

マイコンをはじめよう

割り込みを使おう

徳島大学大学院ソシオテクノサイエンス研究部
技術専門職員 辻 明典

連絡先：

770-8506 徳島市南常三島町2-1

TEL/FAX： 088-656-7485

E-mail: : a-tsuji@is.tokushima-u.ac.jp

割り込みを使おう

第8回

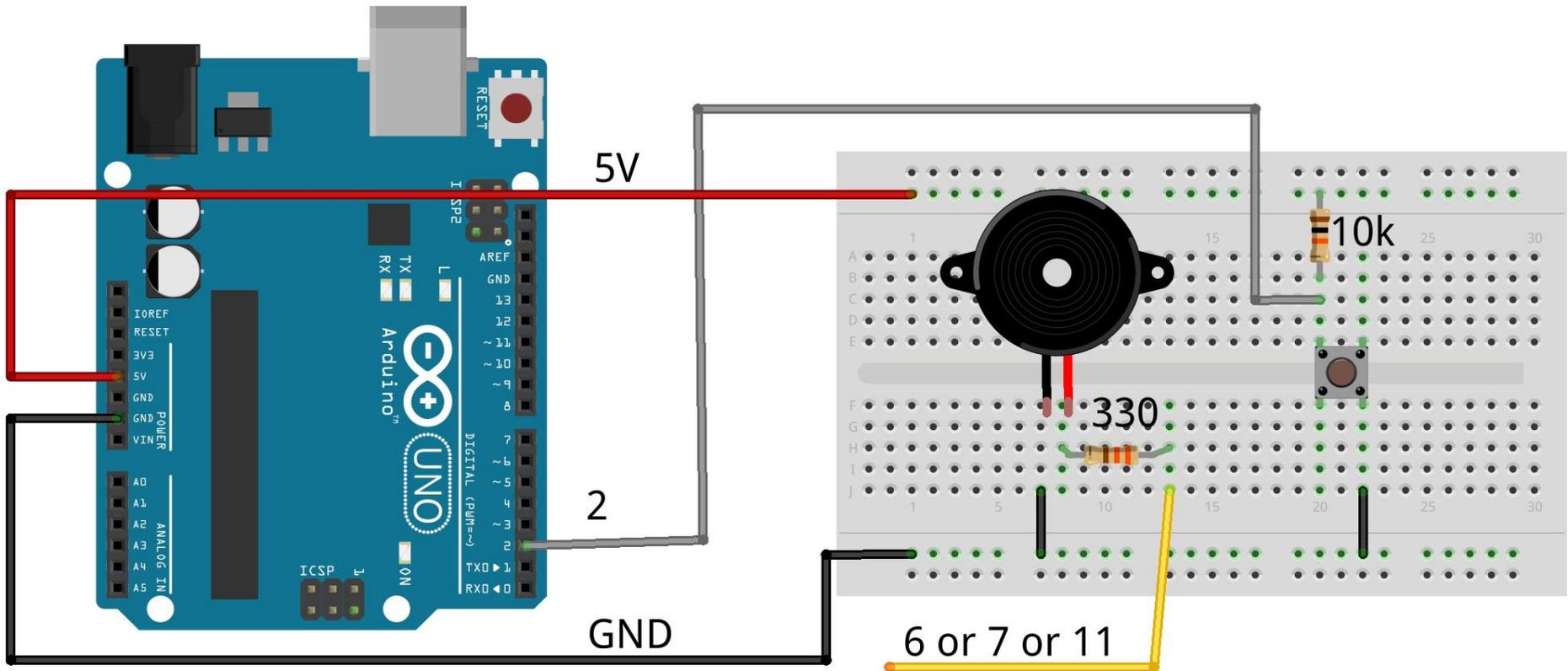
2013/9/14(Sat) 10:00—11:30

本日の予定

- 1 割り込みについて
- 2 タイマ割り込み
- 3 外部割り込み
- 4 割り込みでブザーを鳴らそう

本日の回路

割り込みスイッチとブザー

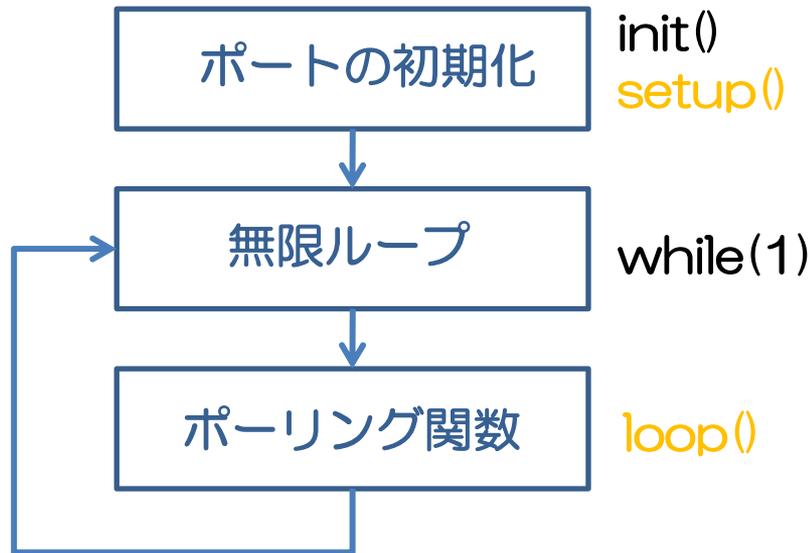


Made with Fritzing.org

