

# Nios II プログラミングその3 割り込み編

## そもそも割り込みってなに

- “プログラム実行中にボタンを押したとき”や”プログラムのバグで例外発生”などのとに実行中のプログラムに割り込んで別な処理をさせること。
- 割り込みが発生したときにする別な処理を割り込みハンドラ(ISR)っていう。

# 割り込みの種類

- ハードウェア割り込み

スイッチを押したりキーボードから入力があったりするときの割り込み。他には一定時間ごとに発生させるタイマー割り込みなど。

- ソフトウェア割り込み

プログラムが起因することで起こる割り込み。  
例外割り込みなど。

# Nios IIで割り込み処理を使う手順

1. 割り込みハンドラ用の関数を作る
2. (デバイスの割り込みで使うマスクを設定する)
3. (デバイスのEDGEをリセットする)
4. 割り込みハンドラを登録する

※割り込みハンドラでは最後にデバイスのEDGEをリセットしないと未来永劫延々と割り込みをしつづけるようになるみたいです。

## というわけで割り込みの使い方を

- 具体例に使うのは添付資料のparallel\_port\_test.cです。とりあえずこれと睨めっこしてください。
- ボタン押したらコンソールにメッセージが表示されるプログラムですよ。

# 割り込みの使い方具体例

```
1 #include <stdio.h>
2 #include "altera_up_avalon_parallel_port.h"
3 #include "sys/alt_irq.h"
4 #include "system.h"
5
6 /* Global variables */
7 alt_up_parallel_port_dev *green_LEDs_dev;
8 alt_up_parallel_port_dev *red_LEDs_dev;
9 alt_up_parallel_port_dev *hex3_hex0_dev;
10 alt_up_parallel_port_dev *slider_switch_dev;
11 alt_up_parallel_port_dev *push_button_dev;
12
13 //interrupt handler
14 void button_handler();
15
```

毎度おなじみ標準ライブラリ

パラレルポートのドライバ

割り込み関連

デバイスの名とかレジスタのアドレスの定数とか

使うデバイスの構造体ポインタ

割り込みハンドラ(割り込みあるとこれ実行する)

# 割り込みの使い方具体例

```
67 //
68 //init interrupt
69 //
70 alt_up_parallel_port_set_interrupt_mask(push_button_dev, 0xF);
71
72 //clear edge_capture resister
73 alt_up_parallel_port_clear_edge_capture(push_button_dev);
74
75 //resist interrupt handler
76 alt_irq_register(
77     PUSHBUTTONS_IRQ,
78     NULL,
79     (void *)button_handler);
80
```

alt\_up\_parallel\_port\_set\_interrupt\_mask()関数はマスクレジスタを操作して、どのボタンが割り込みできるかを設定している。  
ボタンに対応したビットを指定して。

# 割り込みの使い方具体例

```
67 //
68 //init interrupt
69 //
70 alt_up_parallel_port_set_interrupt_mask(push_button_dev, 0xF);
71
72 //clear edge_capture register
73 alt_up_parallel_port_clear_edge_capture(push_button_dev);
74
75 //register interrupt handler
76 alt_irq_register(
77     PUSHBUTTONS_IRQ,
78     NULL,
79     (void *)button_handler);
80
```

alt\_up\_parallel\_port\_clear\_edge\_capture()関数はエッジレジスタを操作してエッジトリガを全部Offにしている。エッジトリガが立っているときに割り込みが入るので、下げおかないと割り込み連発で暴走するかもしれない。



# 割り込みの使い方具体例

```
67 //
68 //init interrupt
69 //
70 alt_up_parallel_port_set_interrupt_mask(push_button_dev, 0xF);
71
72 //clear edge_capture register
73 alt_up_parallel_port_clear_edge_capture(push_button_dev);
74
75 //register interrupt handler
76 alt_irq_register(
77     PUSHBUTTONS_IRQ,
78     NULL,
79     (void *)button_handler);
80
```

