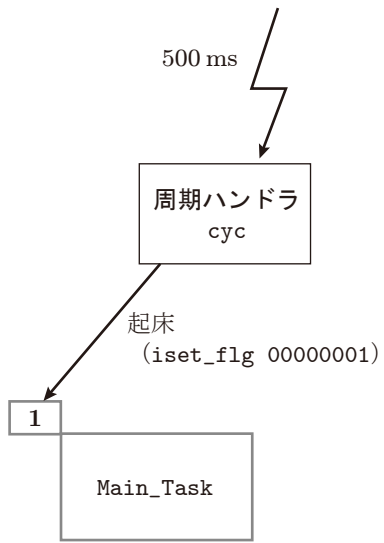
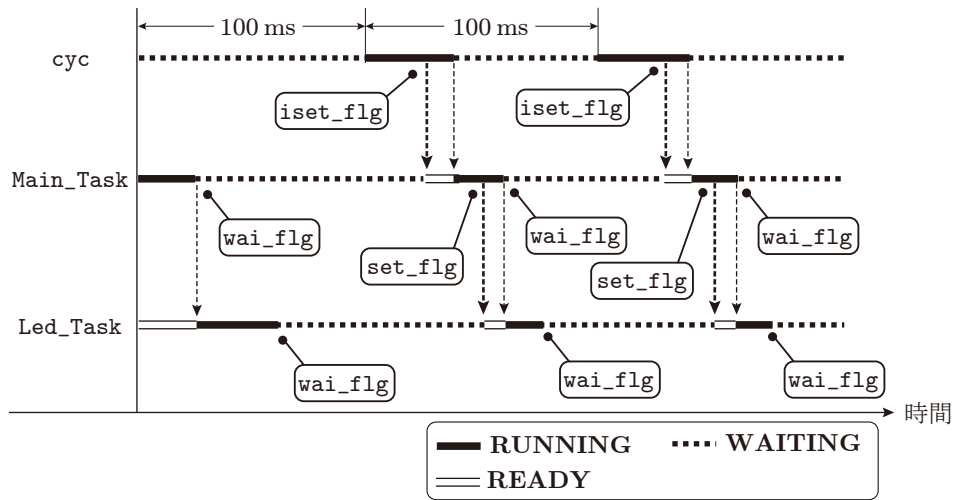


▼演習問題 6.1 : sample2 のオブジェクト関連図



▼演習問題 6.2 : sample3 のシーケンス図



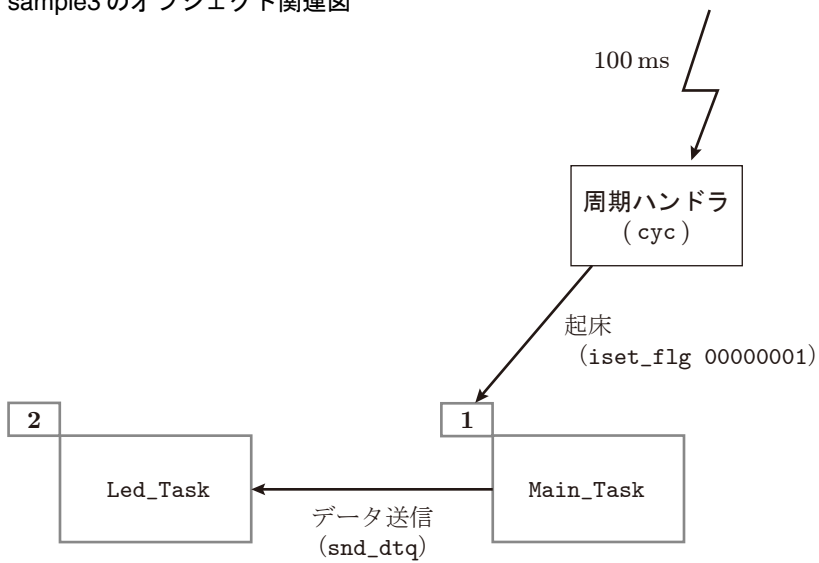
注 : cyc は周期ハンドラなので, 厳密には WAITING という状態は存在しません.
 は「起動されていない状態」という意味です.