1 割り込みについて

1.1 マイコンのプログラム

基本構成



ポーリング



割り込みが発生したときにのみ
呼ばれる関数

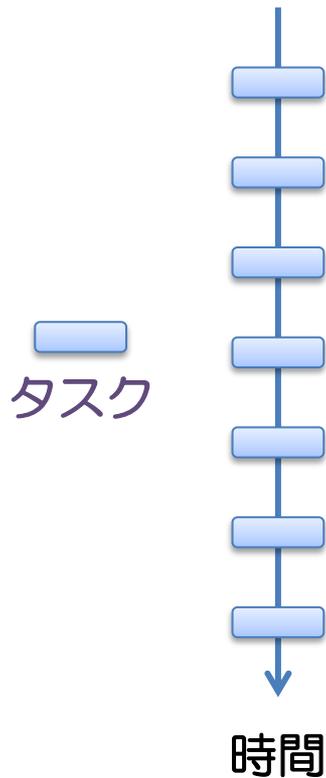
ISR: Interrupt Service Routine

割り込み要因：内部，外部
何がきっかけ？

割り込み

1.2 ポーリング

ポーリング関数の実行



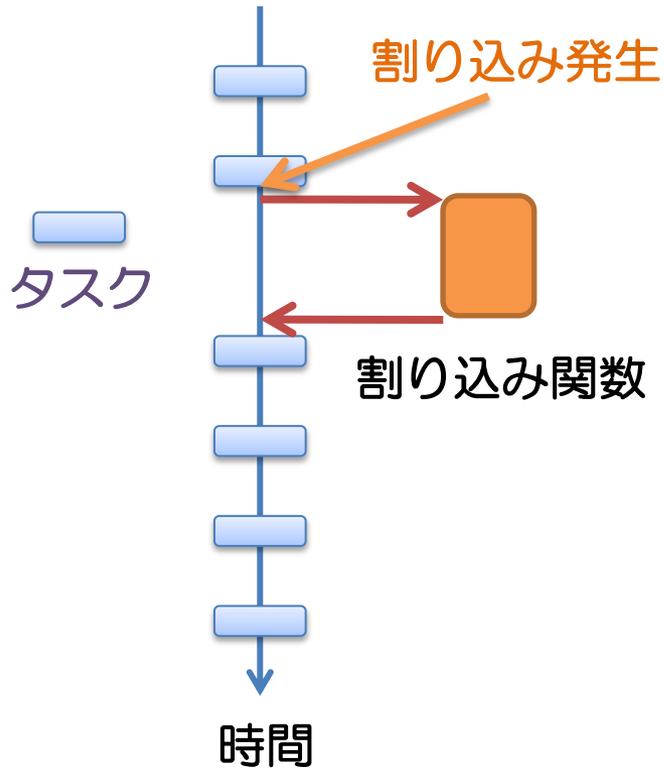
```
void main()
{
    init();    // ハードウェア初期化

    setup();  // ユーザによる初期設定

    while (1) { // 無限ループ
        loop(); // タスク：ポーリング関数
    }
}
```