alt\_irq\_register()関数で割り込みハンドラを登録する。

PUSHBUTTONS\_IRQはIRQ番号を表す定数。system.hで定義されてる。

IRQ番号自体はQsysで指定して回路作られているはず

NULL指定されてるものは割り込みハンドラに渡す引数のアドレス。使わないので今回はNULL

# 割り込みの使い方具体例

```
67 //
68 //init interrupt
69 //
70 alt_up_parallel_port_set_interrupt_mask(push_button_dev, 0xF);
71
72 //clear edge_capture register
73 alt_up_parallel_port_clear_edge_capture(push_button_dev);
74
75 //register interrupt handler
76 alt_irq_register(
77     PUSHBUTTONS_IRQ,
78     NULL,
79     (void *)button_handler);
80
```

alt\_irq\_register()関数で割り込みハンドラを登録する。

最後にbutton\_handlerが割り込みハンドラとして実行する関数のアドレス。  
関数ポインタ! コールバック関数!

# 割り込みの使い方具体例

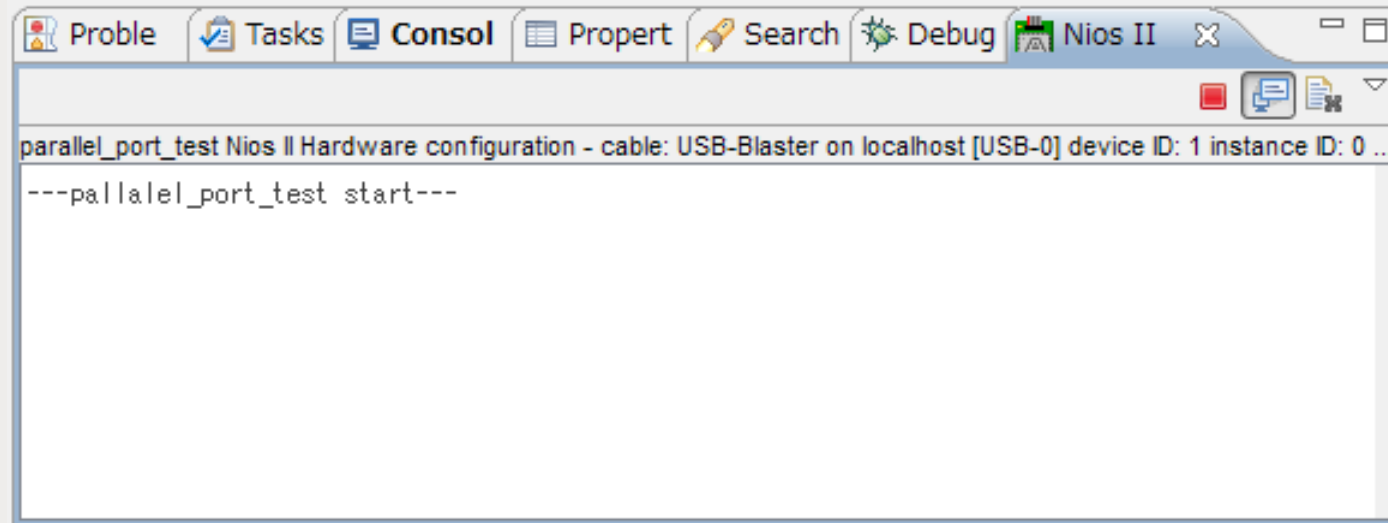
```
88 void button_handler()  
89 {  
90     int button_position;  
91  
92     button_position = alt_up_parallel_port_read_edge_capture(push_button_dev);  
93  
94     printf("interrupted, %x\n", button_position);  
95  
96     //clear EDGE resister  
97     alt_up_parallel_port_clear_edge_capture(push_button_dev);  
98 }
```