▼演習問題 6.3：データキューを用いた sample3 のドキュメント

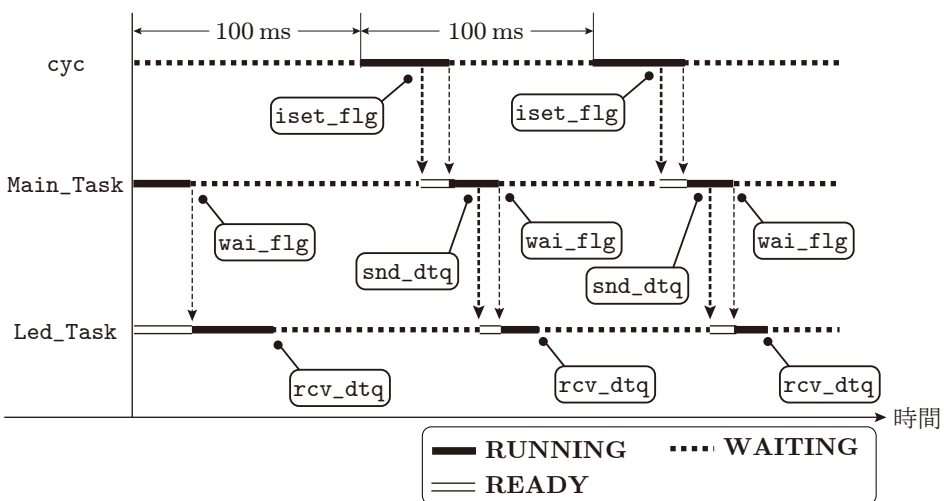
sample3 のオブジェクト一覧表

種類	名称	内容
プライオリティタスク	Main_Task	A / D 変換結果の読み取り
プライオリティタスク	Led_Task	青色 LED 表示
周期ハンドラ	cyc	100 ms の時間間隔作成
disp 有割込みハンドラ	intTim	システム時刻更新
イベントフラグ	フラグ 1	時間間隔通知
データキュー	キュー 1	A / D 変換結果通知

