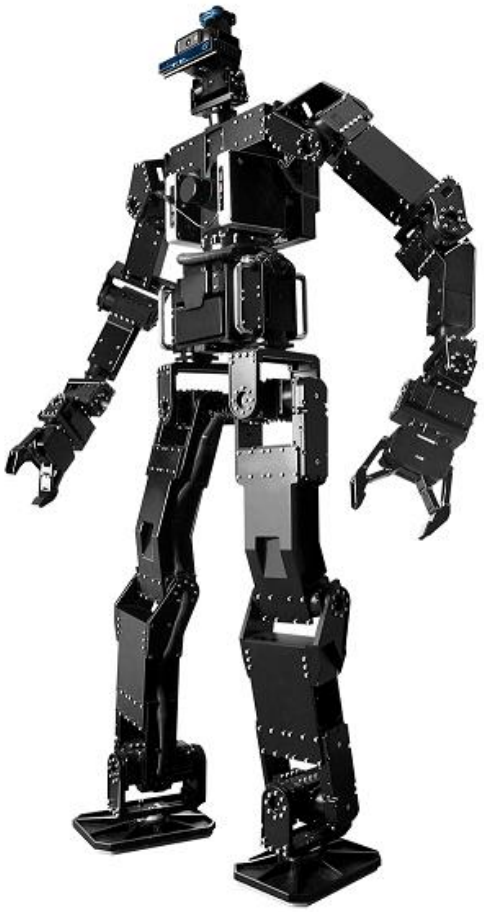


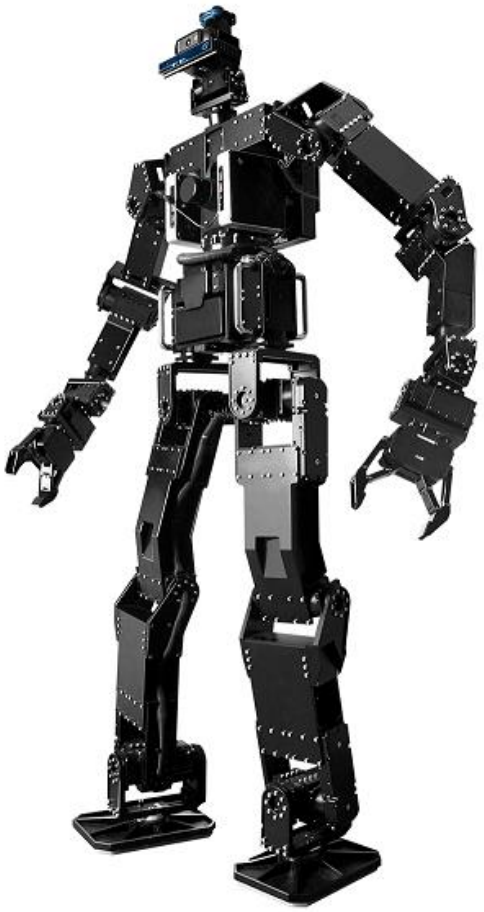
THORMANG3

THORMANG3 Tutorial

Module

Agenda

- Manipulation Module
- Gripper Module
- Walking Module
- Head Control Module
- Feet FT Module



THORMANG3

THORMANG3 Tutorial

Manipulation

Agenda

- **Robotis Library**
  - robotis_math
  - thormang3_kinamtcis_dynamics
- **Manipulation Module**
  - Overview
    - Structure
    - Files
  - Messages
  - Topic List
  - Programming Guide

Robotis Library

- **robotis_math**
  - This library includes basic operations which are used in motion modules.
  - These files describes the function related to coordinate transformation.
    - robotis_math_base.cpp
    - robotis_linear_algebra.cpp

Eigen::Vector3d getTransitionXYZ(double position_x, double position_y, double position_z)
Eigen::Matrix3d convertRPYToRotation(double roll, double pitch, double yaw)

  - These files illustrate the function for trajectory calculation.
    - robotis_trajectory_calculator.cpp
    - bezier_curve.cpp
    - fifth_order_polynomial_trajectory.cpp
    - simple_trapezoidal_velocity_profile.cpp

Eigen::MatrixXd calcMinimumJerkTra(double pos_start, double vel_start, double accel_start,
double pos_end, double vel_end, double accel_end,
double smp_time, double mov_time)

  - This function is used in walking module for step data
    - step_data_define.cpp