## 割り込みハンドラ

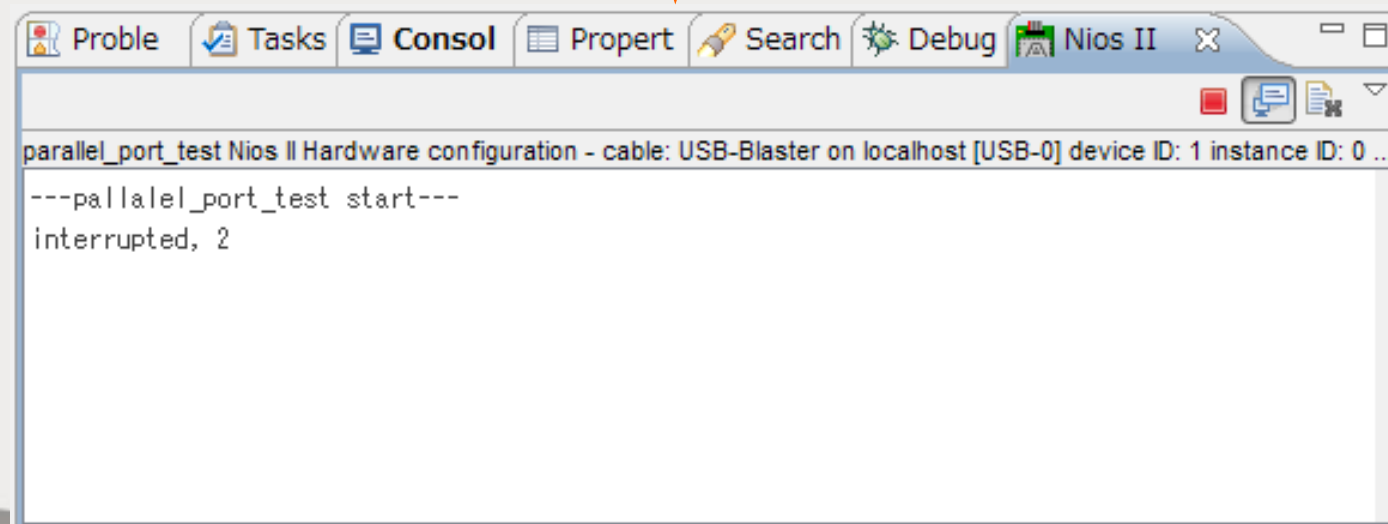
割り込みが入ったらこいつが実行される。  
割り込みが入ったボタンを表示させてる。

最後にエッジトリガOffにしないと暴走するかもしれないですよ…

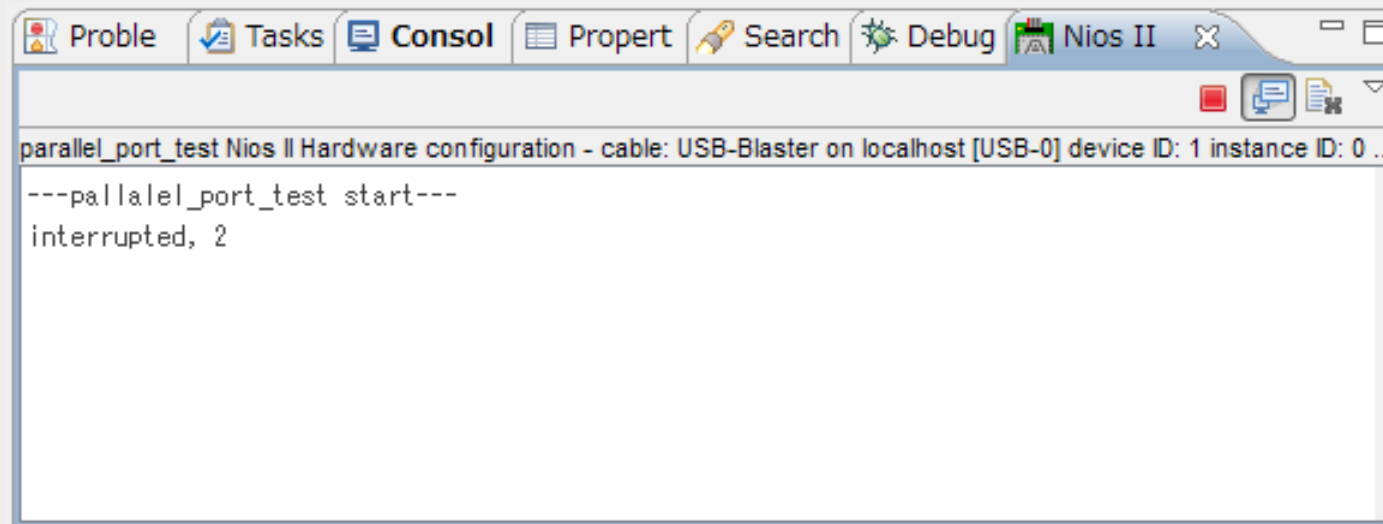
# 割り込みの使い方具体例(実行結果)



