ROBOTIS製
ROSフレームワーク
ROBOTIS-Frameworkの概要

株式会社ロボティズ日本支店
田中義丸

Agenda

- 自己紹介
- 事業紹介
- 2種類のROSフレームワーク
  - Dynamixel-Workbench
  - Robotis-Framework
- Robotis-Frameworkの概要
- GitHub

自己紹介

名前：田中 義丸

所属：株式会社ロボティズ日本支店

Twitter: yoshimalucky

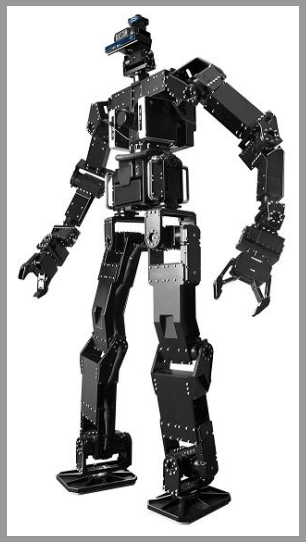
事業概要

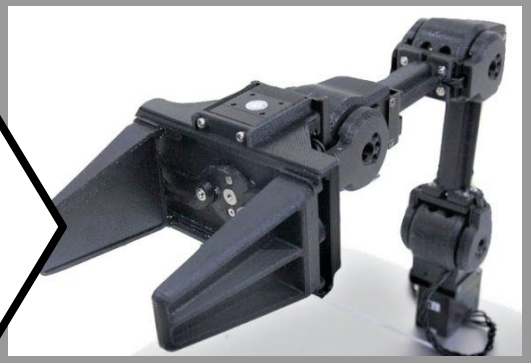
アクチュエータ・モジュール

Dynamixel シリーズ
研究開発・教育向け
デジタルコマンド式
アクチュエータモジュール

適用

ロボット・プラットフォーム





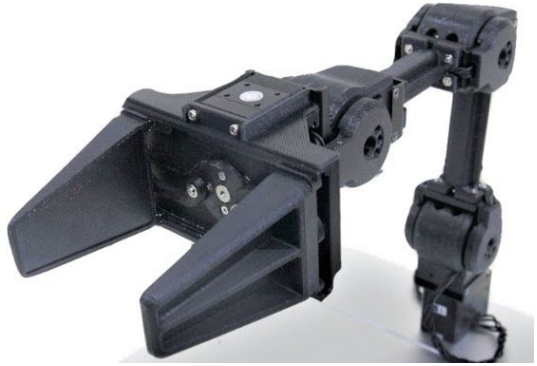
ROS対応

ROBOTIS

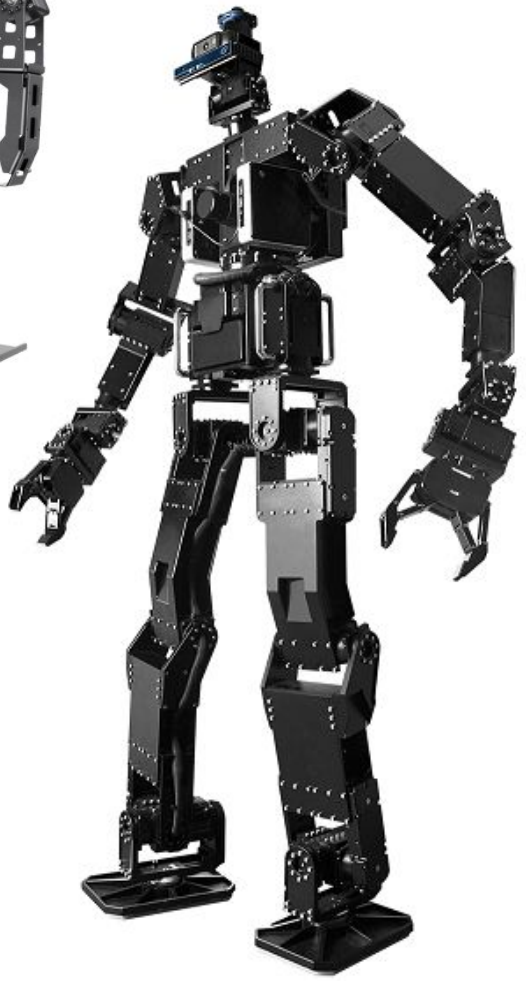
2種類のROSフレームワーク

Dynamixel Workbench

- 後発
- ビギナーにも易しい
- 主にマニピュレータに採用



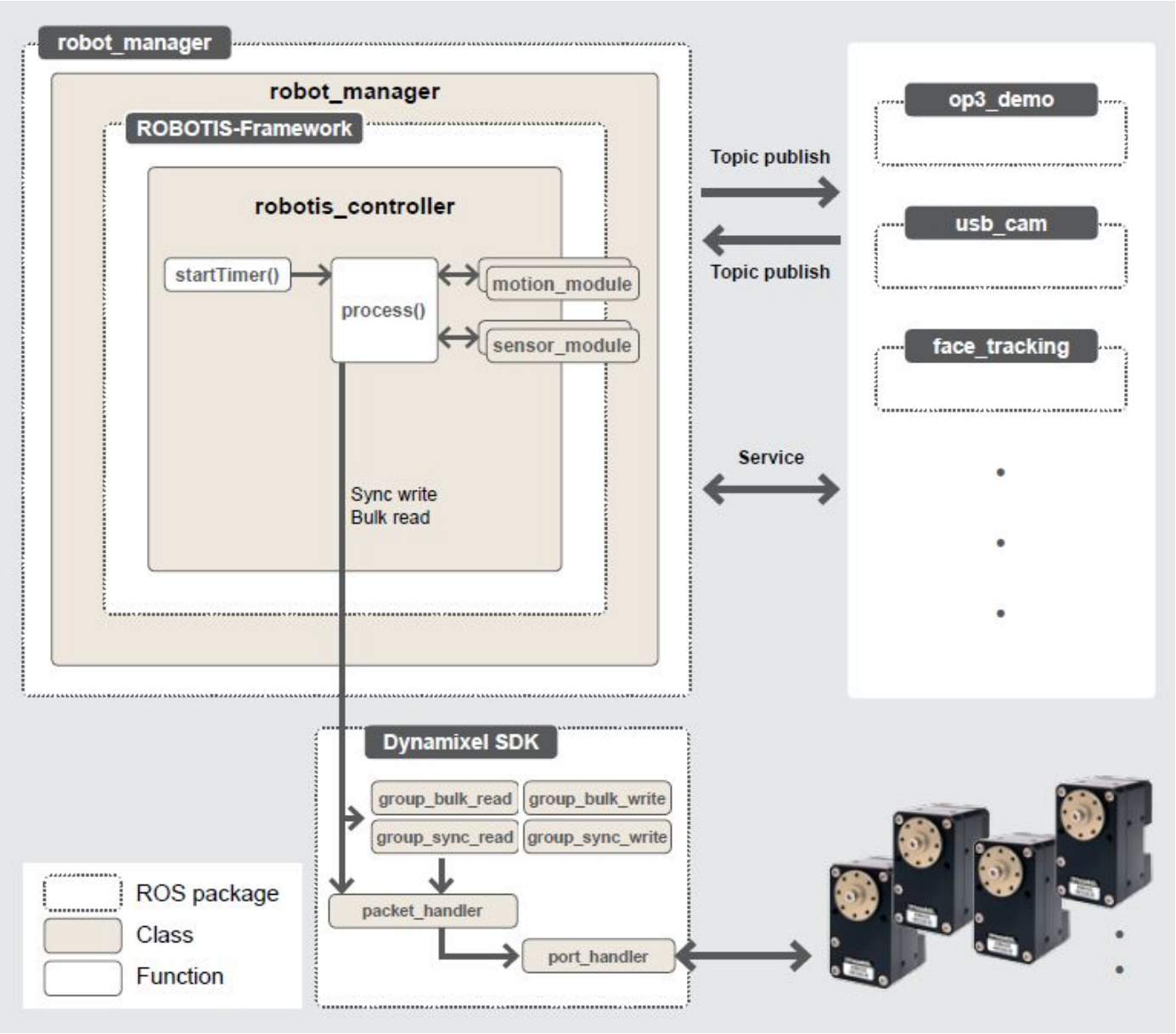
2種類のROSフレームワーク



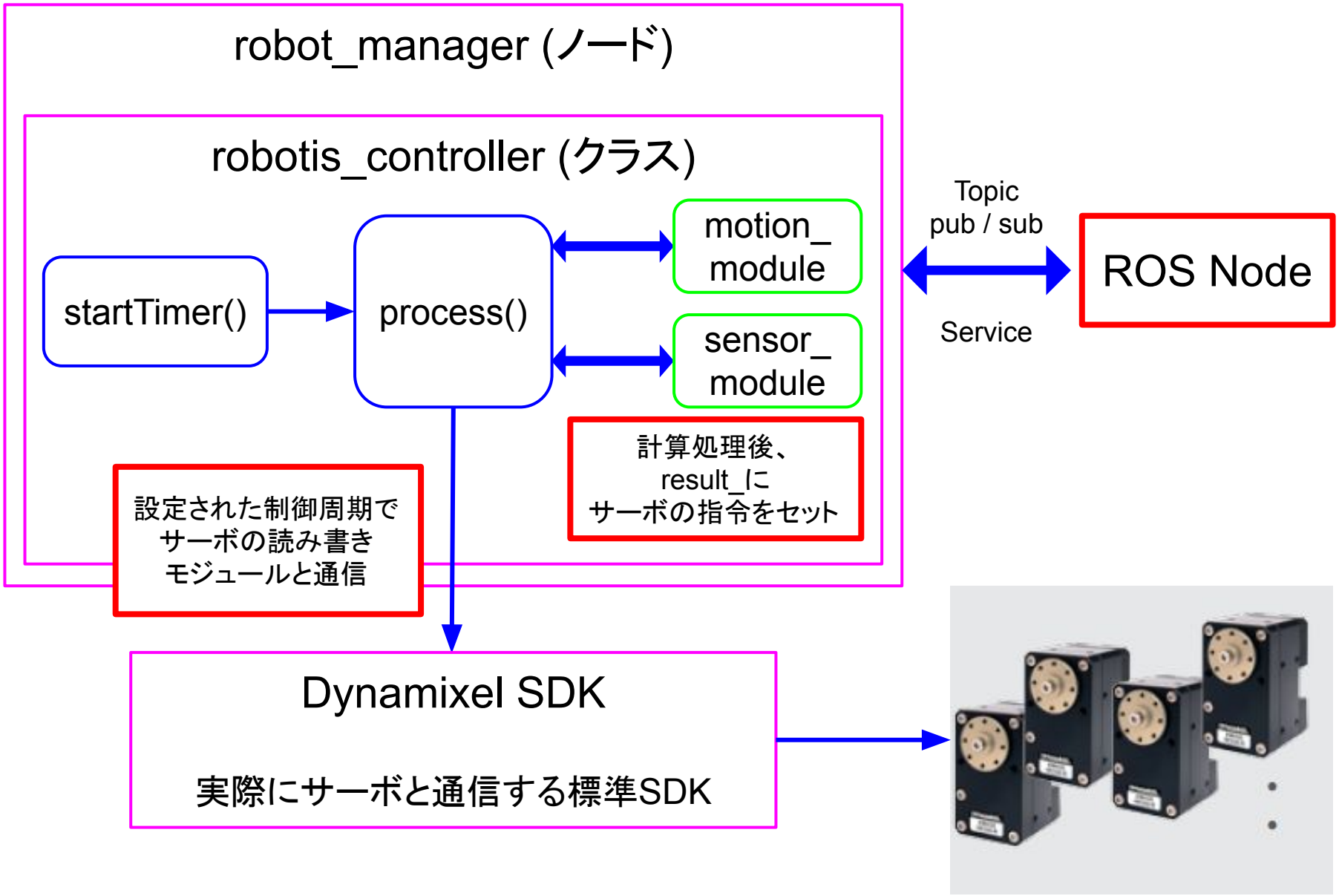
Robotis Framework

- 先発
- リアルタイム性を考慮
- Thormang3, OP3等に採用

ROBOTIS-Framework の概要



ROBOTIS-Framework のポイント



thormang3_manager

```
int main(int argc, char **argv) {
    ros::init(argc, argv, "THORMANG3_P_Manager");
    ros::NodeHandle nh;
