

2019年度 人と地域共創センター公開講座(春・夏)

AI/IoTセンサのしくみを知ろう(基礎編)

第5回 距離をはかろう



徳島大学技術支援部

辻 明典 博士(工学)

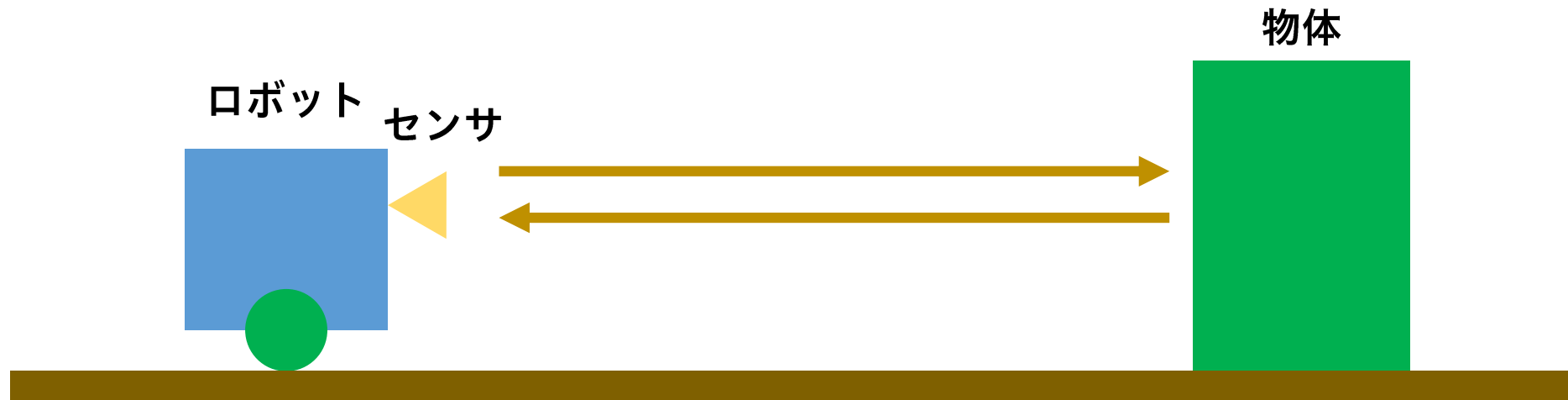
E-mail: a-tsuji@is.tokushima-u.ac.jp

概要

- ▶ 距離センサ
- ▶ 距離センサの計測原理
- ▶ 距離センサのプログラム
- ▶ 距離センサの関数

距離の計測

- ▶ 距離センサ・・・光や音が物体にあたって反射するのにかかる時間や変位を利用
 - 物体の有無，物体との距離，物体の位置や動き
 - 赤外線，超音波，レーザーなどを使用
 - 距離の計測方法：位相差検出方式，ToF方式，三角測距方式



レーザー距離センサ(VL53L0X)

▶ レーザー：VCSEL (Vertical Cavity Surface Emitting LASER)