ボタン押した



# 割り込みの使い方具体例(実行結果)

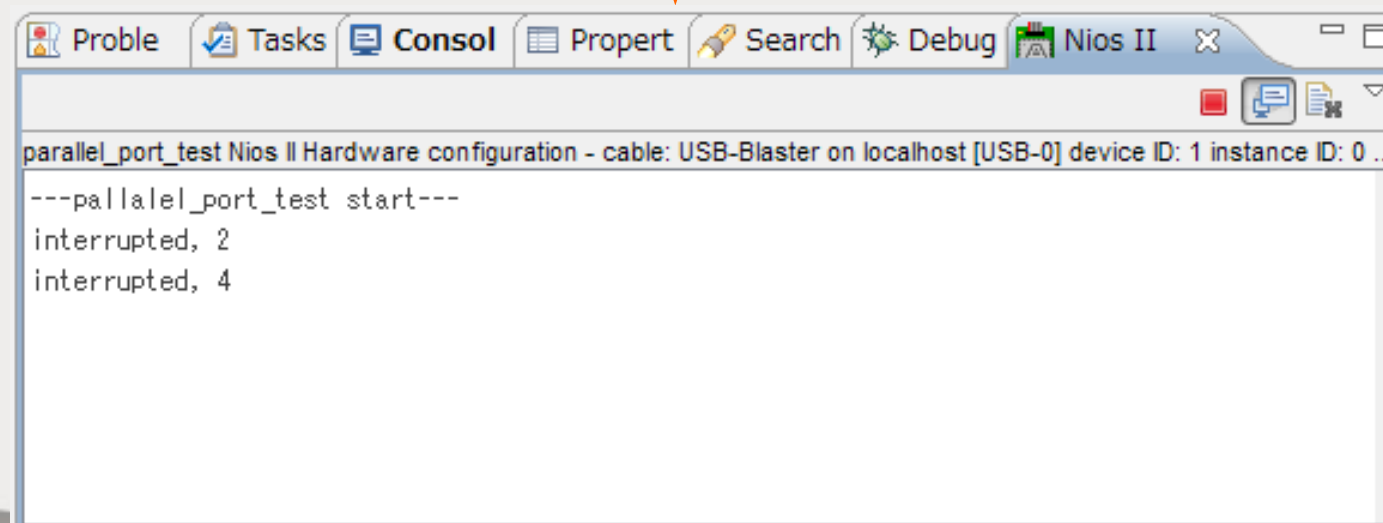


The screenshot shows the Nios II console window with the following text:

```
parallel_port_test Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0 ...  
---parallel_port_test start---  
interrupted, 2
```