1.3 ポーリングと割り込み

ポーリング関数と割り込み関数の実行



```
int main()
{
    init();    // ハードウェア初期化

    setup();  // ユーザによる初期設定

    while (1) { // 無限ループ
        loop(); // タスク実行(ポーリング)
    }
}

void isr() { // 割り込み関数
    // タスク実行
}
```

2 タイマ割り込み

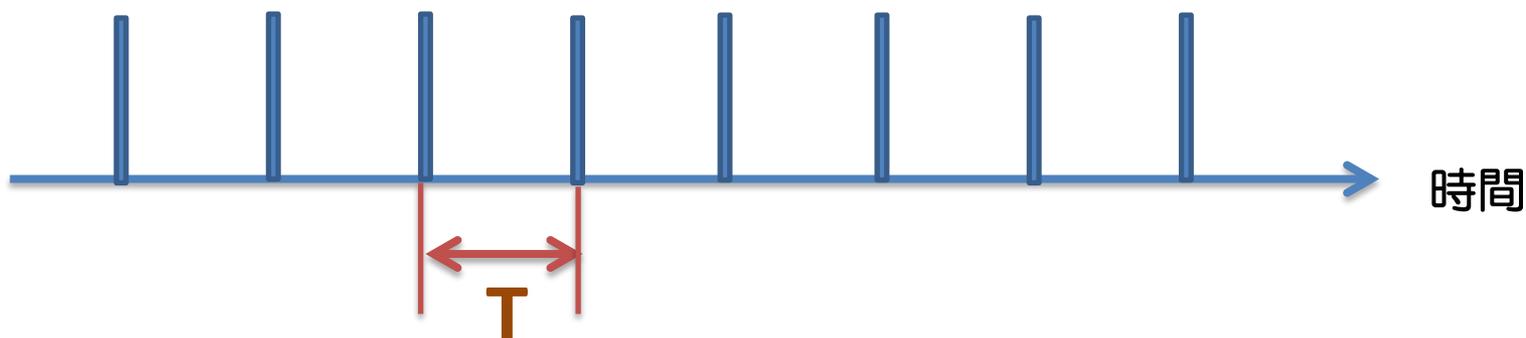
2.1 タイマ割り込み