- 垂直共振器面発光レーザ
- 波長：940nm (赤外線)
- **クラス1レーザー**

▶ 計測原理

- ToF (Time of Flight)方式

▶ 計測距離

- 白色ターゲット～200cm
- 灰色ターゲット～80cm

▶ **レーザーから放出されたレーザー光は、たとえ小さな放出量であってもパワー密度が高く、人体に有害となる場合がある。**

クラス1：直接ビーム内観察を長時間行っても、またそのとき、観察用光学器具（ルーペ又は双眼鏡）を用いても安全であるレーザー製品。



ToF (Time of Flight)方式

▶ 距離： $L = (c \times t) / 2$

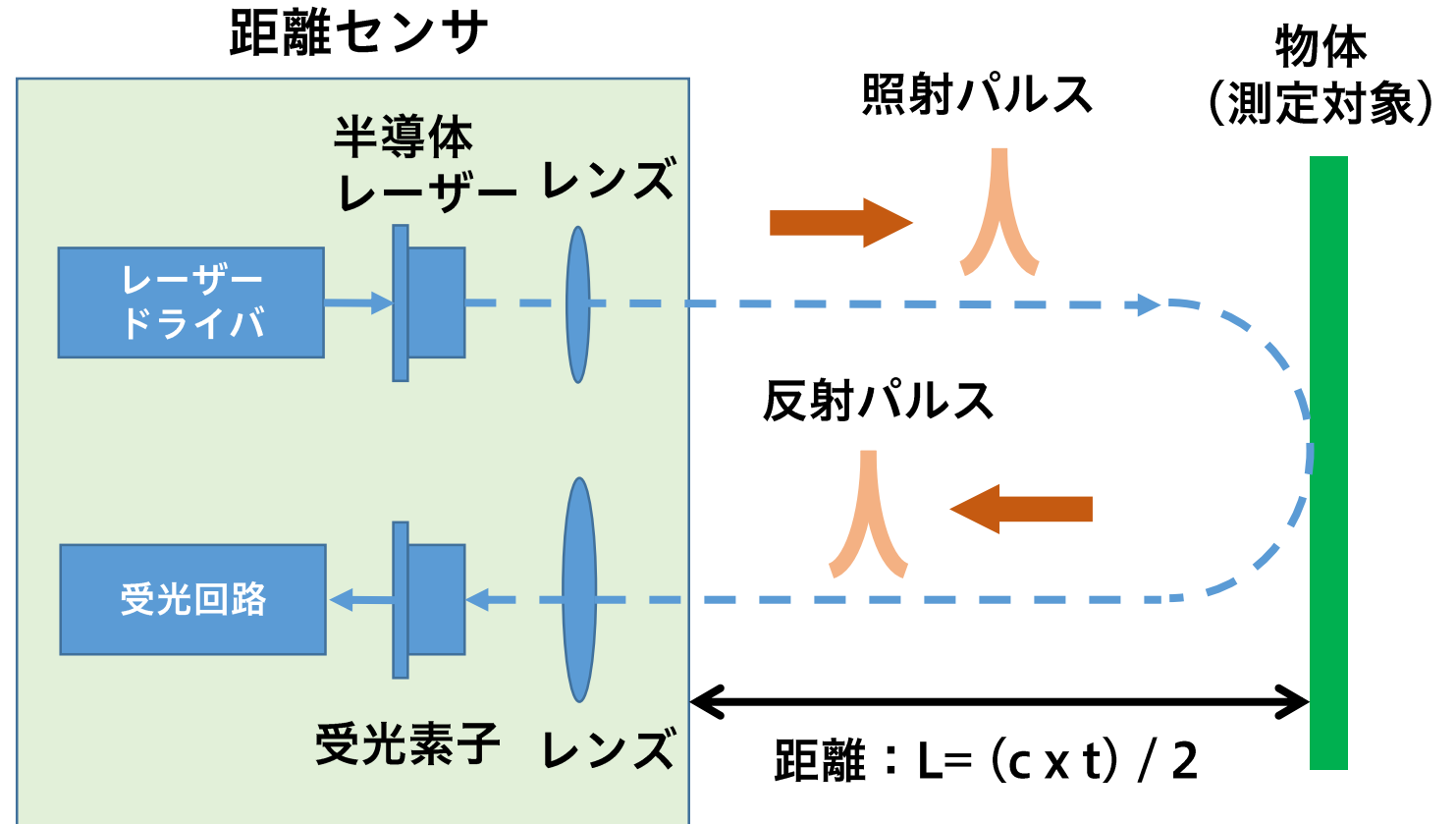
- c : 光速
- t : 往復の時間

▶ 光速

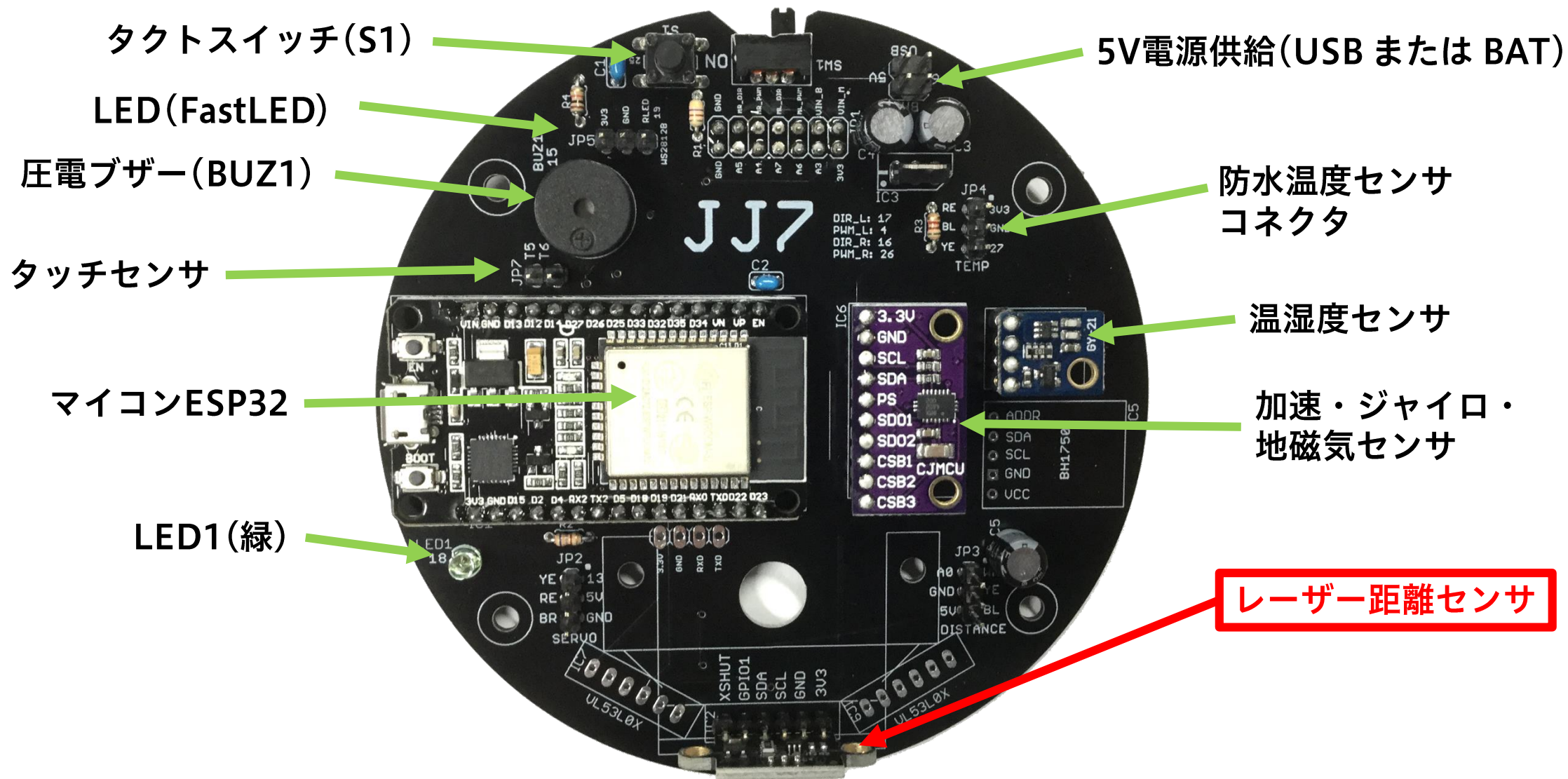
- $c = 299,792,458 \text{ m/s}$
- 約 30万 km/s

▶ 1cmの分解能

- 約67psの検出精度が必要
- p(ピコ)は, 10の -12 乗
- $t = (L \times 2) / c$
 - $L = 0.01\text{m}$



マイコンボード(JJ7)



