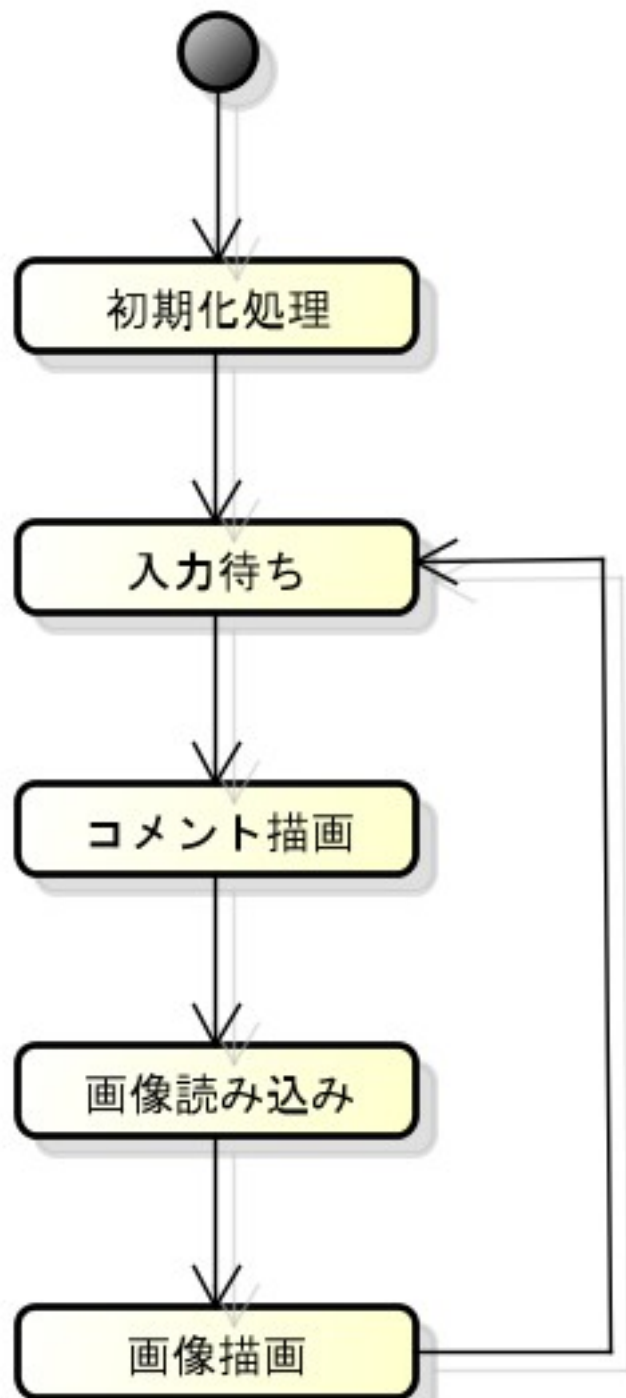


# システム内部の詳細な設計

- 画像をスライドするシステムをどう作るのか
- 画像の読み込みと描画やボタンなどからの入力はどうすればいいのか



安直に設計するとうなる。

これだと画像読み込みや画像描画のような重い処理をしている間は入力操作ができなくなる

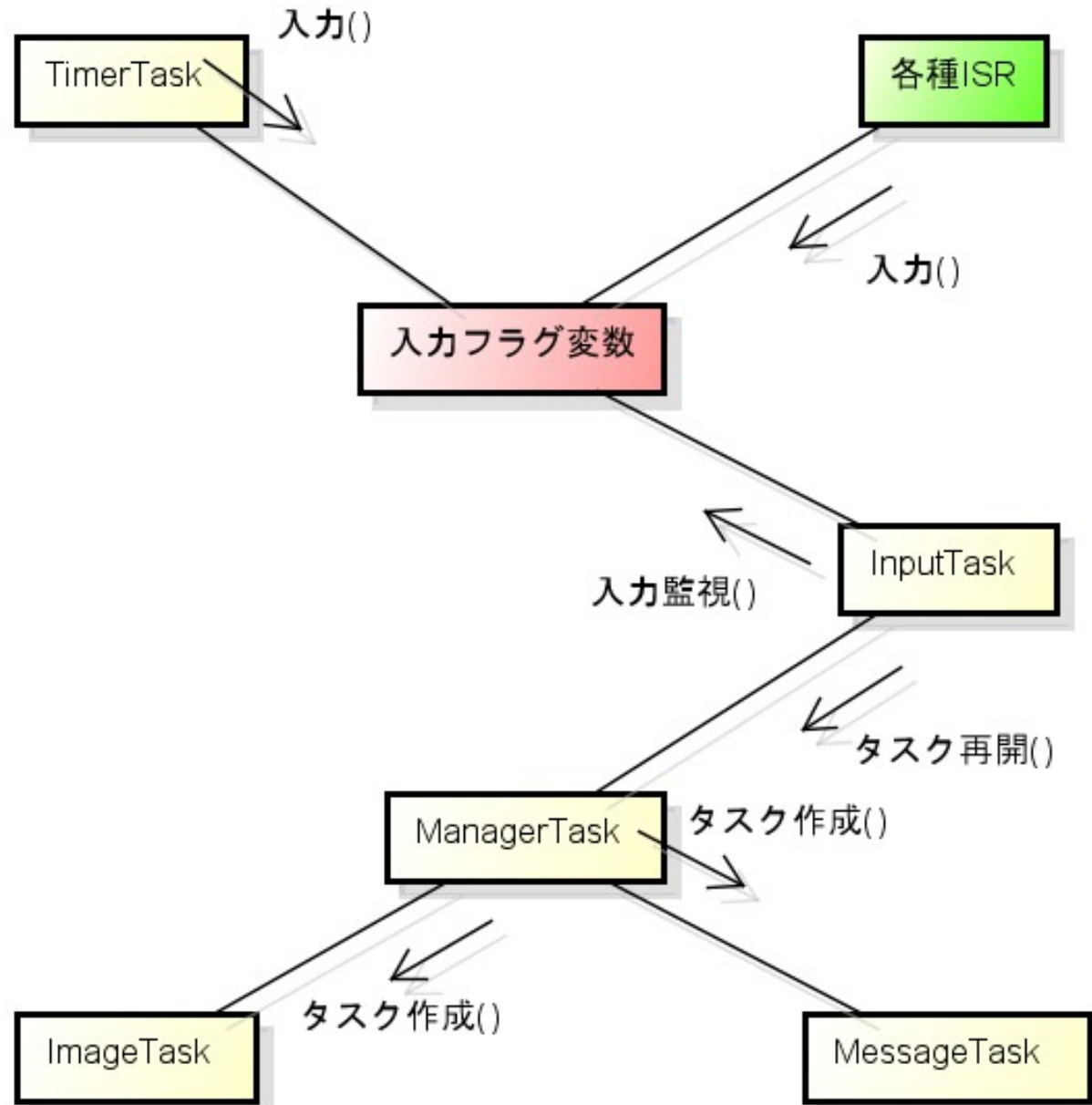
なぜならシングルタスクだから

だからマルチタスク化できれば

OSの導入

# OSを導入してマルチタスク化

↓ コミュニケーション図で動作の概要を説明



# このコミュニケーション図の意味

各種ISR (割り込みハンドラ)	スイッチやボタン, リモコンからの入力を行う 入力があれば入力フラグ変数を入力がされている状態にする
入力フラグ変数	入力されたかどうか管理するための変数
TimerTask	任意の時間間隔のタイマー入力をする
InputTask	入力フラグ変数を監視し, 入力があるかを判別 入力フラグ変数が入力状態であればManagerタスクを再開する
ManagerTask	各種入力から読み込む画像の指定や表示するコメントなどを変更し その後ImageTaskとMessageTaskを起動 二つのタスクを起動すれば自タスクを中断状態にする
ImageTask	画像の読み込みと画面への描画を行う この処理が完了すれば自タスクを終了する
MessegeTask	画像ファイルDBに書かれたコメント等をLCDや画面に描画する この処理が完了すれば自タスクを終了する

# マルチタスク化(タスク分割)のメリット

- 作業の並列化
  - シングルタスクだと重たい処理をするとシステム全体が停止する
  - マルチタスクでタスクごとに作業分担させるとこれらをさげることができる
    - ※タスクに適切な優先順位を与えた場合
- ルーチンの粒度を小さくできる
  - 作業内容を細かく分けてわかりやすくする
  - その結果メンテナンスしやすくなる

# 他技術的な話

- 割り込み
  - 今動いてる処理を中断して別な処理をさせる
- 割り込みハンドラ(ISR)
  - 割り込み処理で実際にすること
    - たとえば「ボタン押したらLEDを光らせる」など