- 周波数と周期

周波数(f)・・・1秒間に繰り返し発生するイベントの回数

周期(T)・・・繰り返しイベントの発生間隔

周波数と周期の関係： $f = 1 / T$



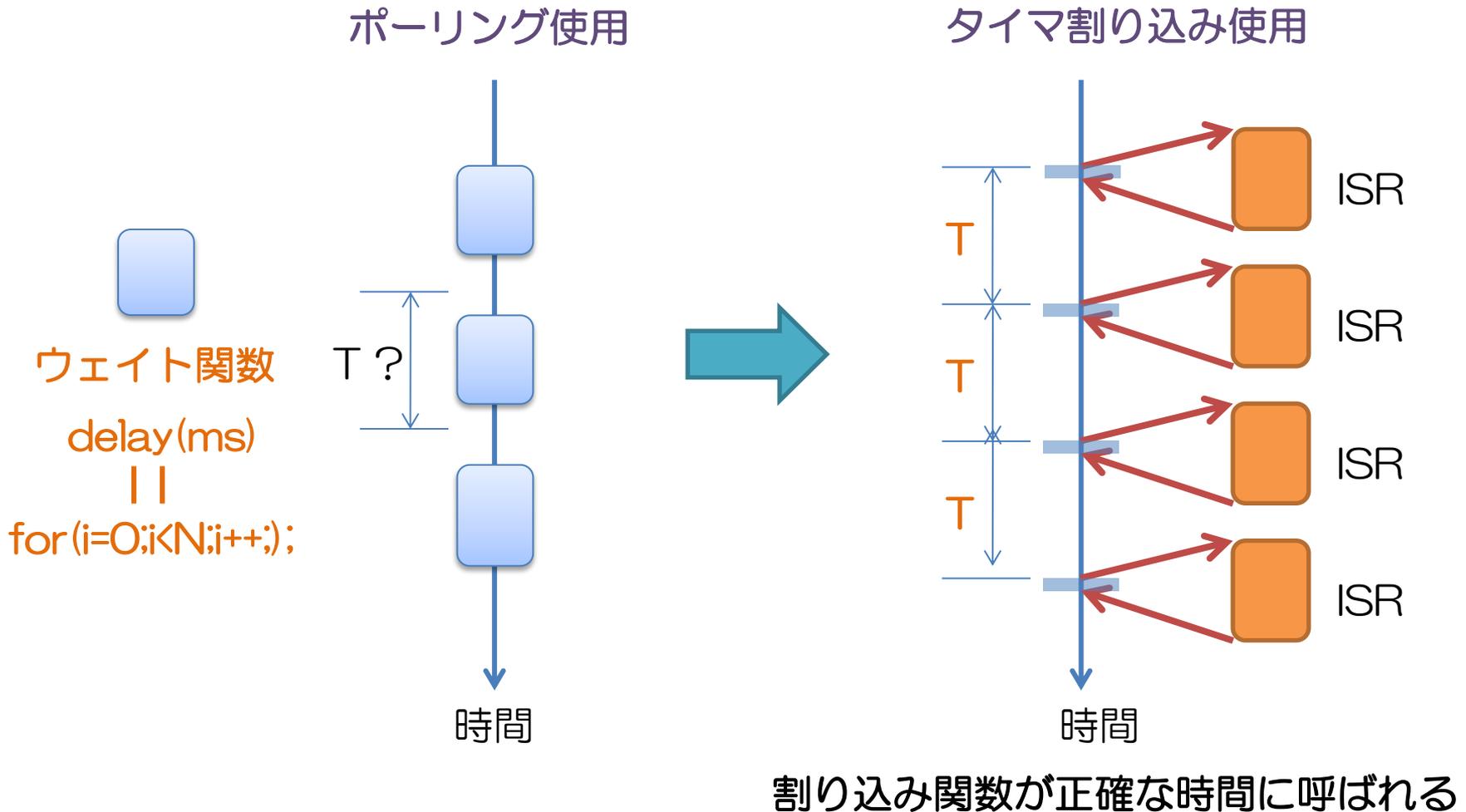
マイコンの動作周波数 16 MHz

新生児の脈拍： 120 times / minute

周期 $T = 0.5$ (sec)

周波数 $f = 2.0$ Hz (2 times / sec)

2.2 タイマ割り込み



2.3 タイマの種類

マイコン：

Arduino UNO (AVR社 ATmega328P)

