# ROSノード軽量実行環境mROS update報告

京都大学 大学院情報学研究科

**祐源 英俊**, 高瀬 英希, 高木 一義, 高木 直史

# 背景：ROS on 組込み機器？

Linux環境を持たない組込み機器では、

ROSの活用が困難

# 背景：ROS on 組込み機器？

Linux環境を持たない組込み機器では、

ROSの活用が困難

そこで

# 背景：ROS on 組込み機器？

Linux環境を持たない組込み機器では、

ROSの活用が困難

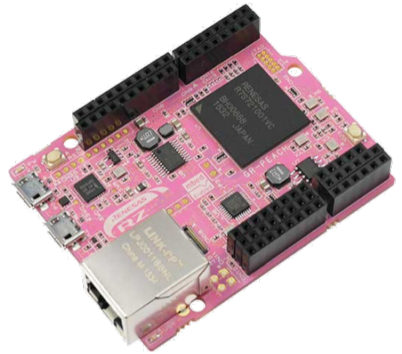
そこで

# mROS

- Linux環境を持たない組込み機器などにおいて
  ROSノードを実行する軽量実行環境
  （トピックを介したPub/Sub通信に対応）
- **TOPPERS/ASPカーネル(RTOS)で動作し、軽量かつ省エネ**
- **ROSノードに近い設計モデルで、移植性が高い**