Ex0501 : 距離計測プログラム

- ▶ 距離の計測(mm単位)
- ▶ ツール→シリアルプロッタ

```
#include <Wire.h>
#include <VL53L0X.h>

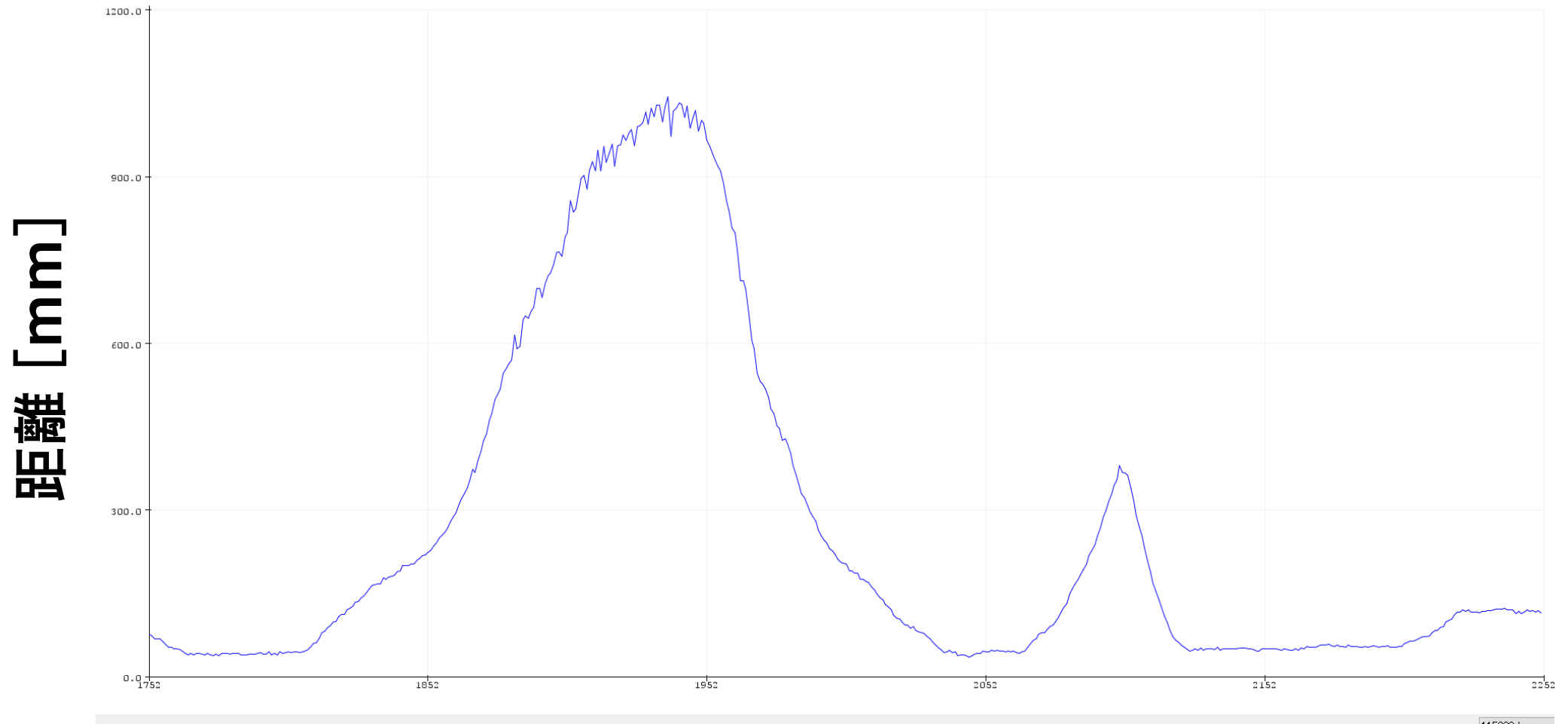
VL53L0X sensor;

void setup() {
  Serial.begin(115200);
  delay(100);

  Wire.begin();
  sensor.init();
  sensor.setTimeout(500);
  sensor.startContinuous();
}
```

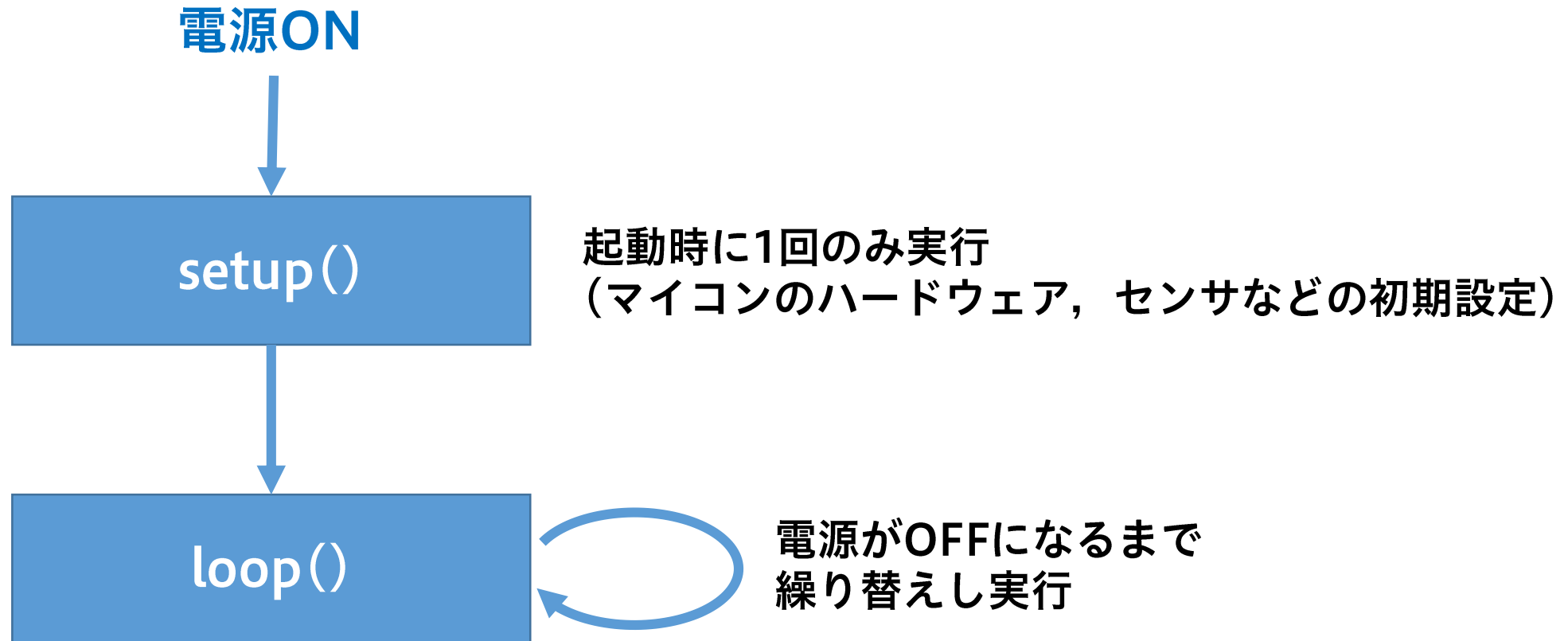
```
void loop() {
  unsigned int d =
  sensor.readRangeContinuousMillimeters();
  if (sensor.timeoutOccurred()) {
    Serial.print(" TIMEOUT");
  }
  Serial.println(d);
}
```

