

# 組み込みシステム基礎

Embedded System

## 第3回

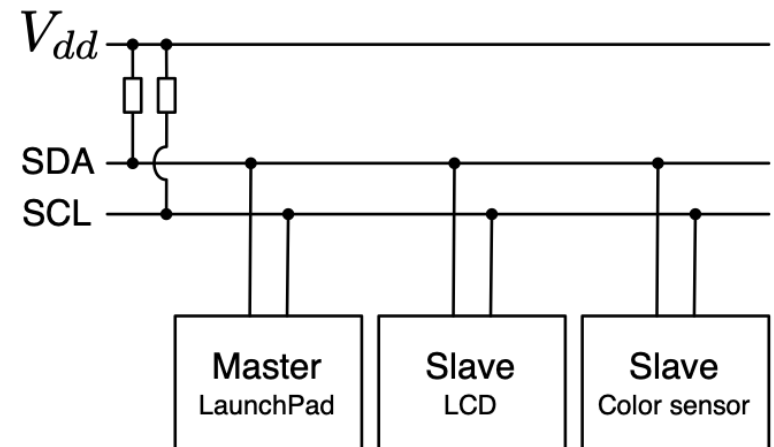
I2C, カラーセンサ, ロータリエンコーダ



# I<sup>2</sup>C (Inter-integrated Circuit)【復習】

Embedded System

- I squared C (アイ・スクエアド・シー) と呼ぶ
  - 他にもアイ・アイ・シー, アイツーシーという呼び方
  - I2C と書くこともある
- 組み込み機器でよく利用されるシリアル通信プロトコル
  - **少ない信号線で多くの機器を利用可能**
  - 通信速度は100kbps~400kbps
  - 機器内での短い距離の通信を想定
- 本日の課題
  - I2Cを利用したLCDへの出力



# i LCDとカラーセンサの同時利用

Embedded System

## ■ カラーセンサ

- フォトダイオードという素子で光を検知
- RGBCの4種類の値を取得



## ■ LCDとカラーセンサを同一のI2Cバス(I2C3)で利用

- カラーセンサから読み取った値をLCDに表示
  - 背景(例えば黒い下敷き)のセンシング時はLCDの表示は変化なし
  - 物体(カラートークン)が接近したときのみRGBを読みLCDに値を表示