3種類のタイマ：

タイマ0, タイマ1, タイマ2

タイマ0：8ビット

—delay(), millis(), micros() 関数に使用

タイマ1：16ビット

—Servoモータライブラリに使用

—TimerOneライブラリに使用

タイマ2：8ビット

—tone() 関数に使用

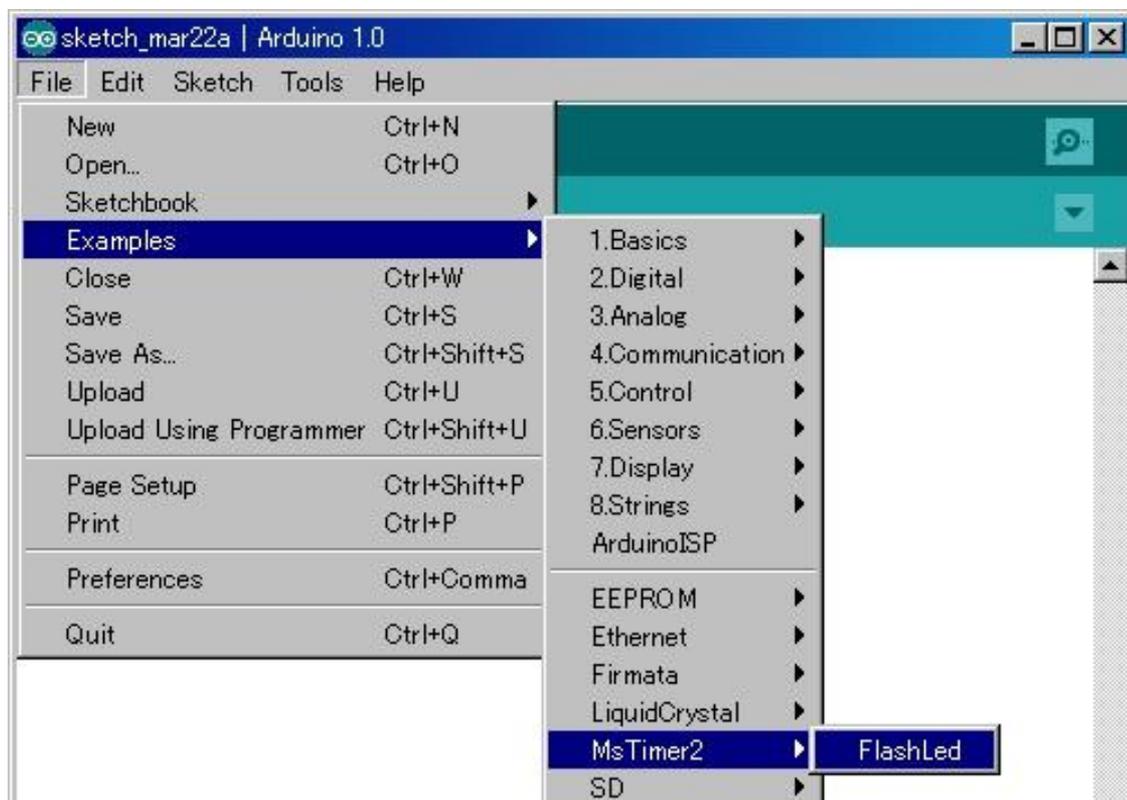
—MsTimer2ライブラリに使用

—3番ピン, 11番ピンのPWMとの「併用不可」

2.4 タイマライブラリの利用

MsTimer2 Library: 1ミリ秒単位でタイマ割り込みを発生させる

1. MsTimer2.zipを展開
2. MsTimer2フォルダを
C:\¥arduino-1.0.5¥libraries
に移動
3. Arduinoを実行
4. File -> Examples ->
MsTimer2 -> FlashLed
5. サンプルスケッチを
実行



E8.1 タイマ割り込み

FlashLED: タイマ割り込みにより500ミリ秒ごとにLEDをON,OFF

```
#include <MsTimer2.h>
```

```
const int ledPin = 13;
```

```
// 割り込み関数
```

```
void flash() {  
    static boolean output = HIGH;  
    digitalWrite(ledPin, output);  
    output = !output; // toggle the LED  
}
```

```
void setup() {
```

```
    pinMode(ledPin, OUTPUT);
```

```
    MsTimer2::set(500, flash); // 500ms 周期, 割り込み関数 isrflash()
```

```
    MsTimer2::start();        // タイマの開始
```

```
}
```

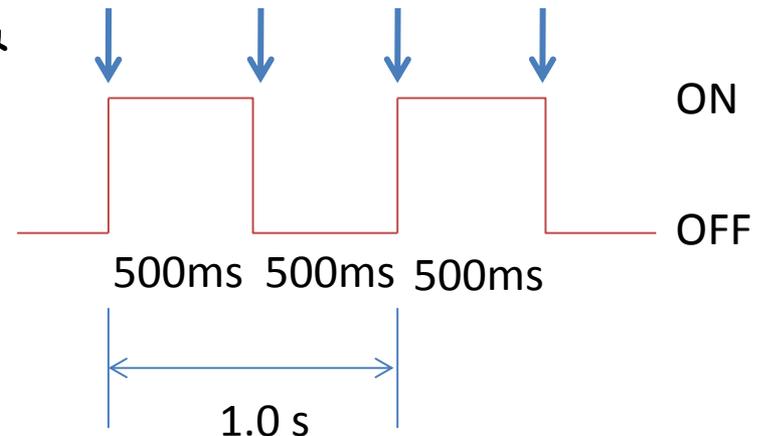
```
void loop() {
```

```
    // Nothing to do
```

```
}
```