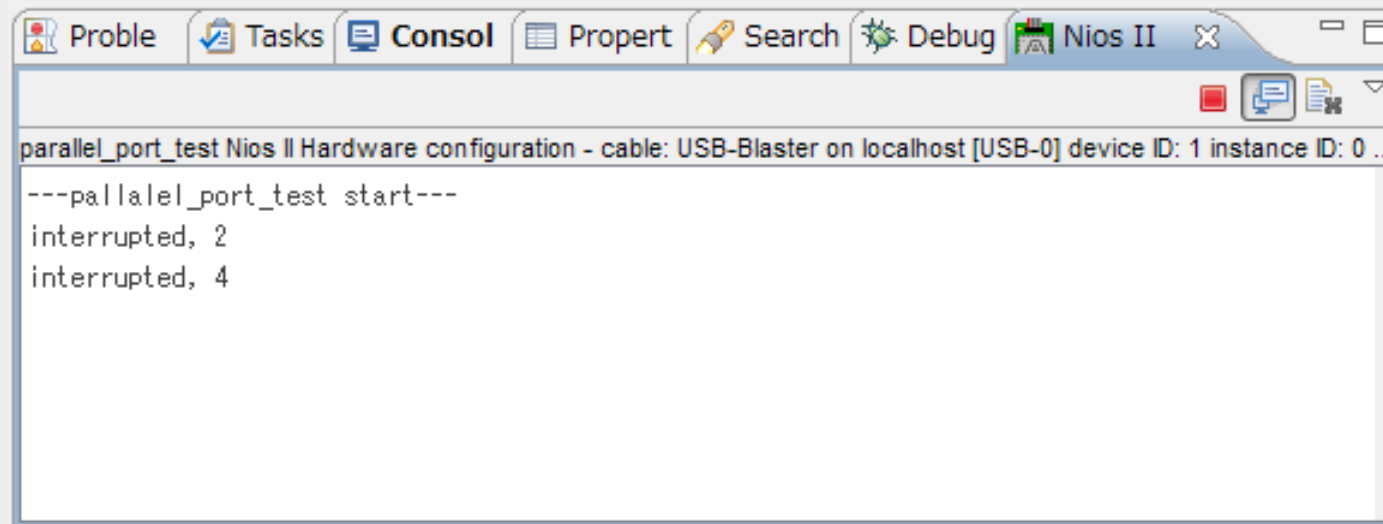
他のボタン押した



The screenshot shows the Nios II console window with the following text:

```
parallel_port_test Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0 ...  
---parallel_port_test start---  
interrupted, 2  
interrupted, 4
```

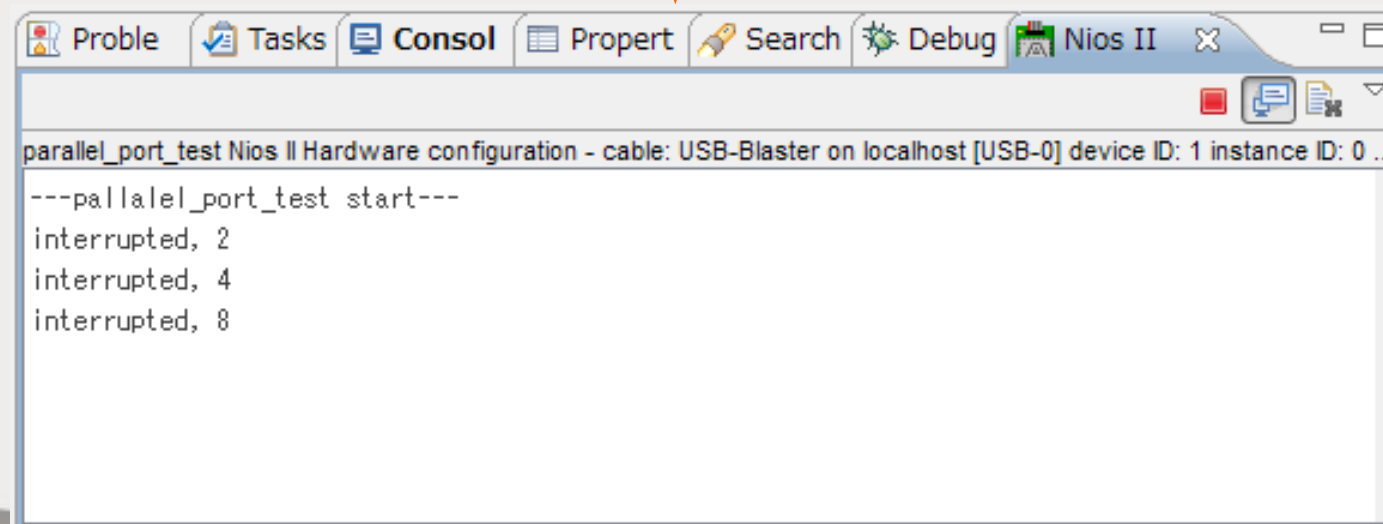
# 割り込みの使い方具体例(実行結果)



```
parallel_port_test Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0 ...  
---parallel_port_test start---  
interrupted, 2  
interrupted, 4
```



さらに他のボタン押した



```
parallel_port_test Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0 ...  
---parallel_port_test start---  
interrupted, 2  
interrupted, 4  
interrupted, 8
```