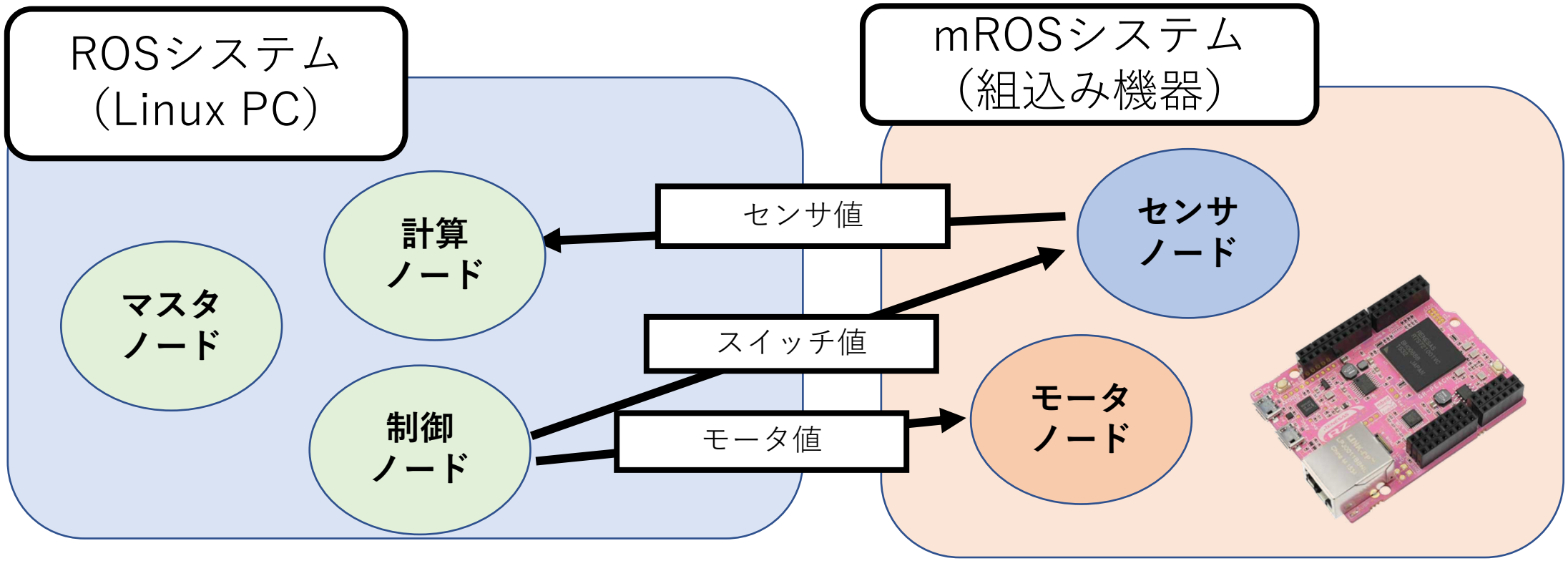
github.com/tlk-emb/mROS

京都大学 高木研究室で
2018年より開発中



# mROS

ROSを用いたシステムの一部に組込み機器を活用

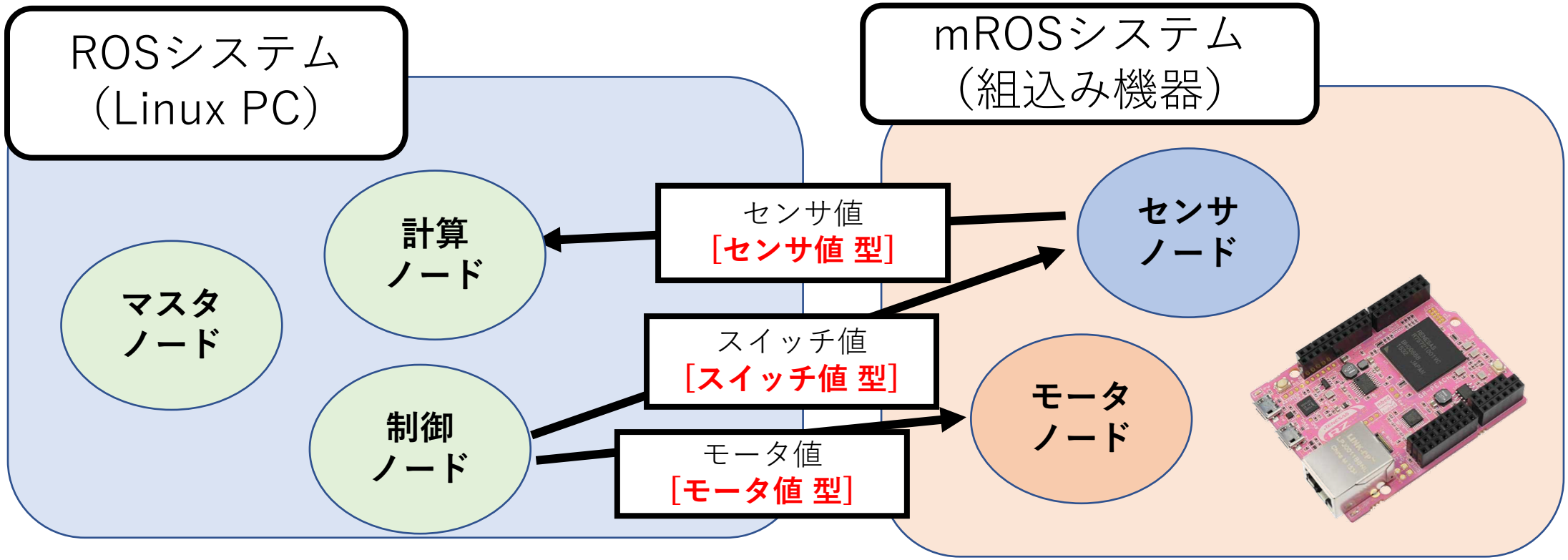


# 昨年からのupdate

MessageType対応

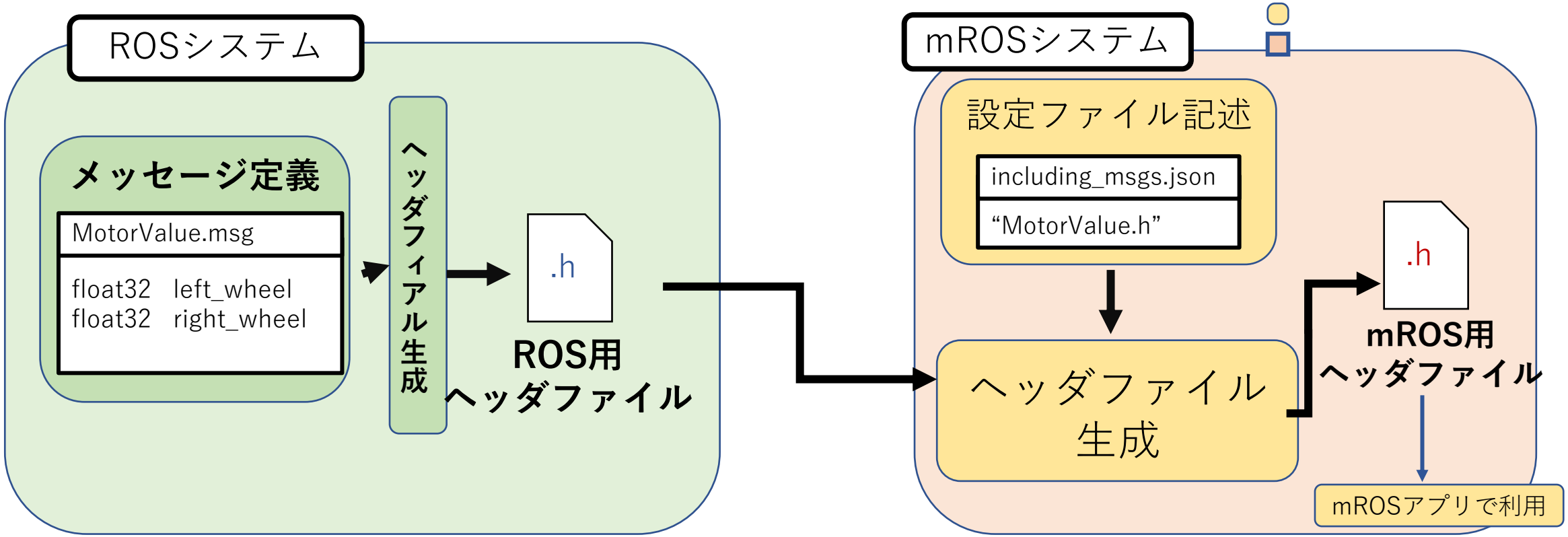
# MessageType対応

- 昨年発表段階では、メッセージ型はStringのみに対応
  **→他のプリミティブ型およびユーザ定義メッセージ型に対応**



# MessageType対応

- 他のプリミティブ型及びユーザ定義メッセージ型に対応
  →**既存のメッセージ型ファイルを流用可能**

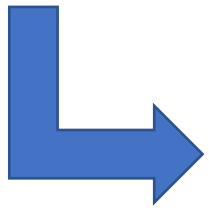


# MessageType対応

- 他のプリミティブ型及びユーザ定義メッセージ型に対応
  **→従来ROSノードに近い開発が可能に**

```
ros::init(argc,argv,"mros_node");
ros::NodeHandle n;
ros::Publisher chatter_pub = n.advertise(
    "mros_msg","std_msgs/String",1);
ros::Rate loop_rate(5);
char msg[100];
while(1){
    wait_ms(1000);
    chatter_pub.publish(msg);
    msg.score ++;
}
```

↑旧実装



新実装→

```
ros::init(argc,argv,"mros_node");
ros::NodeHandle n;
ros::Publisher chatter_pub = n.advertise<
    mros_test::PersonalData>("mros_msg",1);
ros::Rate loop_rate(5);
mros_test::PresonalData msg;
while(1){
    wait_ms(1000);
    chatter_pub.publish(msg);
    msg.score ++;
}
```

Let's

mROS !

github.com/tlk-emb/mROS