割り込み

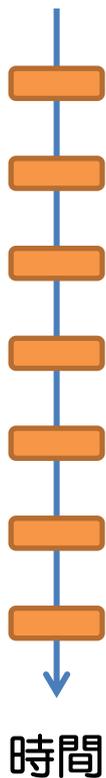
LED



3 外部割り込み

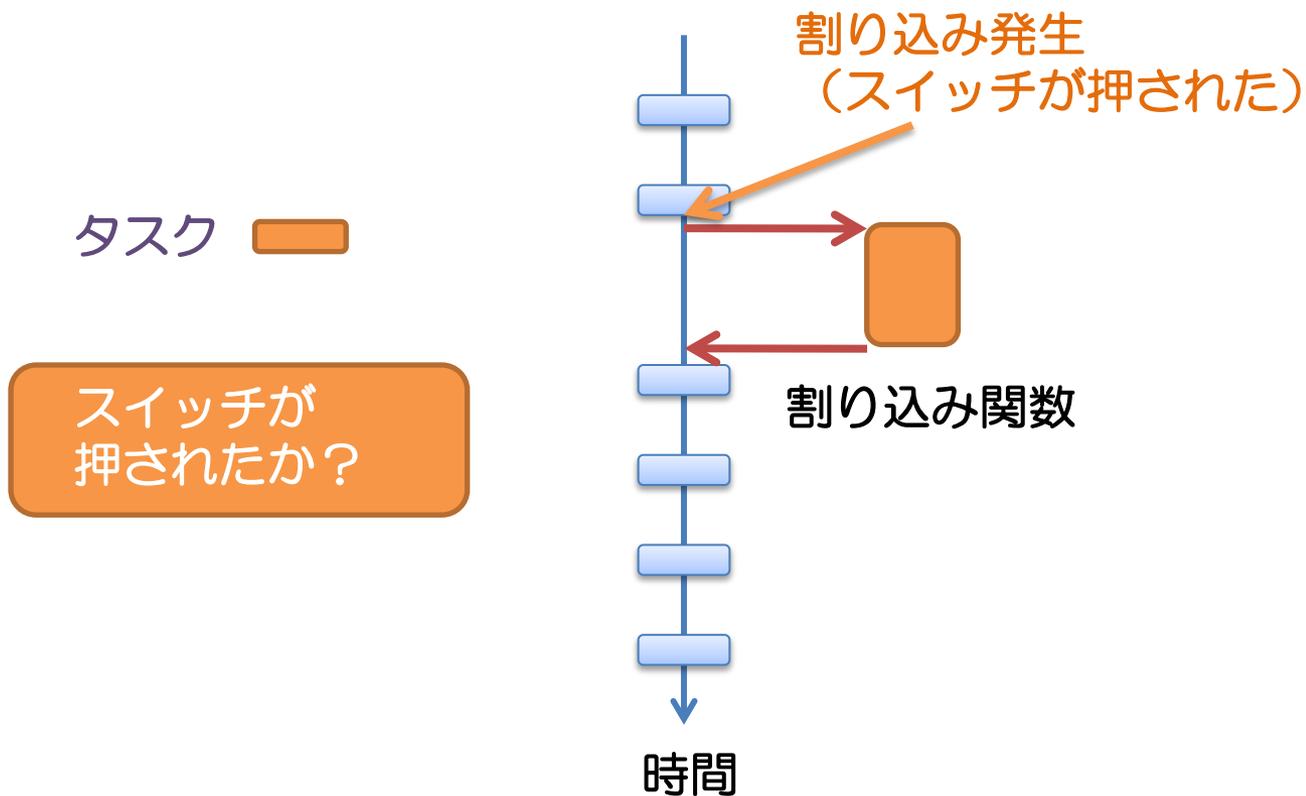
3.1 外部割り込み

ポーリングによる実行



常に監視が必要

外部割り込みによる実行



変化があったときのみ

3.2 外部割り込みのピン

Arduino UNO

