

ソフト班 Linux&ROS教育

# ROS講習会

#4 トピック通信を理解しよう



# 今回の内容

---

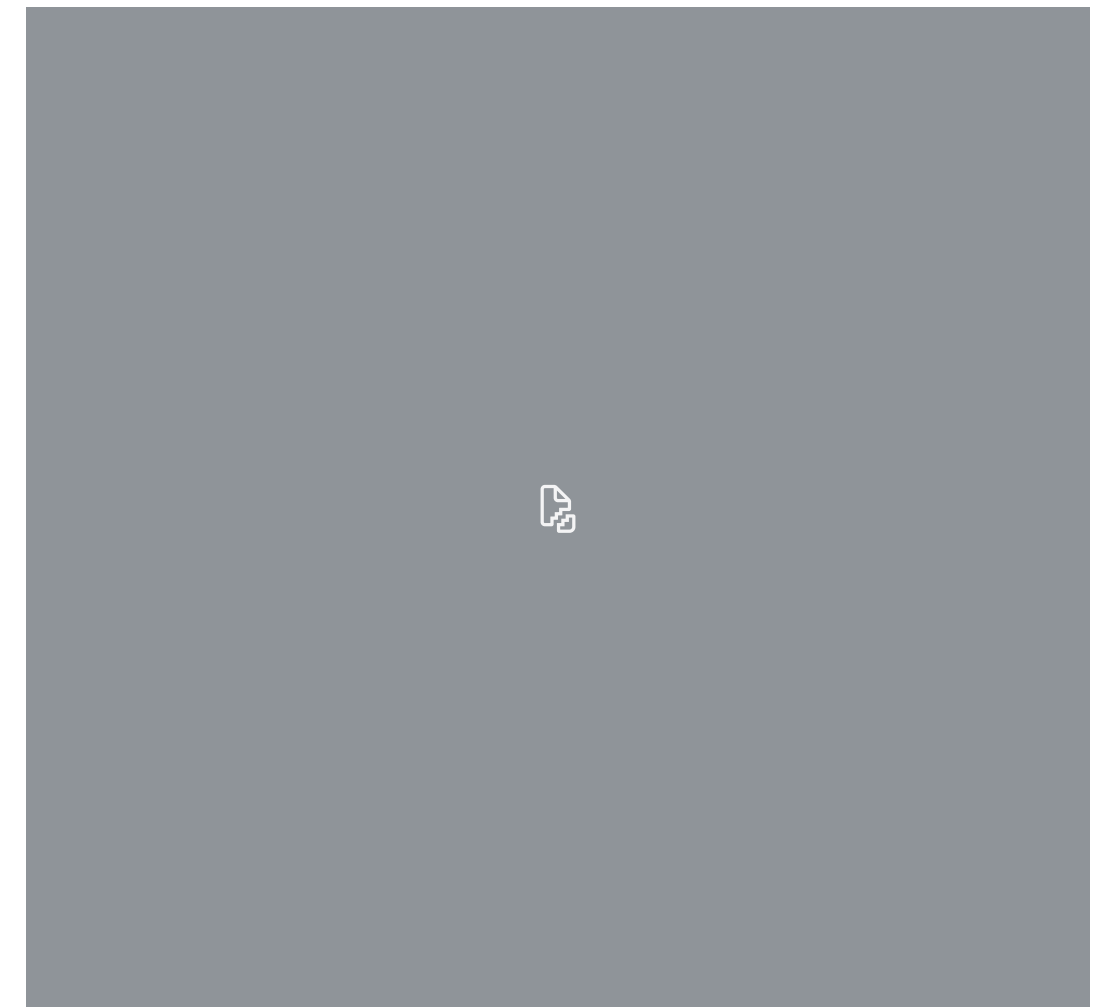
- ROSのデータ通信
- トピック通信とは
- レッツココーディング



# ROSのデータ通信

# ロボットにおいて**データ通信**は必須

- センサーの情報を取得
- プロセス開始・終了の合図
- etc...



### 【必須】 ROSにおける2種類のデータ通信

- **トピック通信**
- **サービス通信**

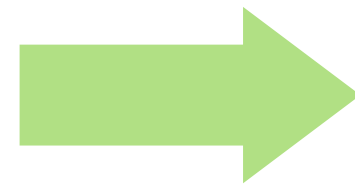
今日はトピック通信やるよ



ROSのデータ通信では、  
送る情報のことを**メッセージ**という



int型で数字の  
データ送りたい



int型の**メッセージ**  
送りたい！

メッセージには  
型とフィールドがある

# 基本的に「`std_msgs`」 の中で定義された型で事足りる

- `int`
- `String`
- `float`
- `etc...`

`std_msgs`はパッケージ作った時に、  
依存パッケージとして指定したよね





# メッセージのフィールド

## メッセージフィールドの例

### メッセージ (測位センサ)



data

min\_range

30

min\_distance

0.3

max\_range

180

max\_distance

10



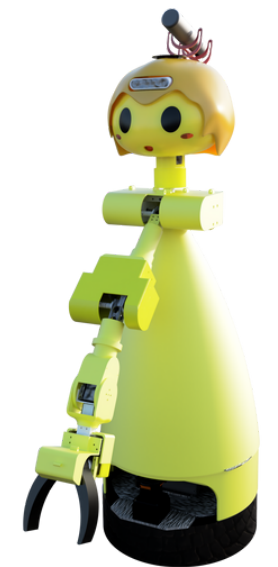
測位センサから  
最大距離知りたい



測位センサの  
メッセージを受け取る



max\_distanceフィールド  
見ればわかるね  
(data.max\_distance)



# フィールド名「data」に さらに4つのフィールドがある

**data**

**min\_range**  
30

**min\_distance**  
0.3

**max\_range**  
180

**max\_distance**  
10

一つのメッセージで  
複数の情報を扱えるね



トピックス通信とは

# トピック通信

II

多対多の非同期通信

# Topic通信

---

送信側  
ノード



受信側  
ノード

- **一方的なメッセージの送信/受信**
- **非同期通信、多対多**
- **継続的なメッセージの送受信が可能**
- **購読者 (Subscriber) と配布者 (Publisher)**  
**存在する。**

# Topic通信

---

送信側  
ノード



受信側  
ノード

ニッポン

コーポレーション



今日の課題

## 今日の課題1

---

**今日のサンプルコードを完成させる**

# 購読者 & 配布者ノードを自作しよう！

- パッケージを新しく作ってください。
- 詳細はTrelloの/活動日記ボード/ソフト班新入生リスト/  
情報共有カード、に書いてあります。