距離のグラフ



Arduinoのプログラム

▶ 関数 : setup() と loop()



Ex0502 : 一定の範囲内に入ったときに距離表示

▶ if文を使用

- 0mmより大きく, かつ, 500mm未満

```
#include <Wire.h>
#include <VL53L0X.h>

VL53L0X sensor;

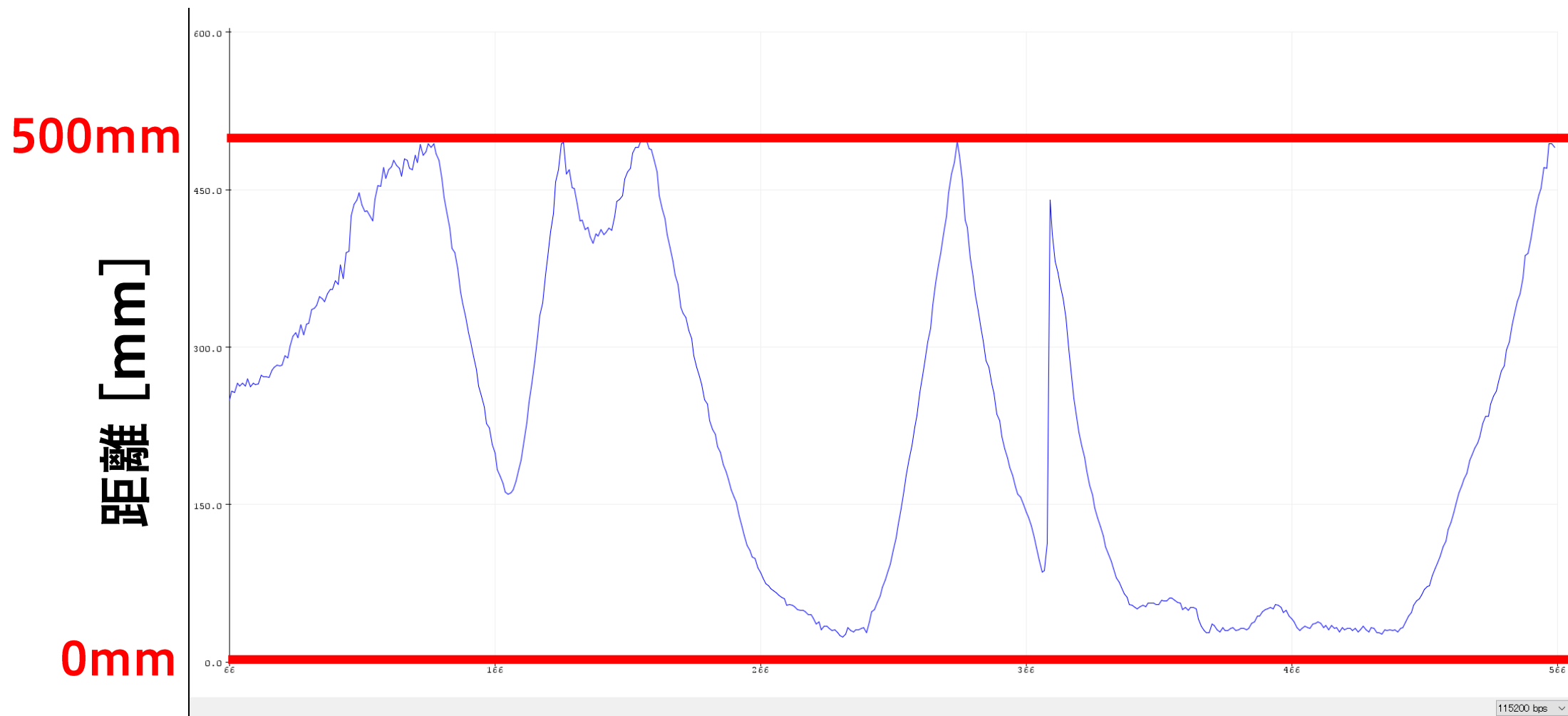
void setup() {
  Serial.begin(115200);
  delay(100);

  Wire.begin();

  sensor.init();
  sensor.setTimeout(500);
  sensor.startContinuous();
}
```