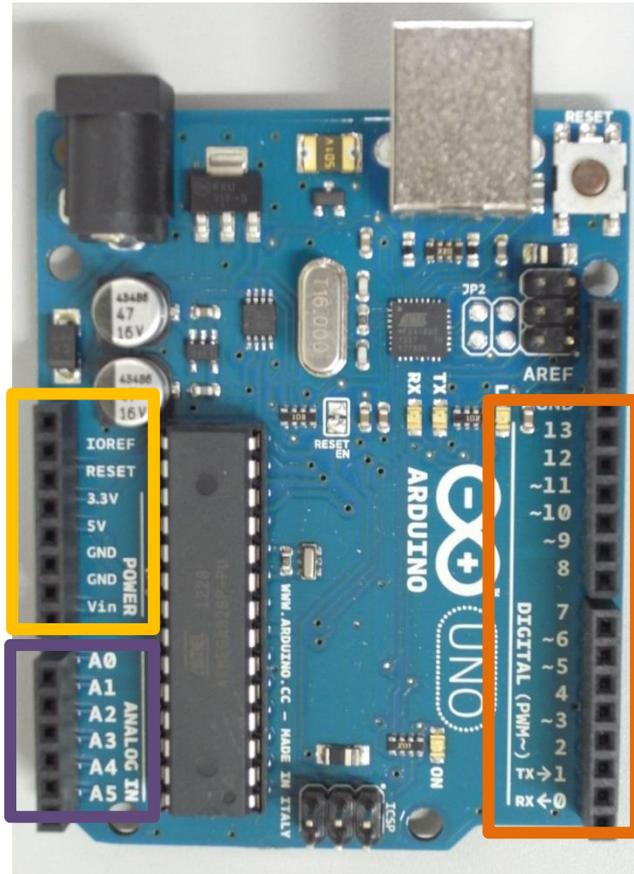
外部割り込み：
2番ピン, 3番ピン

電源

3.3V
5V
GND
GND
GND

アナログ

A0
~
A5



デジタル 0~13

I2C: A4(SDA) A5(SCL), シリアル: D0 (RX), D1 (TX)

SPI: 11 (MOSI), 12 (MISO), 10 (SS), 13 (SCK)

