

## 3. タスク間通信

3.1 同期

3.2 イベントと状態

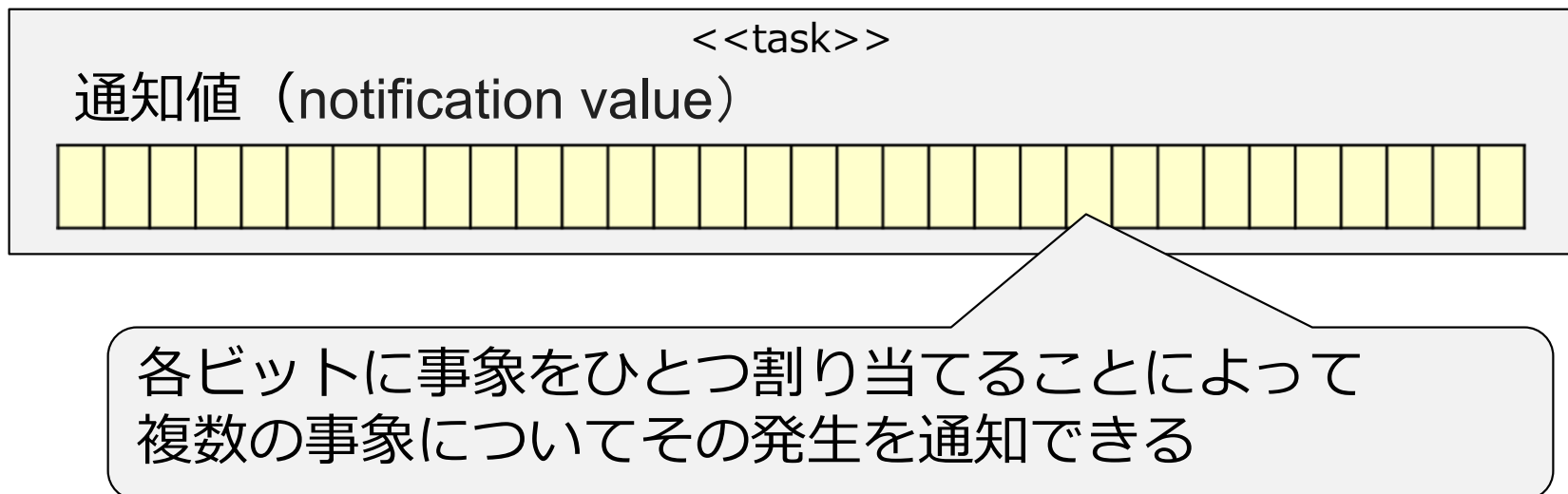
3.3 データの送受信

3.4 データ共有

# 3.1 同期

## ■ 通知値

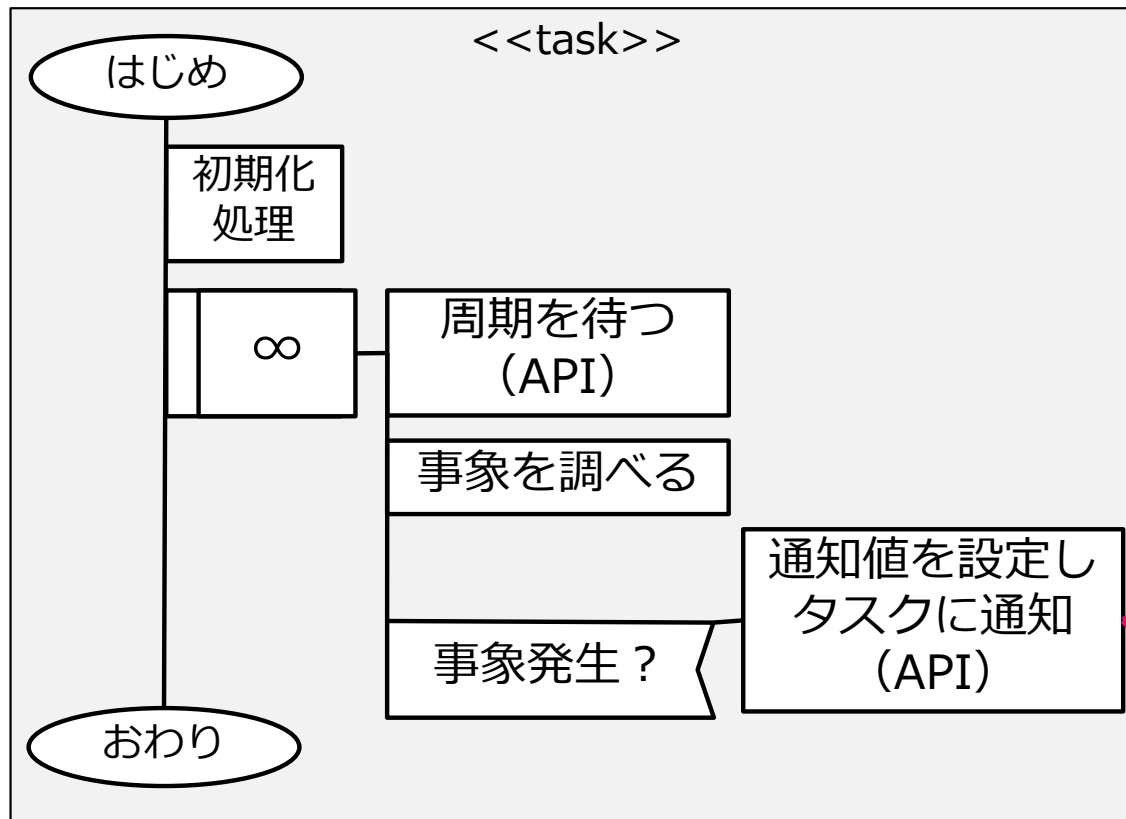
- FreeRTOS では、各タスクの管理する通知値（notification value）を使って複数の事象（イベント）の発生を通知できる
  - ◆ 事象発生 → 通知値の対応するビットを 1 に
  - ◆ 事象発生なし → 通知値の対応するビットを 0 に