sample3 のオブジェクト関連図

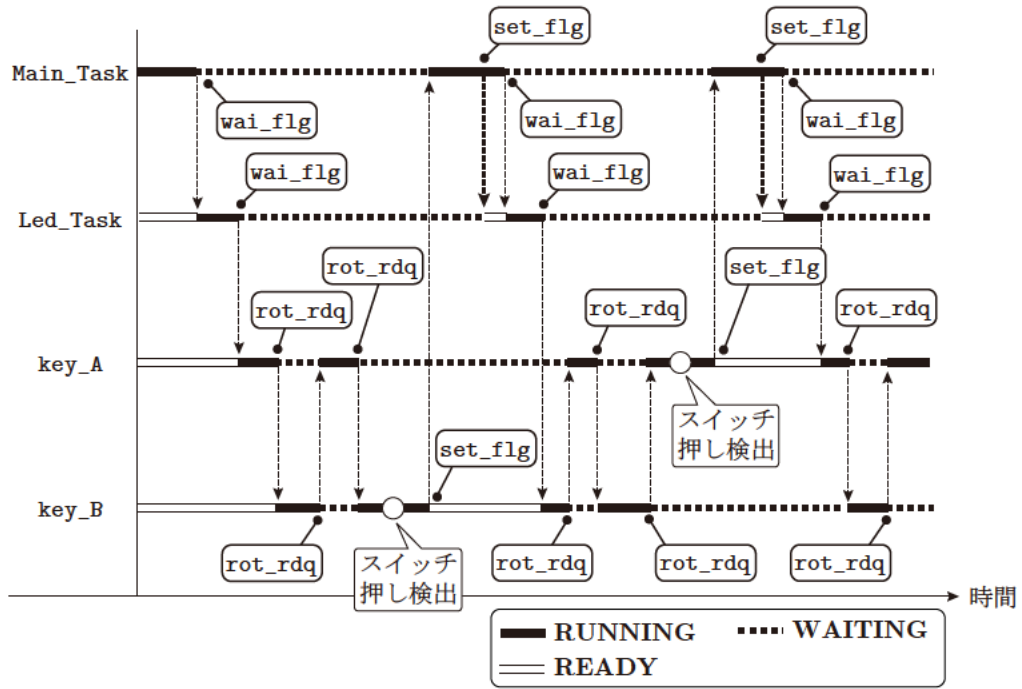


sample3 のシーケンス図



注：cyc は周期ハンドラなので、厳密には WAITING という状態は存在しません。
 は「起動されていない状態」という意味です。

▼演習問題 6.4 : sample4 のシーケンス図

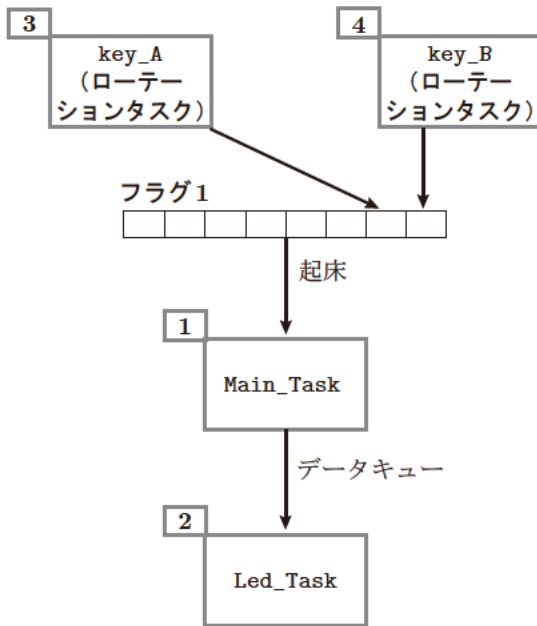


▼演習問題 6.5 : データキューを用いた sample4 のドキュメント

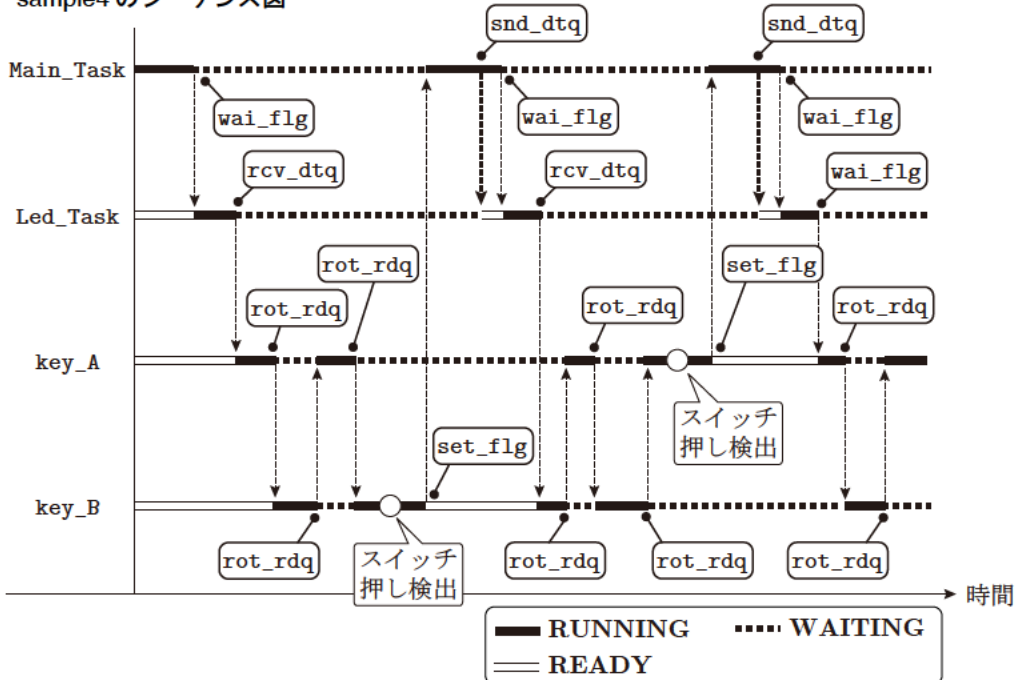
sample4のオブジェクト一覧表

種類	名称	内容
プライオリティタスク	Main_Task	ボリュームの値の読み取り
プライオリティタスク	Led_Task	青色LED表示
ローテーションタスク	key_A	上側プッシュスイッチ検出
ローテーションタスク	key_B	下側プッシュスイッチ検出
イベントフラグ	フラグ1	スイッチ押し検出通知
データキュー	キュー1	ボリュームの値を通知