    robotis_framework::RobotisController *controller = robotis_framework::RobotisController::getInstance();
~~~~~ 中略 (設定ファイルやパラメータの読込) ~~~~~
    sleep(1);
    controller->addSensorModule((robotis_framework::SensorModule*)FeetForceTorqueSensor::getInstance());
    controller->addMotionModule((robotis_framework::MotionModule*)ManipulationModule::getInstance());
    controller->startTimer();
   while(ros::ok()) {
      usleep(1000*1000);
     }
    return 0;
}
```

thormang3_manipulation_module 各メソッドの説明

コンストラクタ:

ManipulationModule::ManipulationModule()

パラメータやサーボの状態取得や指令送信に利用するためのリストを作成

```
result_["r_arm_sh_p1"]  = new robotis_framework::DynamixelState();
result_["l_arm_sh_p1"]  = new robotis_framework::DynamixelState();
~~~~~ 中略 ~~~~~
result_["torso_y"]      = new robotis_framework::DynamixelState();
```

デストラクタ:

ManipulationModule::~ManipulationModule()

スレッドを破棄

thormang3_manipulation_module 各メソッドの説明

ManipulationModule::initialize

- スレッド初期化
- パブリッシャ初期化
- その他設定

ManipulationModule::queueThread()

- スレッドにサブスクライバのコールバック処理を登録
- ros::CallbackQueueの開始

thormang3_manipulation_module 各メソッドの説明

ManipulationModule::process(std::map<std::string, robotis_framework::Dynamixel *> dxls, std::map<std::string, double> sensors)

dxls と sensors から現在のサーボとセンサの状態を取得

その他のメソッドで計算した目標関節角等の値をresult_[]にセットする

result_[“torso_y”]->goal_position_ = goal_joint_position_(27);

controllerで定められた制御周期で実行され、

サーボの状態取得と指令送信が行われる

thormang3_manipulation_module 各メソッドの説明

ManipulationModule::stop()

　　モジュールの利用が停止した時に呼ばれる

　　停止処理を書く

ManipulationModule::isRunning()

　　モジュールが実行されているかどうか確認する

GitHub Organization

ROBOTIS-GIT:

- 本社管理
- https://github.com/robotis-git

ROBOTIS-JAPAN-GIT:

- 日本支店管理
- https://github.com/robotis-japan-git

ドキュメント

Thormang3

- http://emanual.robotis.com/docs/en/platform/thormang3/introduction/

ROBOTIS-Framework

- http://emanual.robotis.com/docs/en/software/robotis_framework_packages/