# 余談

- 割り込みハンドラで割り込み時にエッジトリガをOffにしないと…

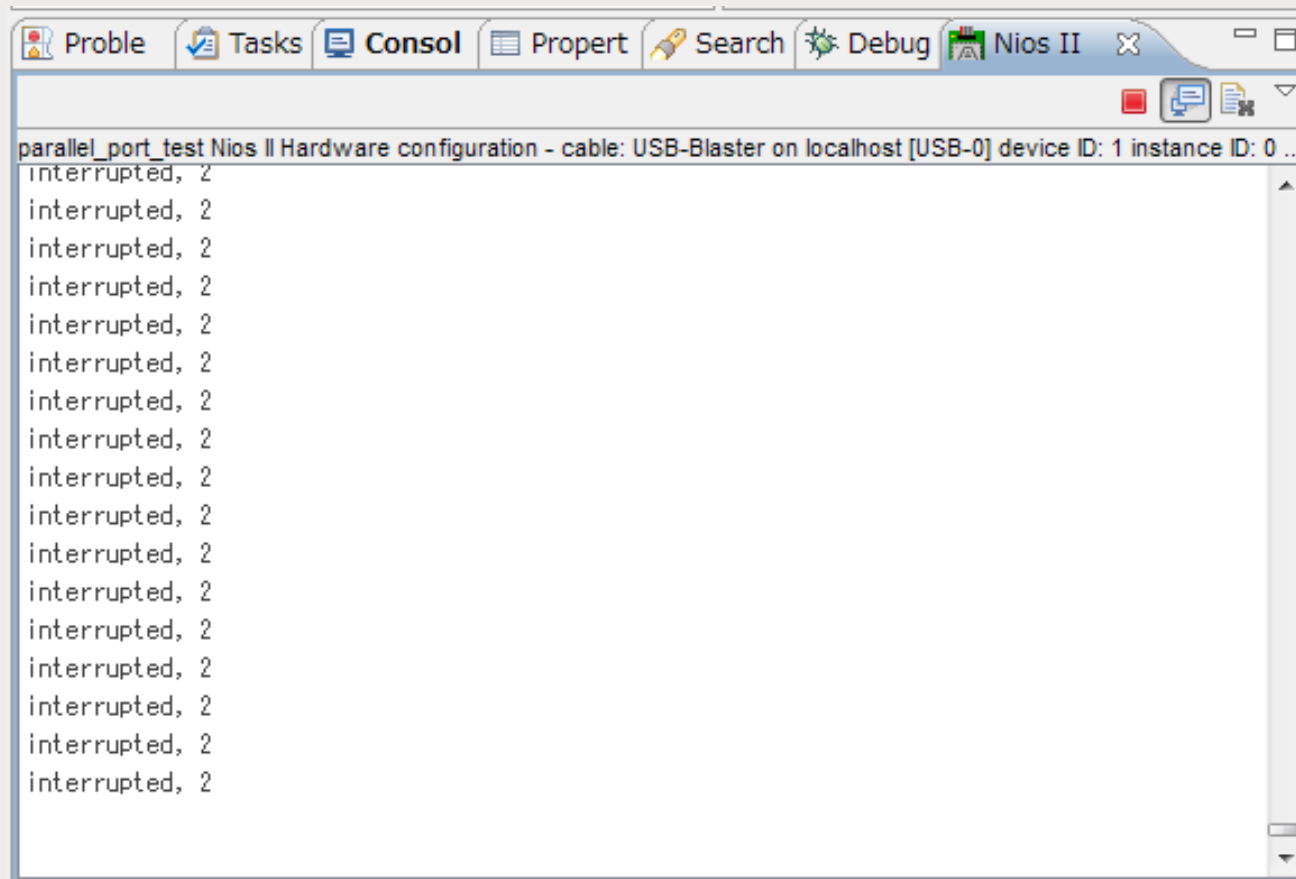
# 余談

```
88 void button_handler()  
89 {  
90     int button_position;  
91  
92     button_position = alt_up_parallel_port_read_edge_capture(push_button_dev);  
93  
94     printf("interrupted, %x\n", button_position);  
95  
96     //clear EDGE resister  
97     alt_up_parallel_port_clear_edge_capture(push_button_dev);  
98 }
```

じゃあ割り込みハンドラでエッジトリガをOffにしないで実行してみよう



# 余談



暴走して同じ割り込みハンドラをずっと呼び出しています。