sample4のオブジェクト関連図

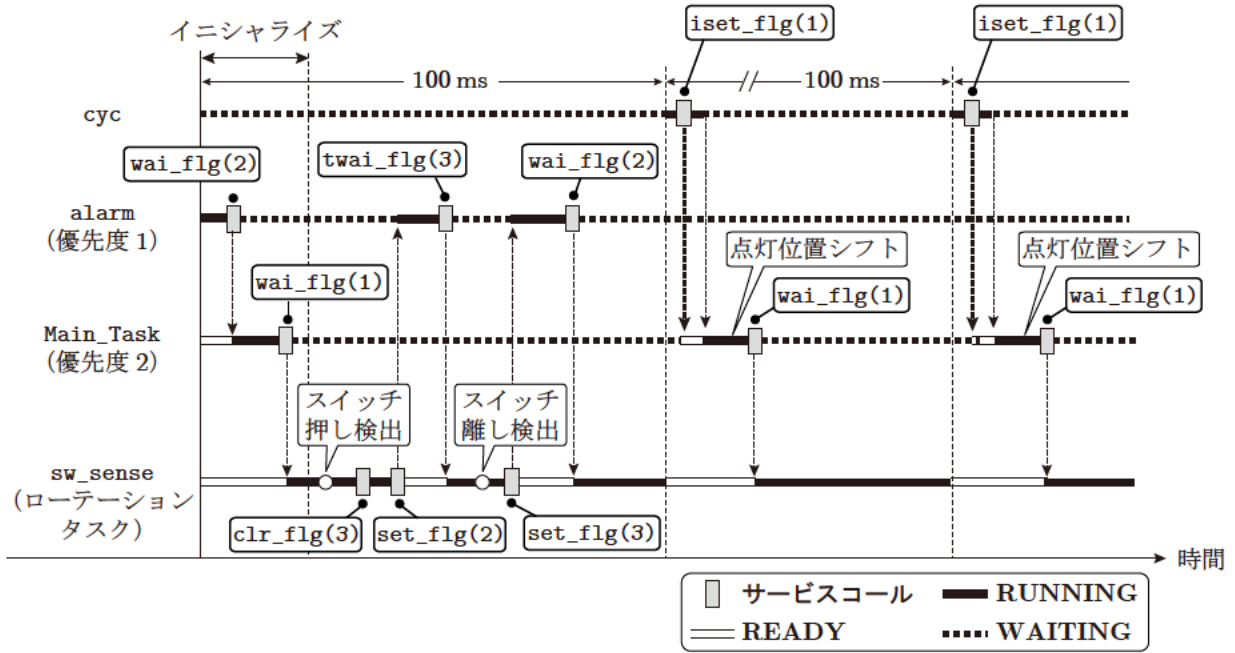


sample4のシーケンス図

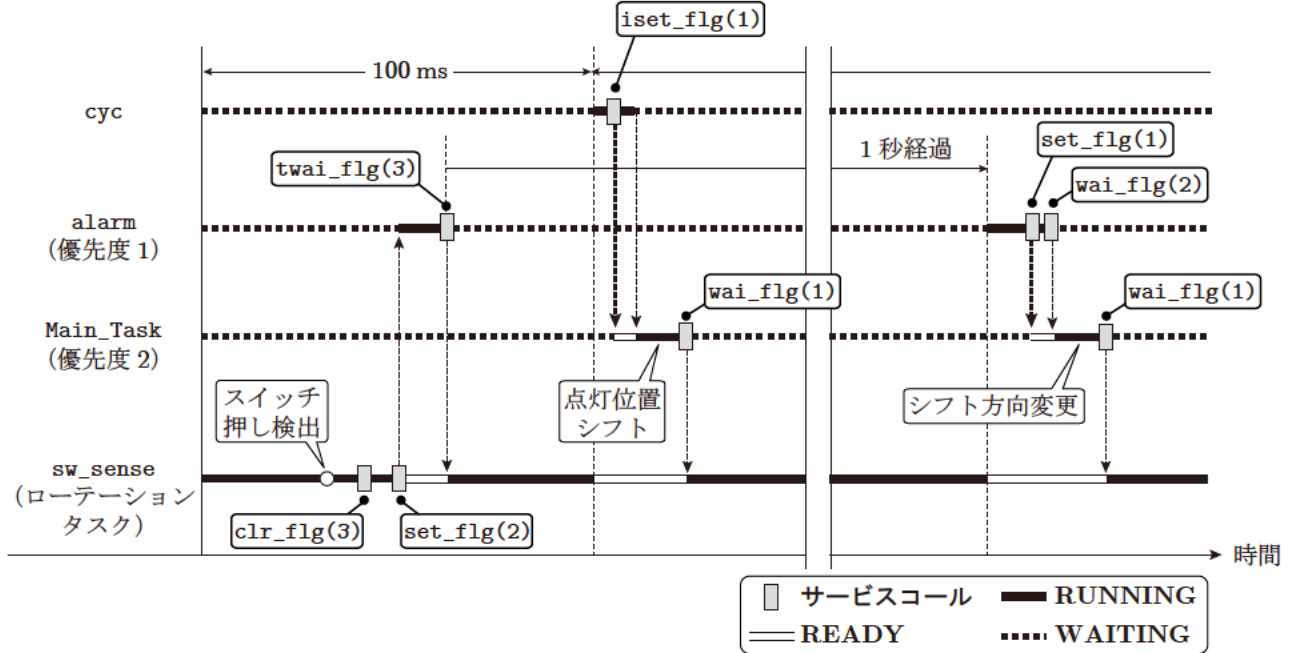


▼演習問題 6.6 : sample5 のシーケンス図

(1) 長押し不成立の場合



(2) 長押し成立の場合



▼演習問題 8.1

(1) イベントフラグのクリア指定がなかった場合の動作

ボリュームが低電圧から高電圧に変化する際はボリュームの値に追従して正常に7セグメントLEDの表示が変化するが、高電圧から低電圧に変化する際はボリュームの値に追従せず、7セグメントLEDの表示は変化しない。

上記のような結果になる原因は、イベントフラグのクリア指定を設定しなかったため、一度設定されたフラグがクリアされないからである。つまり、フラグのセット（意味的にはプラスの方向）には変化するが、フラグのクリア（意味的にはマイナスの方向）には変化しないからである。

(2) データ数が100個であった場合の動作

ボリュームの変化に対する7セグメントLEDの変更に遅延が生じる。

上記のような結果になる原因は、データキューのデータ数を100個に設定したため、`led_tsk`によるデータキューからのデータ読み込みに関係なく、100個のデータを書き込むことができるからである。つまり、ボリュームの最新のデータは、それ以前にデータキューに送信された100の古いデータが処理されなければ処理の対象とはならない。

`led_tsk`がデータキューからデータの受信を行う間隔は $5\text{ms} \times 2$ 桁の 10ms であるため、100個のデータの受信には $100 \times 10\text{ms}$ の1秒が必要である。結果、ボリュームの値を変更後、約1秒後に7セグメントLEDの表示が変化する。

