みハンドラ関数）
でタスクへの通知を使用するときは
xTaskNotifyFromISR
を使用する

タスクへの通知

xTaskNotifyFromISR

タスクに通知する（割込みコンテキスト用）

■ 形式

```
 BaseType_t xTaskNotifyFromISR  
 (TaskHandle_t xTaskToNotify,  
  uint32_t ulValue,  
  eNotifyAction eAction,  
  BaseType_t *pxHigherPriorityTaskWoken )
```

■ 返却値

- pdPASS 通知は通知先のタスクに受理された
引数 eAction が eNoAction/eSetBit/eIncrement
のとき、返却値は常に pdPASS

パラメータ

xTaskNotifyFromISR

パラメータ		指定する内容
xTaskToNotify	タスクのハンドル	通知先のタスクのハンドル
ulValue	通知に関する値	RTOS によって値がどのように使われるかは、引数 eAction との関連で決まる
eAction	通知時のアクション	eNoAction 何もしない eSetBits ulValue のビットを設定する eIncrement 通知値を 1 増やす など
pxHigherPriorityTaskWoken	通知先のタスクの優先度の情報	pdTRUE 通知によってブロック状態が解除されるタスクの優先度が実行中のタスクよりも高いとき (コンテキストの切り替えを要求する必要がある) pdFALSE 上記以外

ファイル task.c のポイント

```
// --- Header files (system)
#include <stdio.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
// --- Header files (project)
#include "led.h"
#include "sevenSegmentLed.h"
#include "pushButton.h"
#include "countLed.h"
```

#include 、マクロ、変数、プロトタイプ

```
// --- macros
// task
#define STACK_DEPTH ((uint32_t) 4096)
#define PRIORITY_INCREMENT (tskIDLE_PRIORITY + 1)
// notify
#define CLEAR_NONE ((uint32_t) 0)
#define TICKS_TO_WAIT pdMS_TO_TICKS(1000 * 60 * 60) // 1 hour
#define VALUE_NONE ((uint32_t) 0)
```

通知を待つ時間
1 時間 PB1 を操作しないと
タイムアウトする

```
// --- data (static)
static TaskHandle_t taskHandleIncrement = NULL;
```

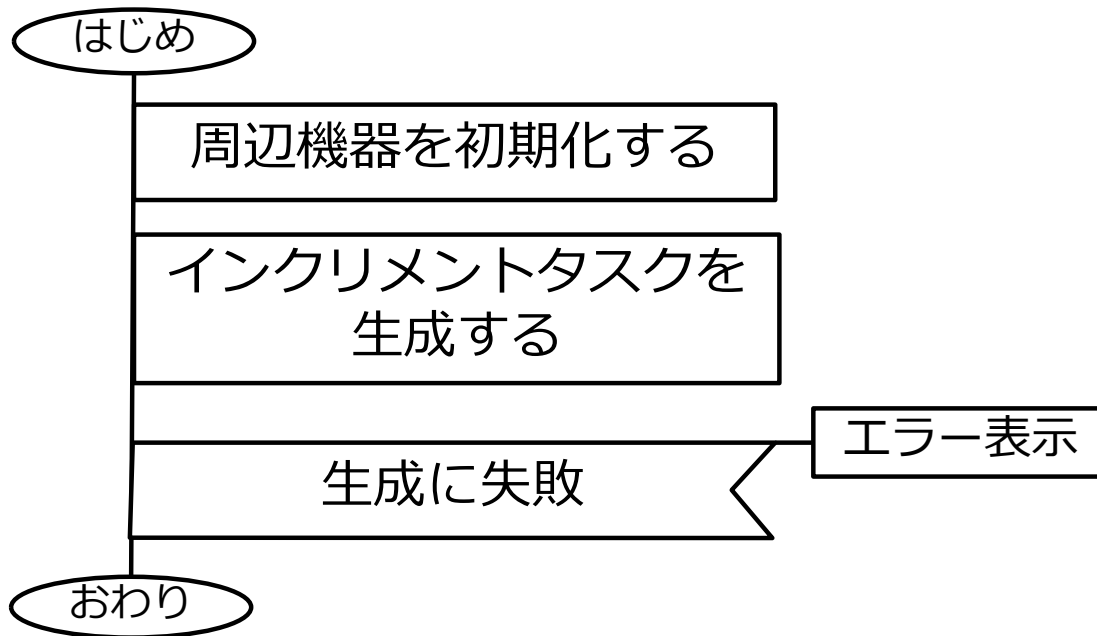
タスクのハンドルを代入する変数

```
// --- prototypes (static)
static void taskIncrement(void *arg);
static void initialize(void);
```