Robotis Library

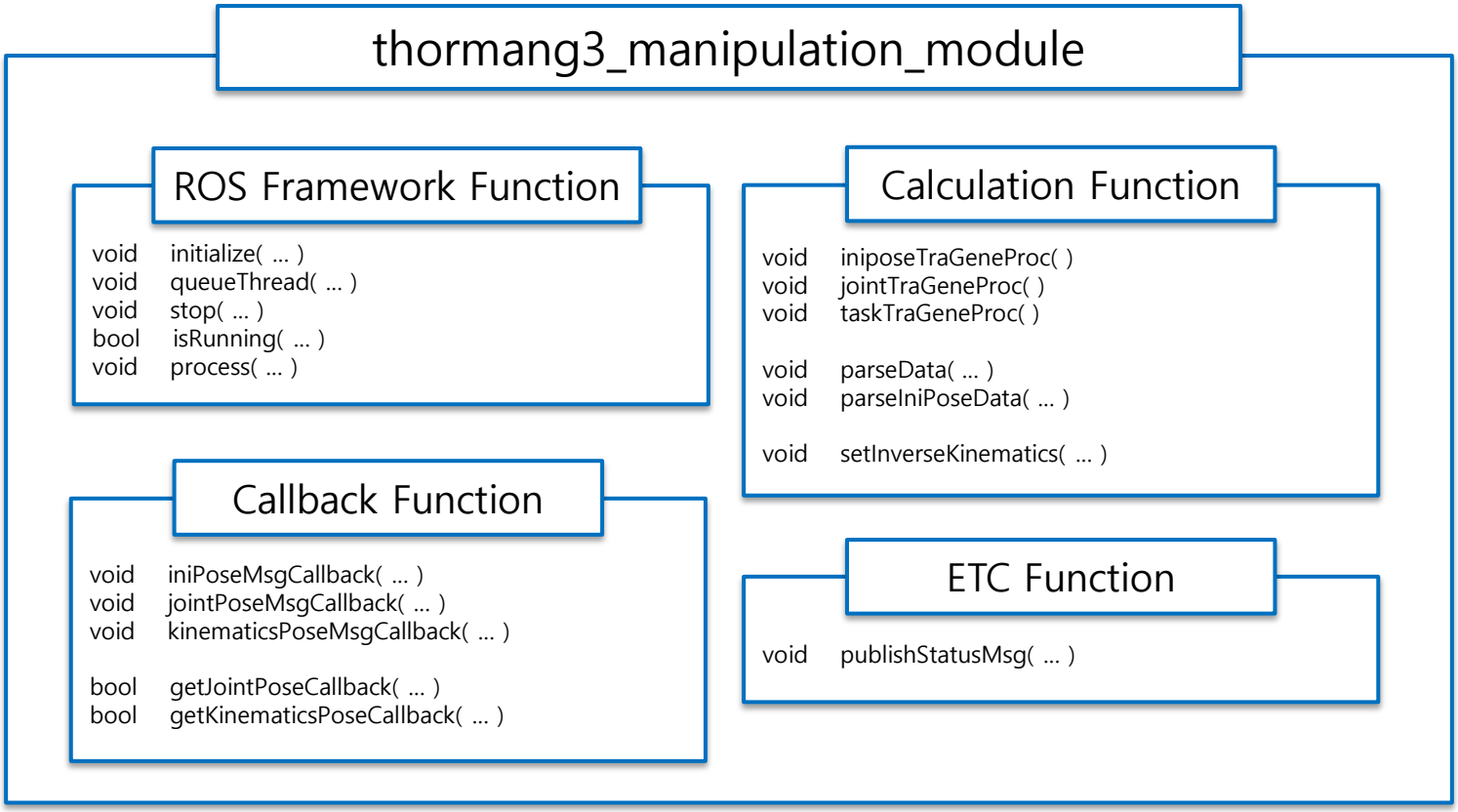
- **thormang3_kinematics_dynamics**
  - This library is included in ROBOTIS-THORMAG-MPC package.
  - It describes thormang3's kinematics & dynamics information [1].
  - We also provide robotics function such as forward & inverse kinematics [1].
  - For example, each joint information is written as

```
// right arm shoulder roll
thormang3_link_data_[3]->name_                  =  "r_arm_sh_r";
thormang3_link_data_[3]->parent_                =  1;
thormang3_link_data_[3]->sibling_               =  -1;
thormang3_link_data_[3]->child_                 =  5;
thormang3_link_data_[3]->mass_                  =  0.875;
thormang3_link_data_[3]->relative_position_     =  robotis_framework::getTransitionXYZ( 0.057 , -0.060 , -0.039 );
thormang3_link_data_[3]->joint_axis_            =  robotis_framework::getTransitionXYZ( -1.0 , 0.0 , 0.0 );
thormang3_link_data_[3]->center_of_mass_        =  robotis_framework::getTransitionXYZ( -0.060 , -0.002 , 0.000 );
thormang3_link_data_[3]->joint_limit_max_       =  0.5 * M_PI;
thormang3_link_data_[3]->joint_limit_min_       =  -0.5 * M_PI;
thormang3_link_data_[3]->inertia_
                    =  robotis_framework::getInertiaXYZ( 0.00043 , 0.00000 , 0.00000 , 0.00112 , 0.00000 , 0.00113 );
```