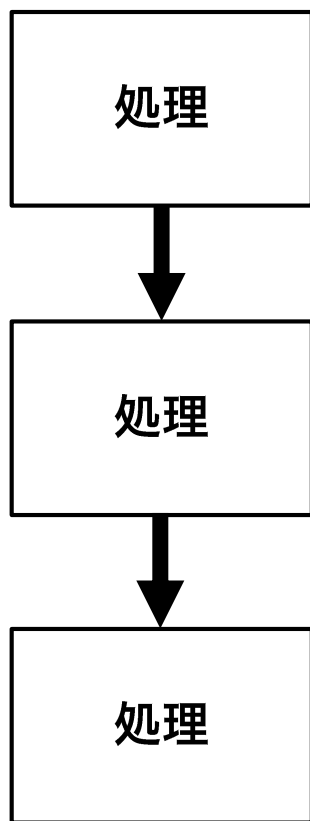
```
void loop() {
  unsigned int d =
  sensor.readRangeContinuousMillimeters();
  if (sensor.timeoutOccurred()) {
    Serial.print(" TIMEOUT");
  }
  if (d > 0 && d < 500) {
    Serial.println(d);
  }
}
```

グラフ(指定した距離範囲のとき描画)

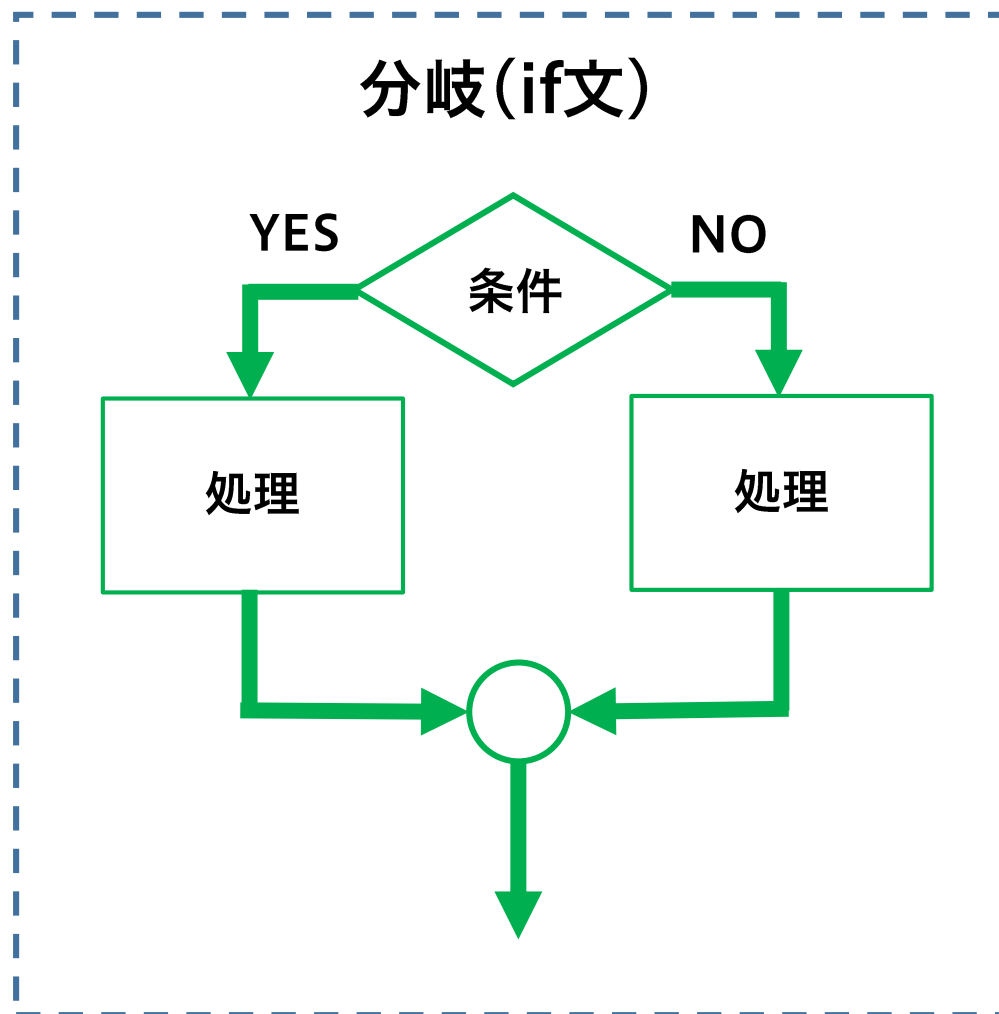


プログラムの基本構成

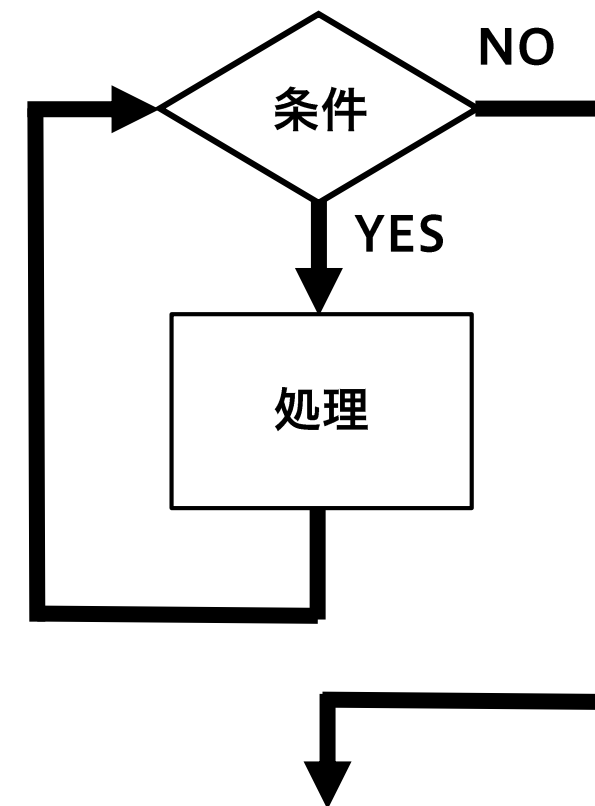
順次



分岐 (if文)