タスク関数 のプロトタイプ

app_main 関数

■ アルゴリズム



■ プログラム

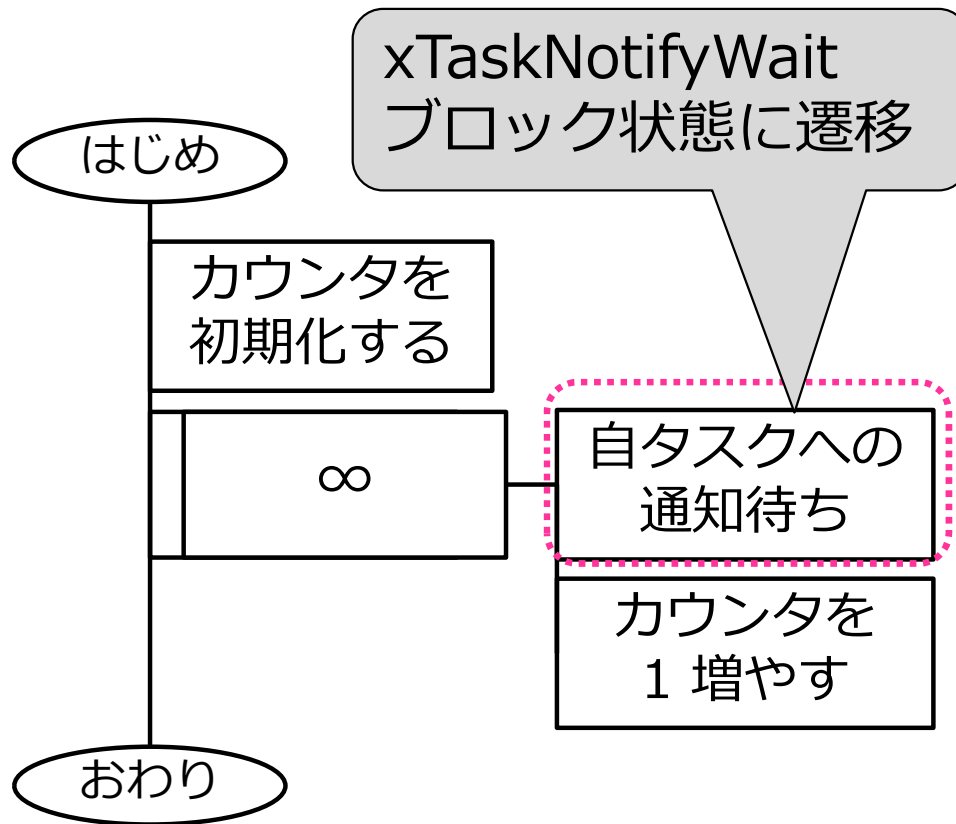
```
void app_main(void)
{
    BaseType_t  pass;

    // initialize devices
    initialize();
    // create task
    pass = xTaskCreate(
        &taskIncrement,
        "taskIncrement",
        STACK_DEPTH,
        NULL,
        PRIORITY_INCREMENT,
        &taskHandleIncrement
    );
    // 生成に失敗したとき（省略）

    return;
}
```


タスク関数 taskBlink

■ アルゴリズム



■ プログラム

```
static void taskIncrement(void *arg)
{
    // 動作確認 (省略)
    ct_init();
    for (;;) { // closed loop
        BaseType_t pd;

        pd = xTaskNotifyWait(
            CLEAR_NONE,
            CLEAR_NONE,
            NULL,
            TICKS_TO_WAIT
        );
        if (pd != pdPASS) { // 省略
        } else {
            ct_increment();
        }
    }
}
```

ファイル pushButton

■ pushButton.c

```
void pb_init(void)
{
```

```
    esp_rom_gpio_pad_select_gpio(PB1);
    ESP_ERROR_CHECK(gpio_set_direction(PB1, GPIO_MODE_INPUT));
    ESP_ERROR_CHECK(gpio_pullup_dis(PB1));
    ESP_ERROR_CHECK(gpio_pulldown_en(PB1));
    // PB2 のポートの初期化 (省略)
```

PB1 のポートの初期化

```
    // initialize interrupt
    ESP_ERROR_CHECK(gpio_set_intr_type(PB1, GPIO_INTR_POSEDGE));
    ESP_ERROR_CHECK(gpio_set_intr_type(PB2, GPIO_INTR_POSEDGE));
    ESP_ERROR_CHECK(gpio_install_isr_service(0));
    ESP_ERROR_CHECK(gpio_isr_handler_add(PB1, &pb1_isr_handler, NULL));
    ESP_ERROR_CHECK(gpio_isr_handler_add(PB2, &pb2_isr_handler, NULL));
    return;
```

ボタン押下で割り込み発生

割り込みハンドラ関数

割り込みハンドラ関数の追加

```
}
```

```
// --- prototypes (extern)
extern void pb_init(void);
// --- interrupt service routine defined task.c
extern void pb1_isr_handler(void *arg);
extern void pb2_isr_handler(void *arg);
```

pushButton.h

割込みハンドラ関数

■ アルゴリズム



■ プログラム

```
void pb1_isr_handler(void *arg)
{
    BaseType_t higherPriorityTaskWoken = pdFALSE;
    (void)xTaskNotifyFromISR(
        taskHandleIncrement,
        VALUE_NONE,
        eNoAction,
        &higherPriorityTaskWoken
    ),
    portYIELD_FROM_ISR(higherPriorityTaskWoken);
    return;
}
```

higherPriorityTaskWoken
が pdTRUE のとき
コンテキスト切り替えを要求