[1] Introduction to Humanoid Robotics | Shuuji Kajita | Springer

Manipulation Module

- **Overview**
  - Structure



Manipulation Module

- **Overview**
  - Files
    - ./src/manipulation_module.cpp
    - ./include/thormang3_manipulation_module/manipulation_module.h
    - ./config/ik_weight.yaml
      -> inverse kinematics weight value
    - ./config/ini_pose.yaml
      -> joint initial pose

Manipulation Module

- **Messages**
  - manipulation_module_msgs
    - msg
      - JointPose.msg
      - KinematicsPose.msg
    - srv
      - GetJointPose.srv
      - GetKinematicsPose.srv

Manipulation Module

- **Messages (msg)**
  - JointPose.msg
    - string name -> joint name to control
    - float64 value -> desired joint value [rad]
    - float64 time -> desired movement time [sec]

  - KinematicsPose.msg
    - string name -> group name such as left arm or right arm
    - geometry_msgs/Pose -> desired pose (position xyz and orientation xyzw)
    - float64 time -> desired movement time [sed]

Manipulation Module

- **Messages (srv)**
  - GetJointPose.srv
    - request :
      string joint_name -> If you request joint name,
    - response :
      float64 joint_value -> it returns present joint value [rad].
  - GetKinematicsPose.srv
    - request :
      string group_name -> If you request group name,
    - response :
      geometry_msgs/Pose group_pose -> it returns present group pose.