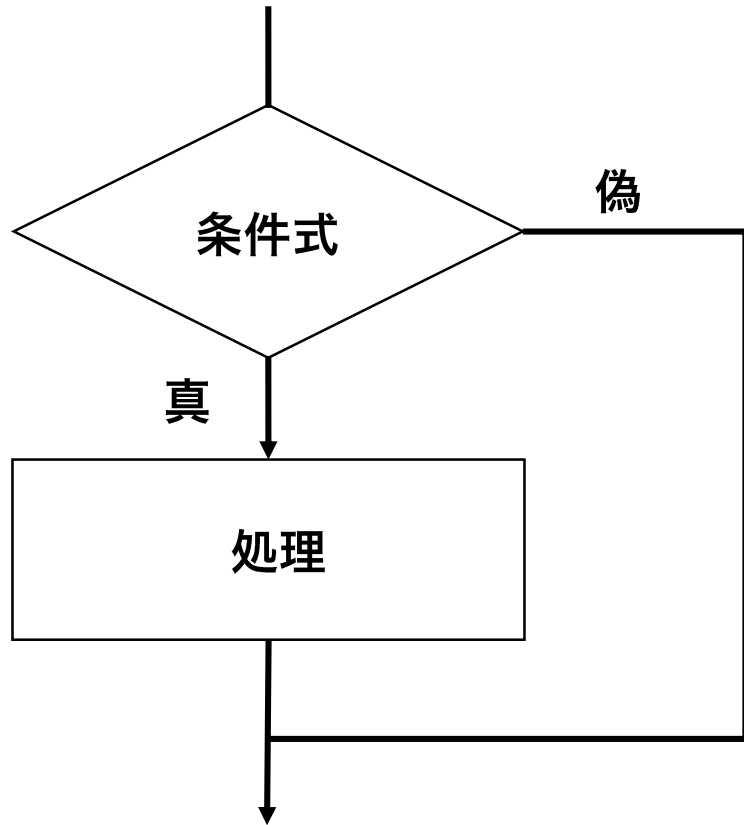
反復



if文

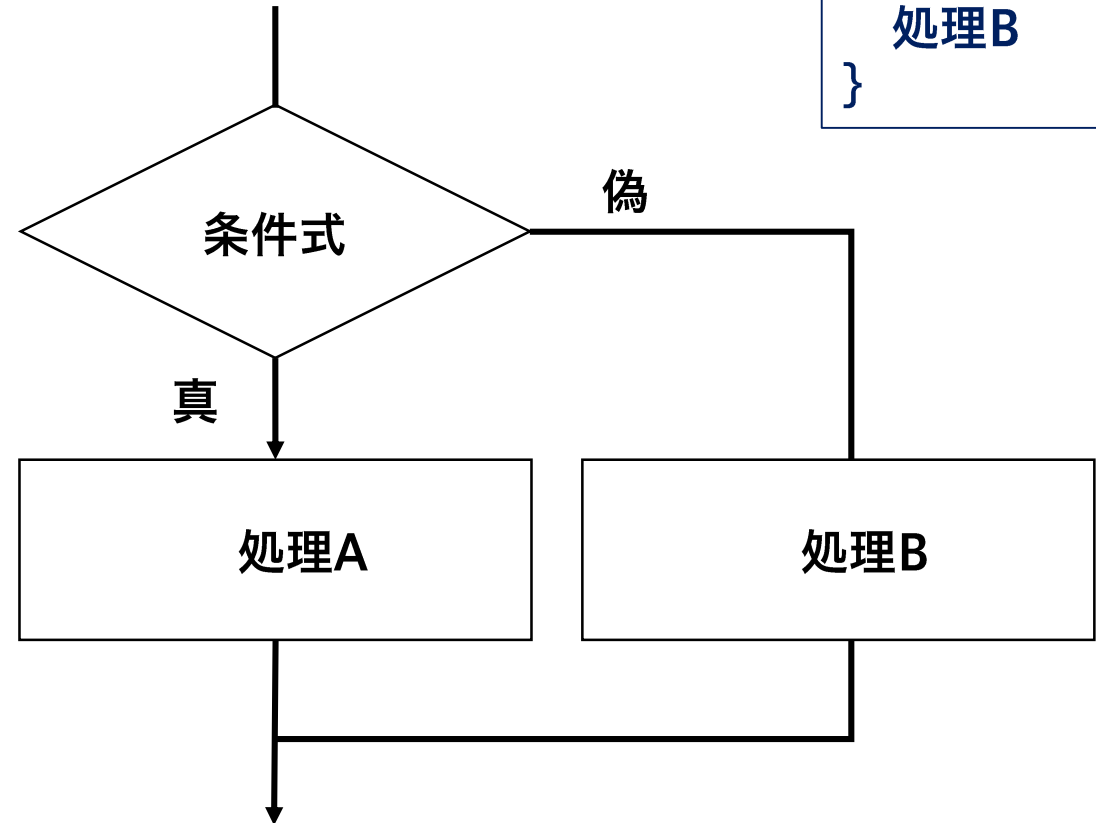
▶ if . . .

```
if (条件式) {  
  処理  
}
```