# 通知値によるタスク間の同期の例

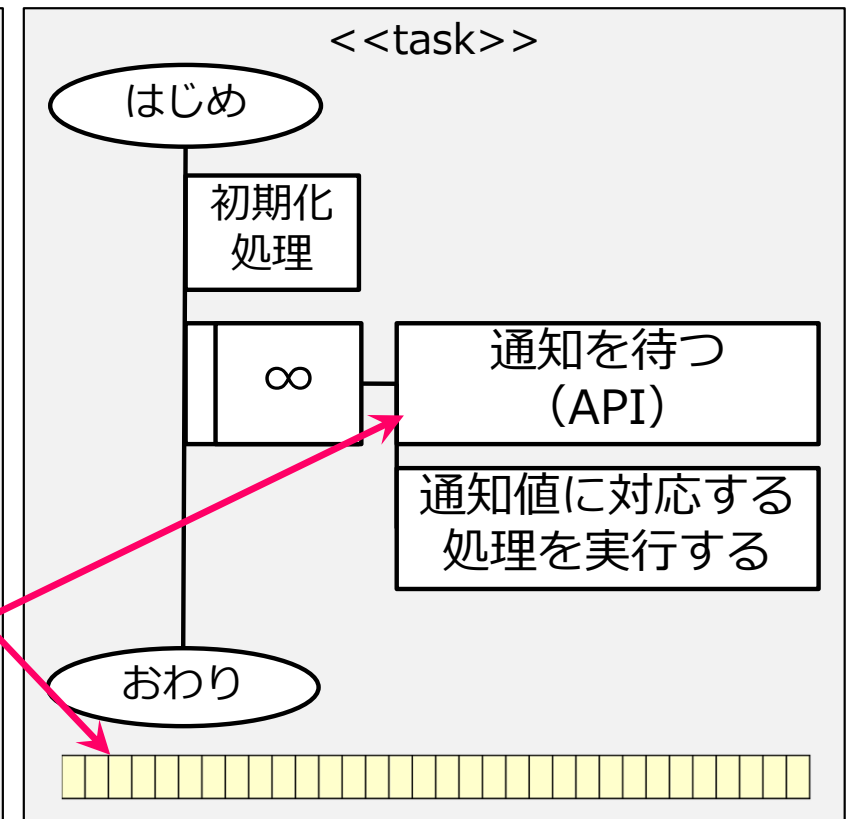
## ■ 事象を調べるタスク

- 事象が発生すると、通知値の対応ビットを 1 としタスクに通知する



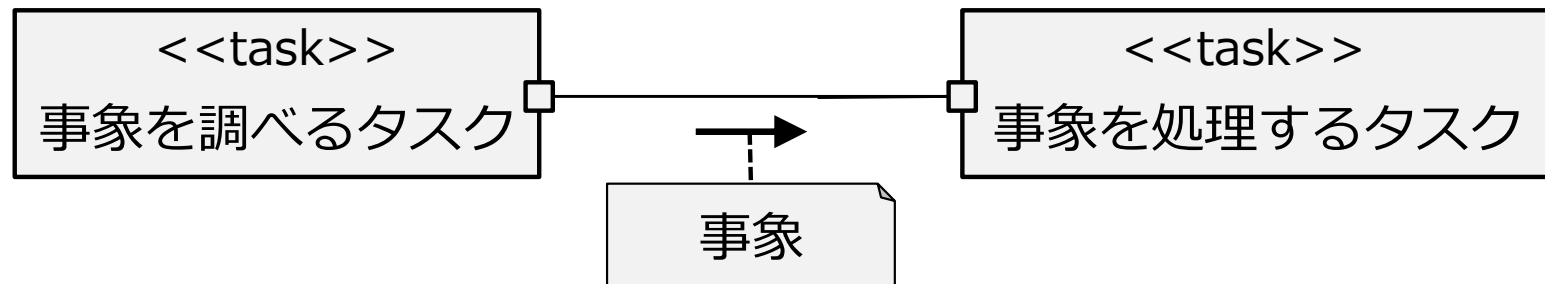
## ■ 事象を処理するタスク

- 通知値にセットされた内容の処理を実行する



# タスクの図面

## ■ 通知値による事象の通知

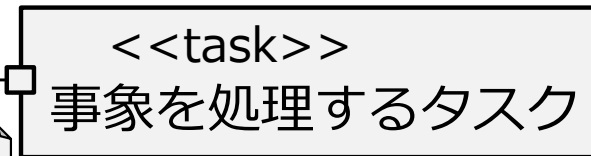


# 通知値による割込みとタスクの同期例

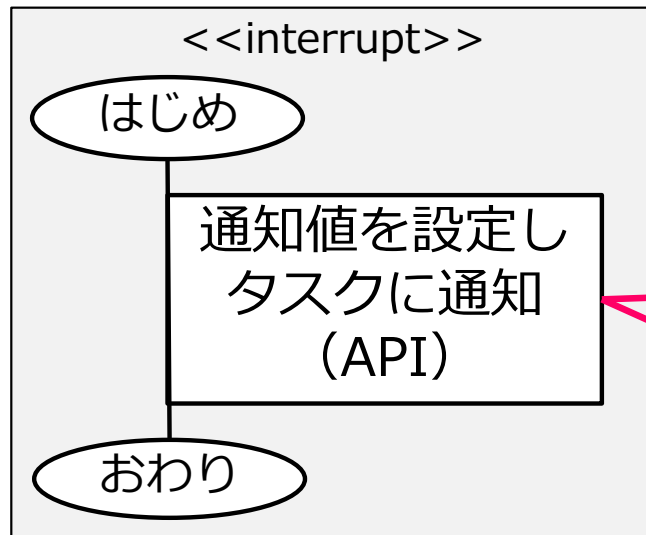
## ■ 割込み処理



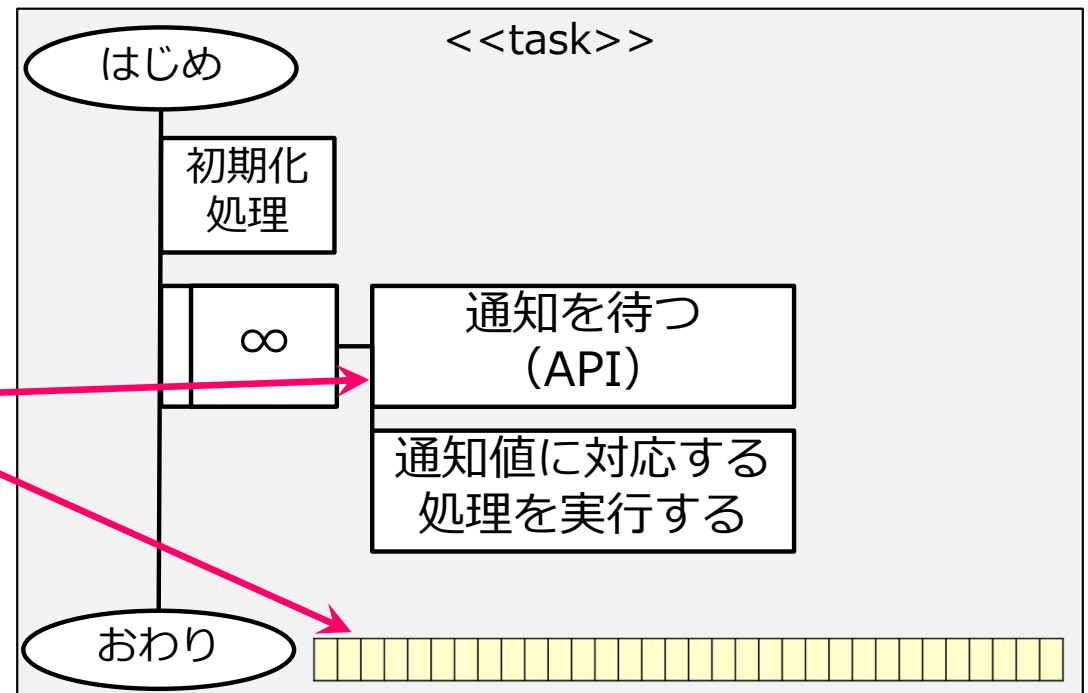
## ■ 事象を処理するタスク



- 通知値の対応ビットを1としタスクに通知する

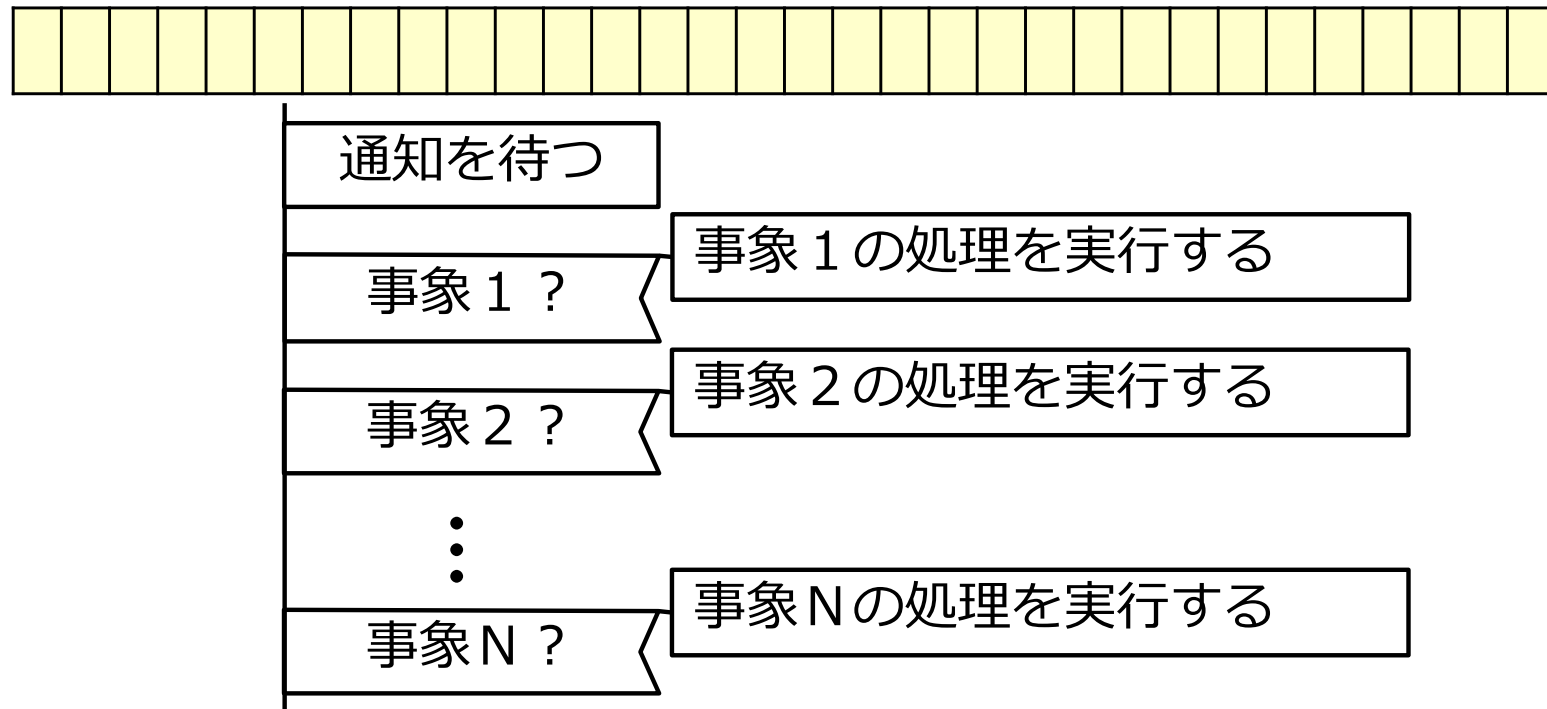


- 通知値にセットされた内容の処理を実行する



# 通知できる事象の数

- 通知値のビット数だけ事象を割り当てることができる



## ➤ 事象の定義（データ辞書）

◆ 事象 = [ 事象1 | 事象 2 | . . . | 事象 N ]