## ■ 注意

- 壊れやすいので扱いは丁寧に！
- ピンヘッダから外れてしまった場合, 取り付け位置に注意

# i カラーセンサの割り込み

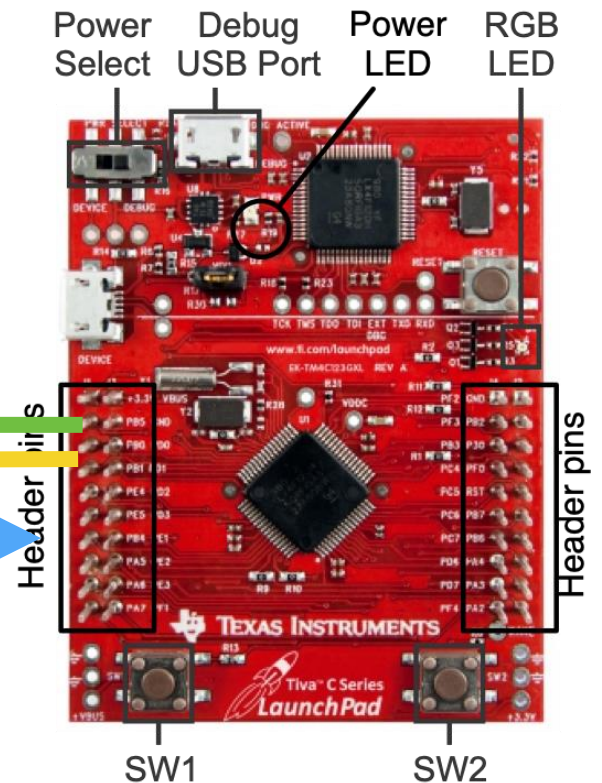
Embedded System



- ← Gainなどの設定
- ← 割り込みのしきい値 (Cの値) を設定

pull-upされているので  
普段はHIGH

PB0



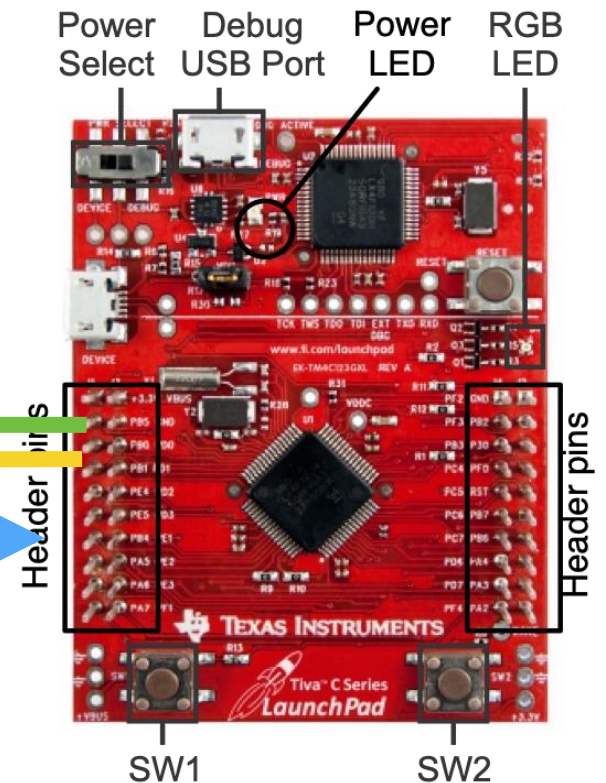
# i カラーセンサの割り込み

Embedded System



Cの値が  
上限閾値以上  
下限閾値以下  
になるとLOWに変化

PB0



## ■ 16bitの値

- AILTL:しきい値下限の下位8bit
- AILTH:しきい値下限の上位8bit
- AIHTL:しきい値上限の下位8bit
- AIHTH:しきい値上限の上位8bit