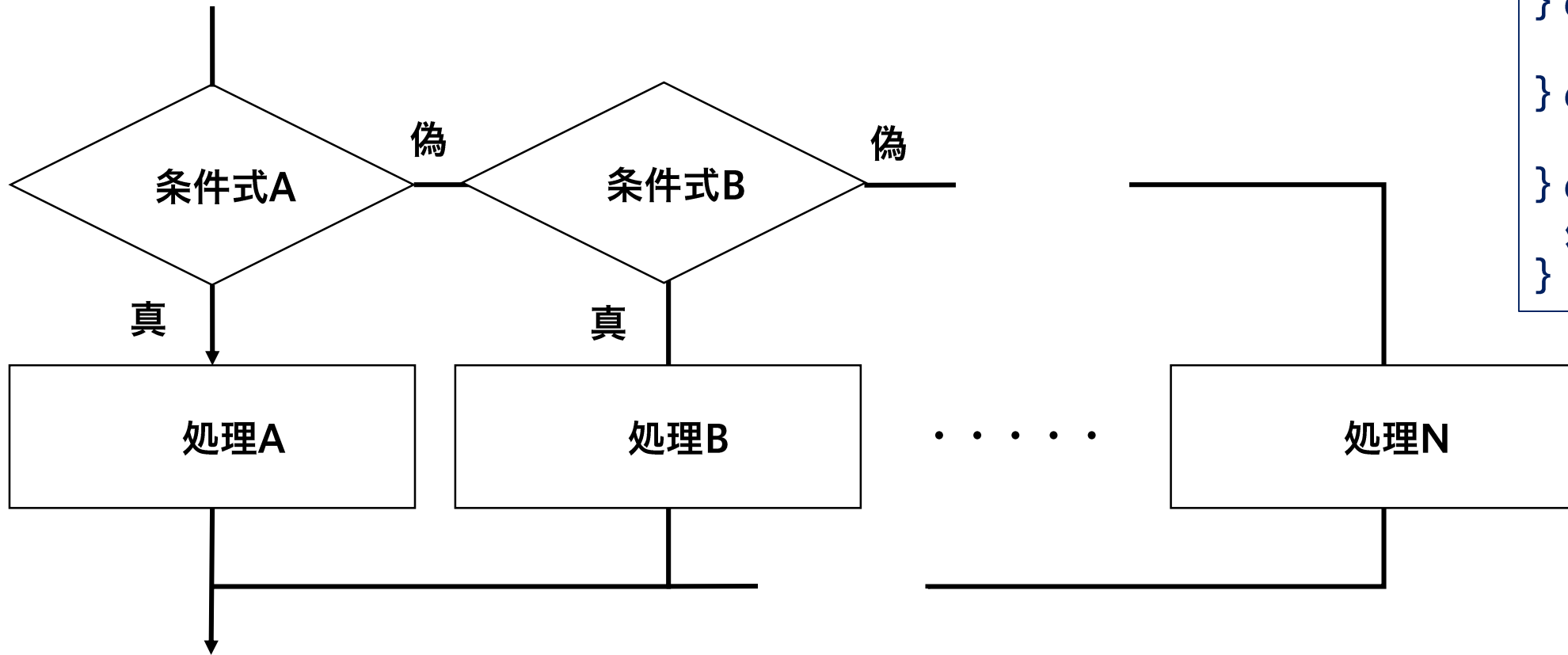
▶ if . . . else . . .

```
if (条件式) {  
  処理A  
} else {  
  処理B  
}
```



if文

▶ if . . . else if . . . else . . .



```
if (条件式A) {  
    処理A  
}  
else if (条件式B) {  
    処理B  
}  
else if (条件式C) {  
    . . .  
}  
else {  
    処理N  
}
```