Manipulation Module

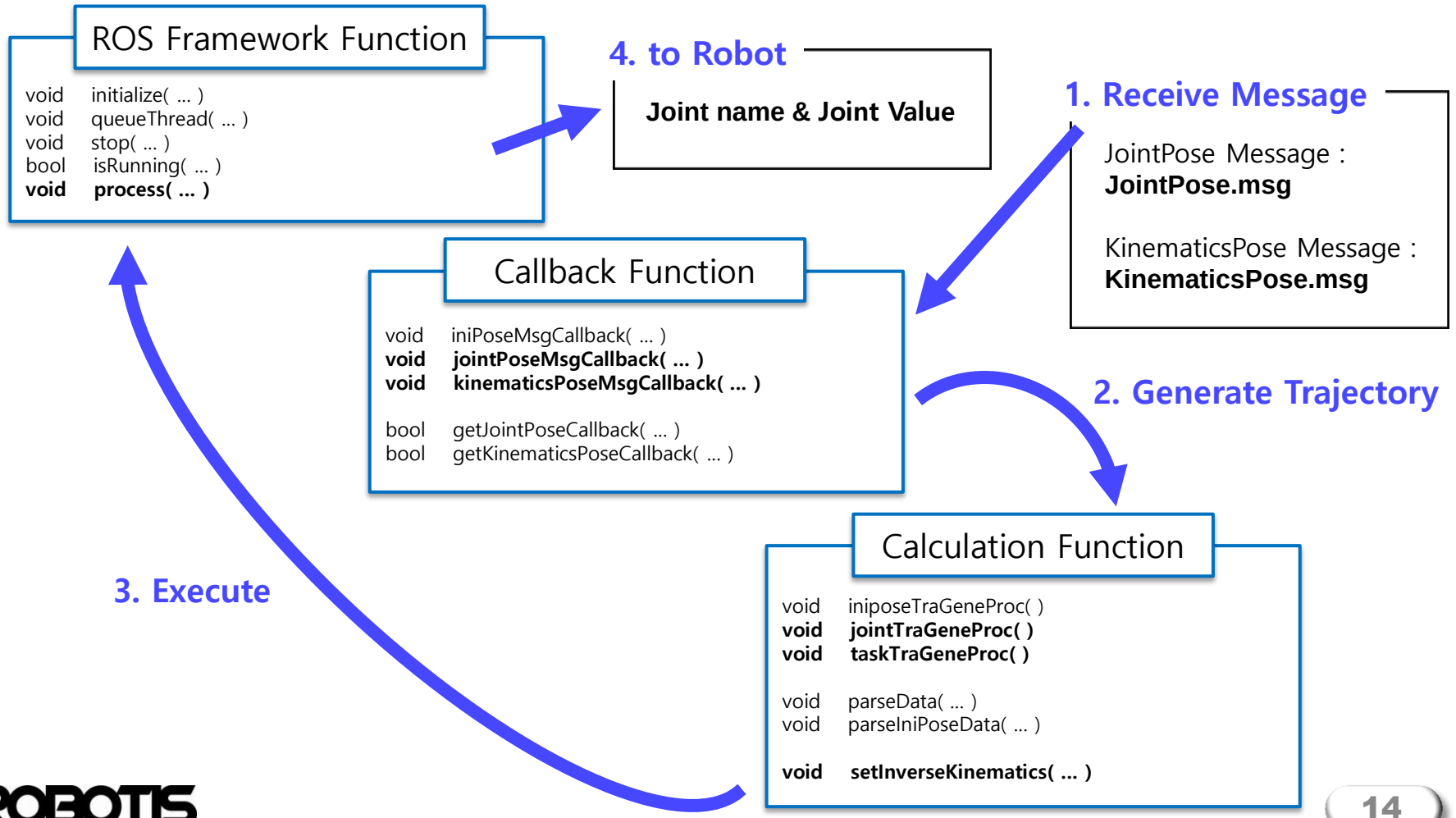
- **Topic List**

	Name		Description
Topic (Publish)	/robotis/status		publisher to send status
	/robotis/movement_done		publisher to send movement done
Topic (Subscribe)	/robotis/manipulation/ini_pose_msg		command for moving to initial pose
	/robotis/manipulation/joint_pose_msg		command for writing desired angle
	/robotis/manipulation/kinematics_pose_msg		command for writing desired end effector's pose
Service (Server)	/robotis/manipulation/get_joint_pose	req	name for user specified joint
		res	current angle if user specified joint
	/robotis/manipulation/get_kinemtacis_pose	req	name for user specified group
		res	current pose for user specified group

- **Programming guide**
  - Send desired pose
    - Joint pose
    - Kinematics pose
  - Get current pose
    - Read joint pose
    - Read kinematics pose

Manipulation Module

- **Programming guide**
  - Send desired pose



Manipulation Module

- **Programming guide**
  - Read current pose

ROS Framework Function

```
void    initialize( ... )
void    queueThread( ... )
void    stop( ... )
bool    isRunning( ... )
void    process( ... )
```

Callback Function

```
void    iniPoseMsgCallback( ... )
void    jointPoseMsgCallback( ... )
void    kinematicsPoseMsgCallback( ... )

bool    getJointPoseCallback( ... )
bool    getKinematicsPoseCallback( ... )
```

1. Receive Service

JointPose Service:
GetJointPose.srv

KinematicsPose Service :
GetKinematicsPose.srv

2. to GUI

Joint name & Joint Value

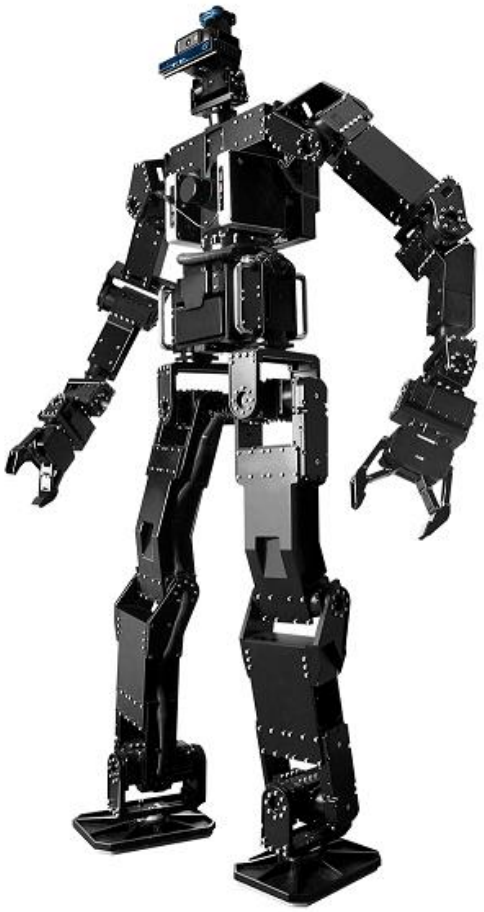
Group name & Kinematics Pose

Calculation Function

```
void    iniposeTraGeneProc( )
void    jointTraGeneProc( )
void    taskTraGeneProc( )

void    parseData( ... )
void    parseIniPoseData( ... )

void    setInverseKinematics( ... )
```