簡単のため  $(x \ll 8 \mid y)$  を  $x.y$  と表記

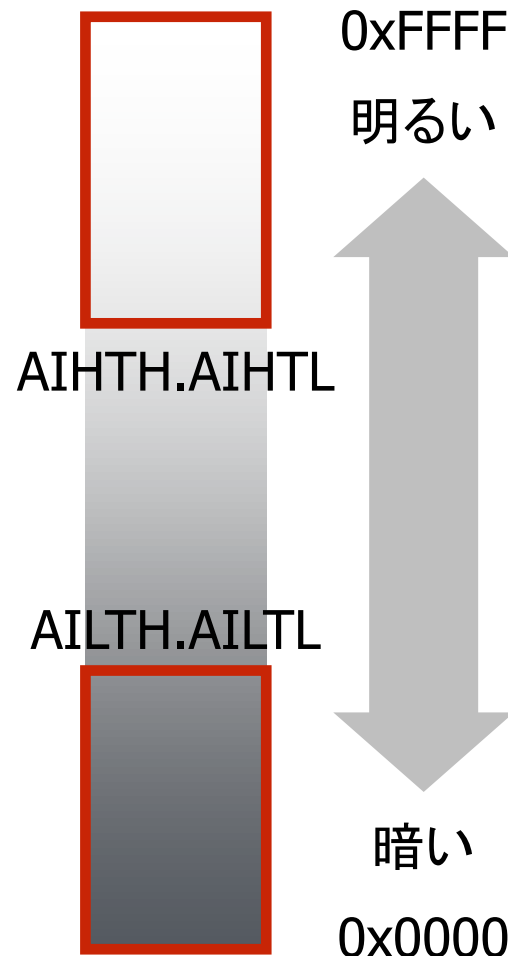
## ■ 割り込み発生範囲

- この範囲に入っていると割り込み処理が発生し続ける



: 割り込みの発生範囲

CDATAH.CDATAI



# i カラーセンサの割り込み

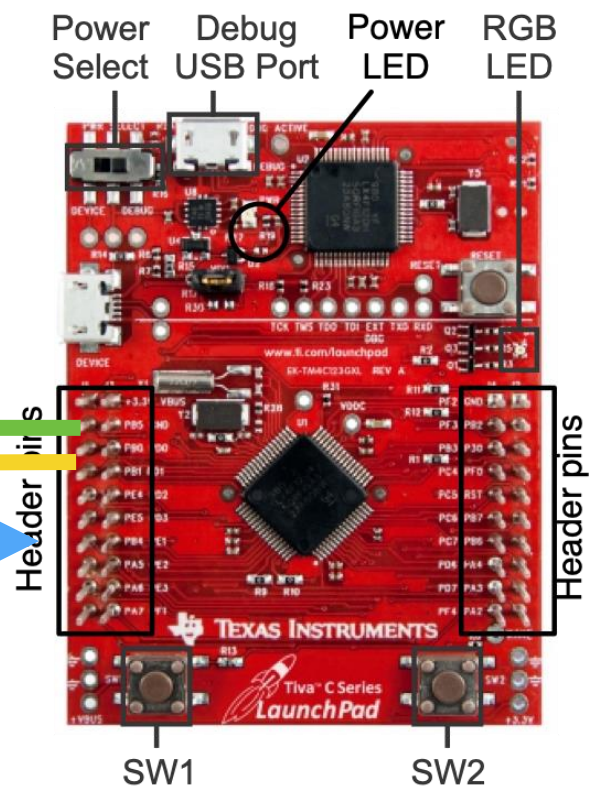
Embedded System



← データ読み出し依頼

PB0がLOWになると  
割り込みハンドラ実行

PB0





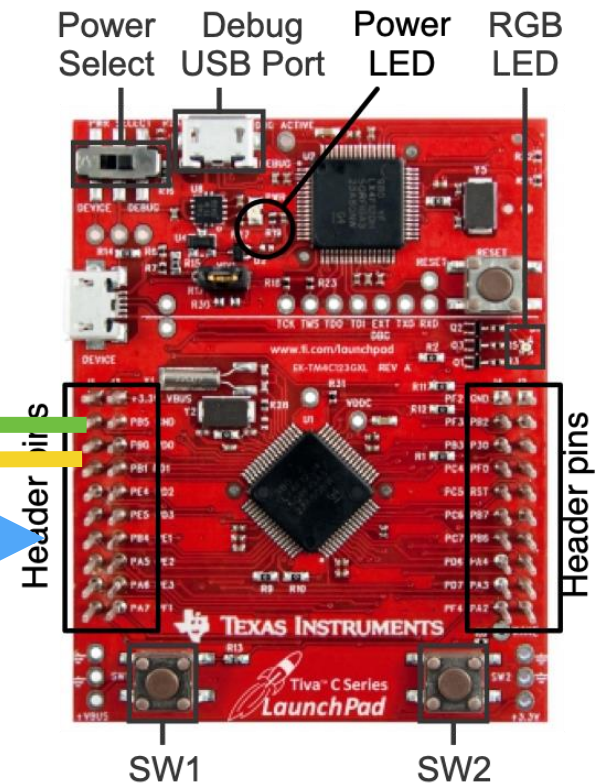
# i カラーセンサの割り込み

Embedded System



← データ受信

PB0

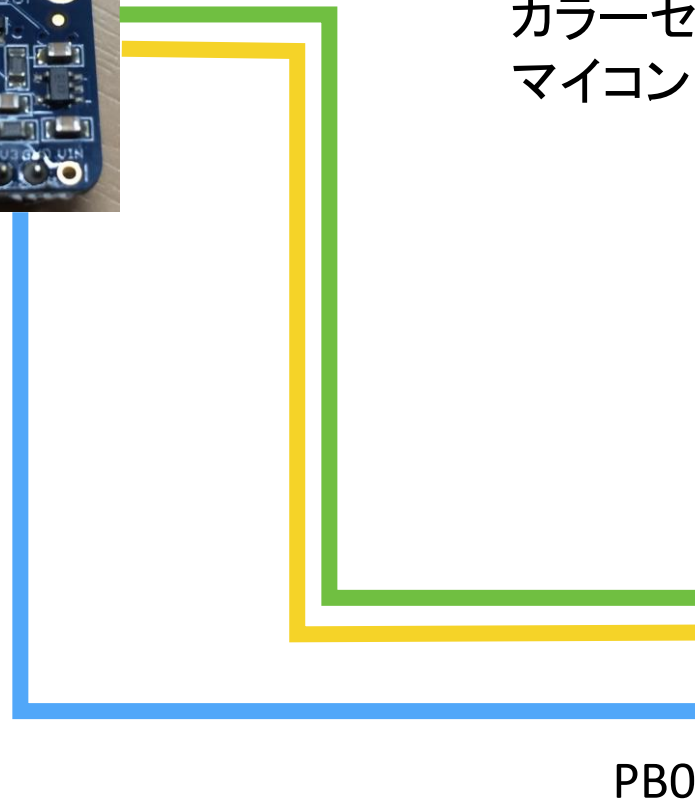




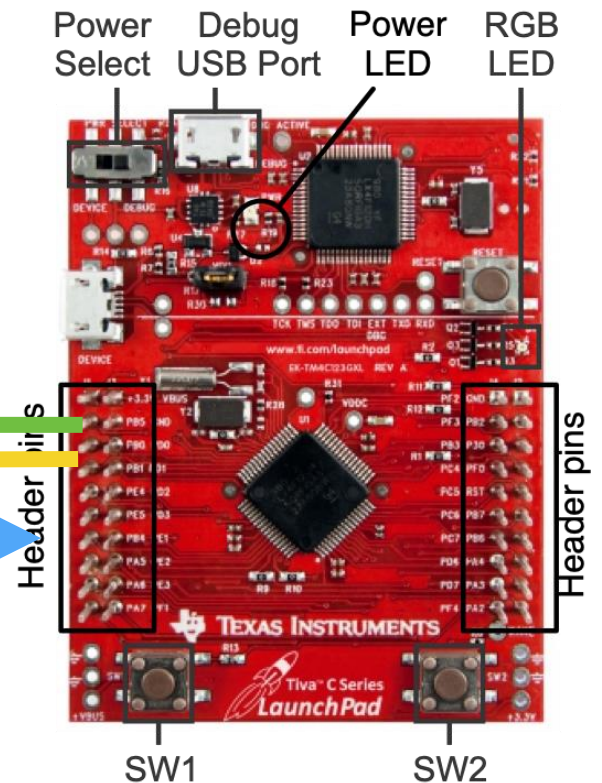
# i カラーセンサの割り込み

Embedded System

カラーセンサ・マイコンの双方で  
割り込みの終了処理  
カラーセンサ: `clearIntColorSensor()`  
マイコン: `GPIOIntClear()`



PB0



- 直交エンコーダインタフェースモジュール
  - QEI (Quadrature Encoder Interface)
- 回転の量をコード化するモジュール
  - 2相式: 2本のピン(PD6とPD7に接続)から位相のずれたパルスが発生
    - 1相式は方向が取れない
  - 回転方向(-1, 1)と回転角度(0~96)を読み取れる
    - 回転角度の情報はあまりあてにしないほうが良い
    - 0から1回転させても0に戻らない場合がある





# ロータリエンコーダの読み取り

Embedded System

- GPIO割り込み型(こちらを使用)
  - これまでやってきたPF4やPB0の割り込みと同様
    - PD6かPD7に対して割り込みを仕掛ける
  - 割り込みハンドラでQEIから値を読み取る
- Systick割り込み型(使用しない):
  - Systick割り込みハンドラ内でQEIから値を読み取る
  - 入力が最速でも1/32秒毎になるため遅れる

- 空の関数を埋める
  - 例: 割り込み初期化 (initInterruptPins())
- 意味を考えながらプログラムを作成
- lcd\_SB1602.h, lcd\_SB1602.c
  - 前回作成した関数を流用してLCDにカラーセンサの値等を表示
- ロータリエンコーダ
  - 回転方向を利用するとプログラムが簡単になるかも