条件式

▶ 条件式

$a > b$: aはbより大きい

$a < b$: aはbより小さい

$a \geq b$: aはb以上に大きい

$a \leq b$: aはb以下に小さい

$a == b$: aとbは等しい

$a != b$: aとbは等しくない

$a \&\& b$: aとbの両方の条件が一致

$a || b$: aまたはbのいずれかの条件が一致

Arduino日本語リファレンス : <http://www.musashinodenpa.com/arduino/ref/>

Ex0503 : 一定の時間間隔で距離を表示

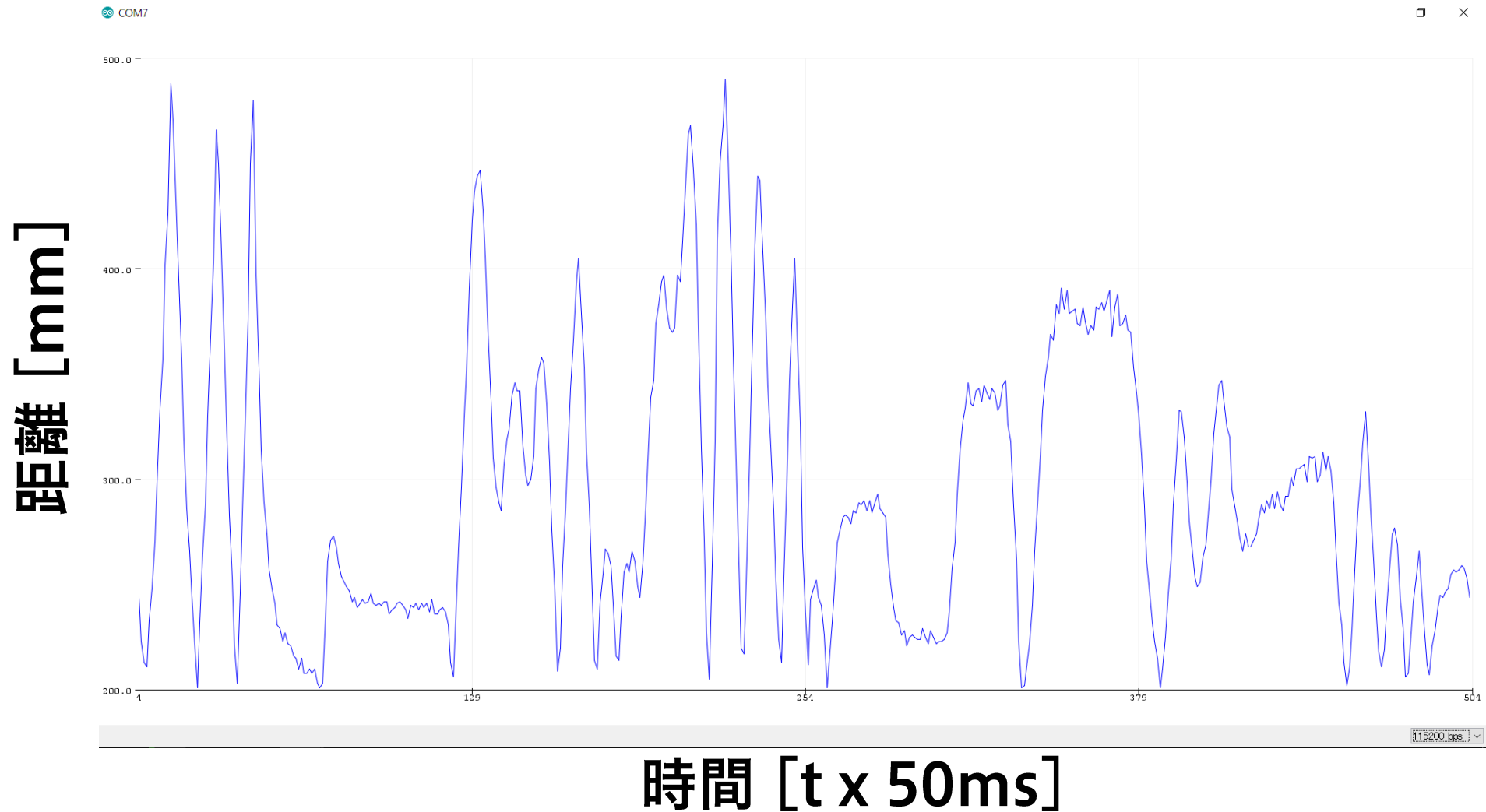
▶ 50m秒ごとに計測

millis() : マイコンが起動してからの時間(ms)

```
unsigned long d_timen=0, d_timep=0; // 距離センサ用の計測タイマー変数

void loop() {
  d_timen = millis(); // 現在の時間(ms)
  if (d_timen - d_timep > 50) { // 50ms経過したら
    d_timep = d_timen; //
    unsigned int d = sensor.readRangeContinuousMillimeters(); // 距離の計測(mm単位)
    if (sensor.timeoutOccurred()) { // タイムアウト発生
      Serial.println(" TIMEOUT");
    }
    if (d > 0 && d < 500) {
      Serial.println(d); // 計測結果の表示
    }
  }
}
```


グラフ(距離 - 時間)



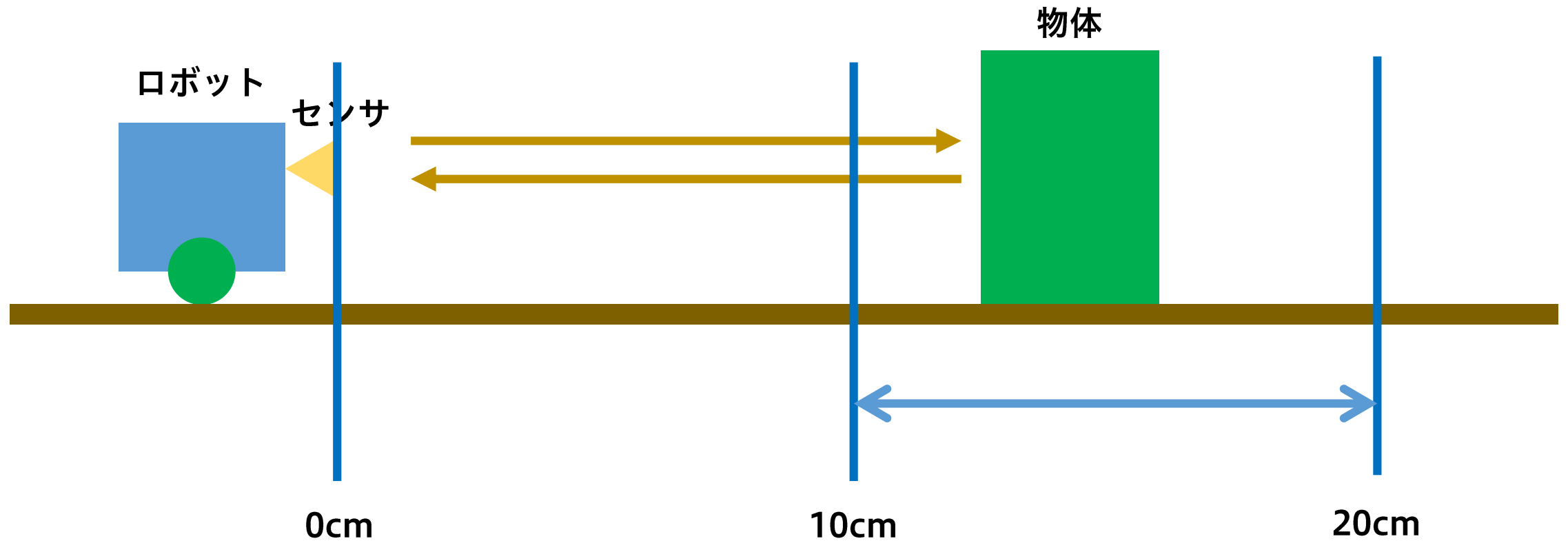
Ex0504 : 距離をはかる関数

- ▶ **dist_measure**: 計測時間間隔, 計測範囲(最小, 最大)を指定する関数

```
int dist_measure(unsigned int d_time, unsigned int d_min, unsigned int d_max) {
    d_timen = millis(); // 現在の時間(ms)
    if (d_timen - d_timep > d_time) { // 50ms経過したら
        d_timep = d_timen; // 前の時間を現在の時間に
        unsigned int d = sensor.readRangeContinuousMillimeters(); // 距離の計測(mm単位)
        if (sensor.timeoutOccurred()) { // タイムアウト発生
            Serial.println(" TIMEOUT");
        }
        if (d > d_min && d < d_max) {
            return d;
        }
    }
    return -1;
}
```

dist_measure関数の使い方

- ▶ `int = dist_measure(50, 100, 200);` // 50ms毎に, 10cm~20cmを計測



考えてみよう

- ▶ Ex0502の距離の計測範囲を変更
- ▶ Ex0503の計測時間間隔, 距離の計測範囲を変更
- ▶ Ex0504の関数を使って, ある範囲に入ったときに"FOUND"と表示
- ▶ Ex0504の関数を使って, ある範囲に入ったときにLEDを点灯

- ▶ レーザー距離センサの応用を考えてみよう。