# 自タスクへの通知待ち

xTaskNotifyWait

## 自タスクへの通知を待つ

### ■ 形式

```
BaseType_t xTaskNotifyWait  
(uint32_t    ulBitsToClearOnEntry,  
 uint32_t    ulBitsToClearOnExit,  
 uint32_t    *pulNotificationValue,  
 TickType_t  xTicksToWait)
```

通知値を受け取る引数

### ■ 返却値

- pdPASS 通知を受理した
- pdFAIL 通知を受理していない

# パラメータ

## xTaskNotifyWait

パラメータ		指定する内容
ulBitsToClearOnEntry	待ち開始時にクリアするビット	通知値のビットのうち待ち開始時にクリアするビット (補足参照)
ulBitsToClearOnExit	待ち終了時にクリアするビット	通知値のビットのうち待ち終了時にクリアするビット (補足参照)
pulNotificationValue	待ち解除時の値を代入する領域	待ちが解除されたときの通知値を代入する領域 (呼出しもとに返す)
xTicksToWait	最大待ち時間	ブロック状態で通知を待つ 最大待ち時間 ティック数で指定

通知値を受け取る引数

補足) タスクは、内部に通知を受け取るためのデータ (通知値/notifications) を持っている  
xTaskNotifyWait は待ち開始/終了時に、通知値をビット単位でクリアできる



# タスクへの通知

## タスクに通知する

### ■ 形式

```
BaseType_t xTaskNotify  
(TaskHandle_t xTaskToNotify,  
uint32_t ulValue,  
eNotifyAction eAction )
```

通知値を設定する引数

### ■ 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

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 (eNoAction や eIncrement ではこの引数は使われない)
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

引数 eAction が eSetBits のとき通知値に設定するビット

引数 ulValue で指定されているビットを通知値に設定する