PWM: 3, 5, 6, 10, 11, ExtINT: 2, 3, LED: 13

3.2 外部割り込みの種類（要因）

立ち下がりエッジ
FALLING



立ち上がりエッジ
RISING



ピン変化
CHANGE



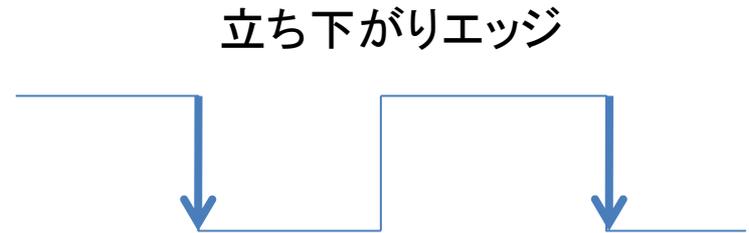
ローレベル
LOW



E8.2 スイッチによる割り込み

スイッチを押すたびにLEDがON,OFF

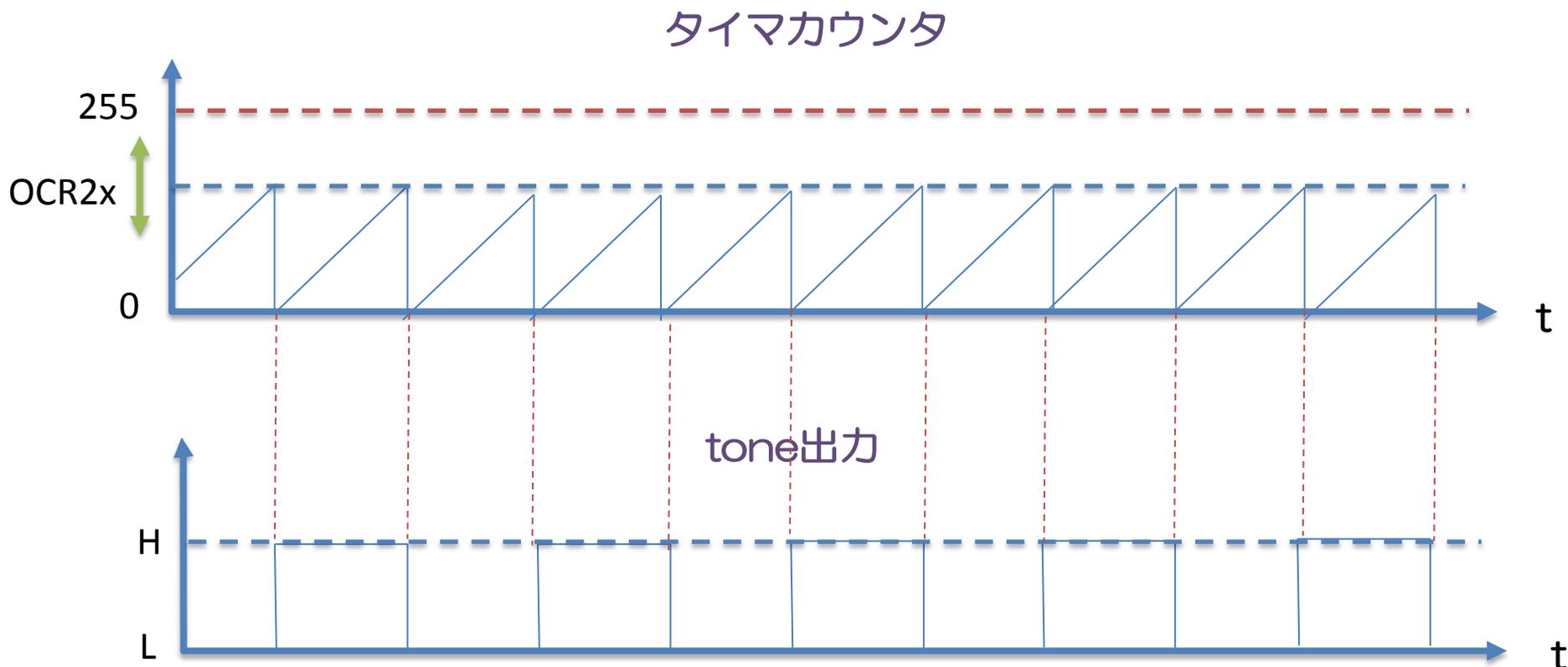
```
const int ledPin = 13;
const int intPin0 = 2; // interrupt 0, pin(2)
volatile int state = LOW;
void setup()
{
  pinMode(ledPin, OUTPUT);
  digitalWrite(intPin0, HIGH); // set HIGH state
  pinMode(intPin0, INPUT);
  attachInterrupt(0, isrSwitch, FALLING); // switch pin on interrupt 0 (pin 2)
}
void loop() {
  digitalWrite(ledPin, state);
}
void isrSwitch() {
  state = !state; // NO delay function in the interrupt function
}
```



4 割り込みでブザーを鳴らそう

4.1 tone()関数の利用

tone()関数・・・タイマを使って方形波を作り出す関数



E8.3 tone関数による音の生成

Sketch Example → Digital→toneMultiple

```
void setup() {  
}  
  
void loop() {  
  tone(6, 440, 200); // ピン6, 440Hzを200ミリ秒発生  
  delay(200);  
  noTone(6);        //ピン6の出力停止  
  
  tone(7, 494, 500); //ピン7, 494Hzを 500ミリ秒発生  
  delay(500);  
  noTone(7);        //ピン7の出力停止  
  
  tone(11, 523, 300); // ピン11, 523Hzを 500ミリ秒発生  
  delay(300);  
  noTone(11);       // ピン11の出力停止  
}
```