▼演習問題 8.2

- LV0 interrupt ⇒ 未使用 (0バイト)
- LV1 interrupt ⇒ 18バイト + 2バイト + 4バイト = 24 (16進 18H) バイト
- LV2 interrupt ⇒ 未使用 (0バイト)

```
; -----  
;      Interrupt stack  
; -----  
@@intsB DSEG UNIT  
_intstack_top:  
_intstackLV0:  
    DS    18H      ; for LV1 interrupt  
_intstackLV1:  
_intstackLV2:
```

【解説】

まず, `intp0_hdr` 割込みハンドラは `disp` 無割込みハンドラであるため対象外である.

つぎに, LV0 interrupt (割込みレベル1) と LV2 interrupt (割込みレベル3) は, 対応した割込みハンドラが存在しないため, 確保する必要がない.

また, LV1 interrupt (割込みレベル2) のように同一レベルに複数の割込みハンドラが存在する場合, より多くスタック領域を必要とする割込みハンドラのみを考えればよい. 結果, この問題の場合は `intad_hdr` 割込みハンドラが使用するスタック領域だけを確保すればよい.

`intad_hdr` 割込みハンドラは,

- 割込みハンドラ自身の使用分 18 バイト
 - 必須分 2 バイト
 - 自身より優先順位の高い `disp` 有割込みハンドラの使用 0 バイト (なし)
 - `disp` 無割込みハンドラの使用 4 バイト (あり)
- となるため, LV0 interrupt のスタック領域は合計 24 バイトを確保すればよい.