THORMANG3

THORMANG3 Tutorial

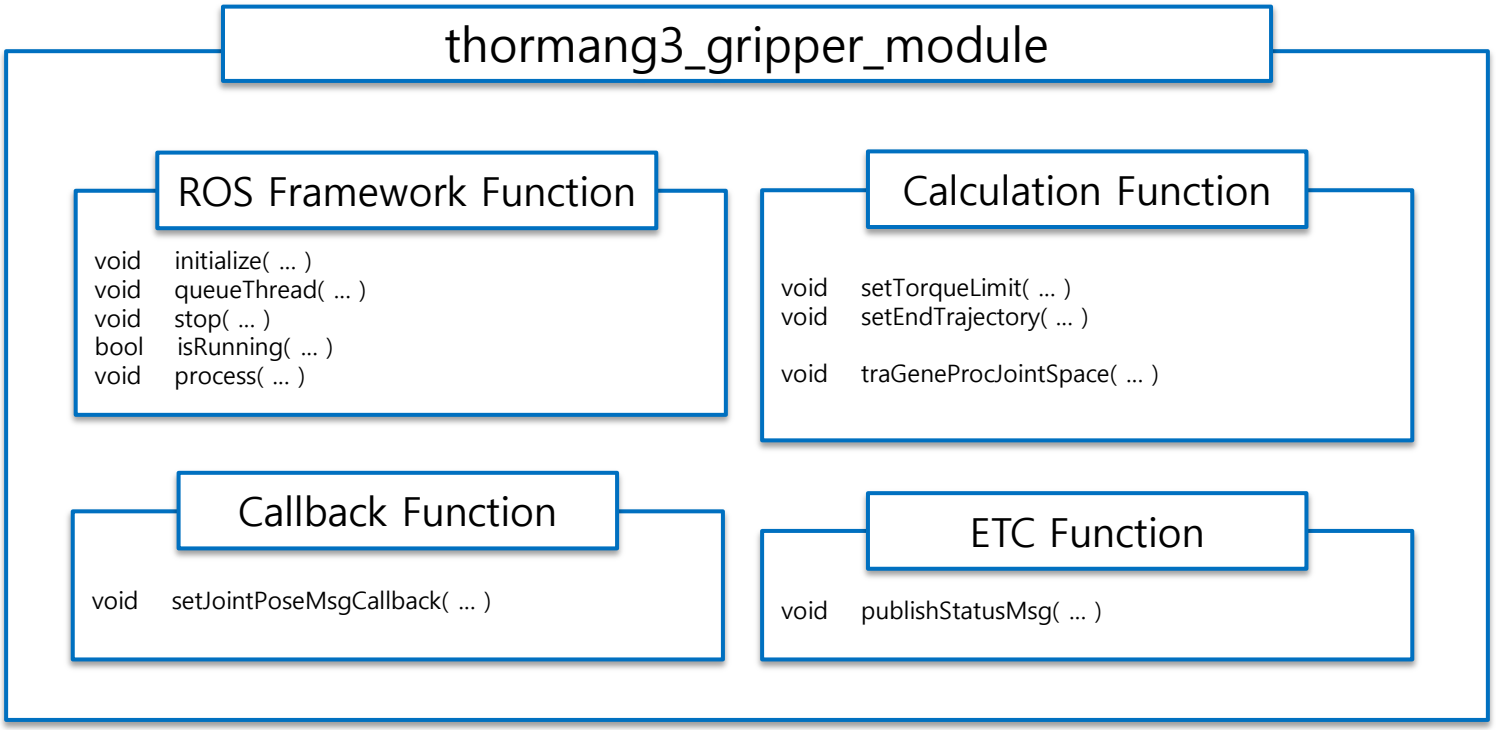
Gripper

Agenda

- **Gripper Module**
  - Overview
    - Structure
    - Files
  - Messages
  - Topic List
  - Programming Guide

Gripper Module

- **Overview**
  - Structure



Gripper Module

- **Overview**
  - Files
    - ./src/gripper_module.cpp
    - ./include/thormang3_gripper_module/gripper_module.h

Gripper Module

- **Messages**
  - In gripper module, we only use general ROS message.
    - msg
      - sensor_msgs/JointState msg

Gripper Module

- **Messages (msg)**
  - JointState.msg
    - string[] name -> joint name to control
    - float64[] position -> desired joint value [rad]
    - float64[] effort -> goal torque limit

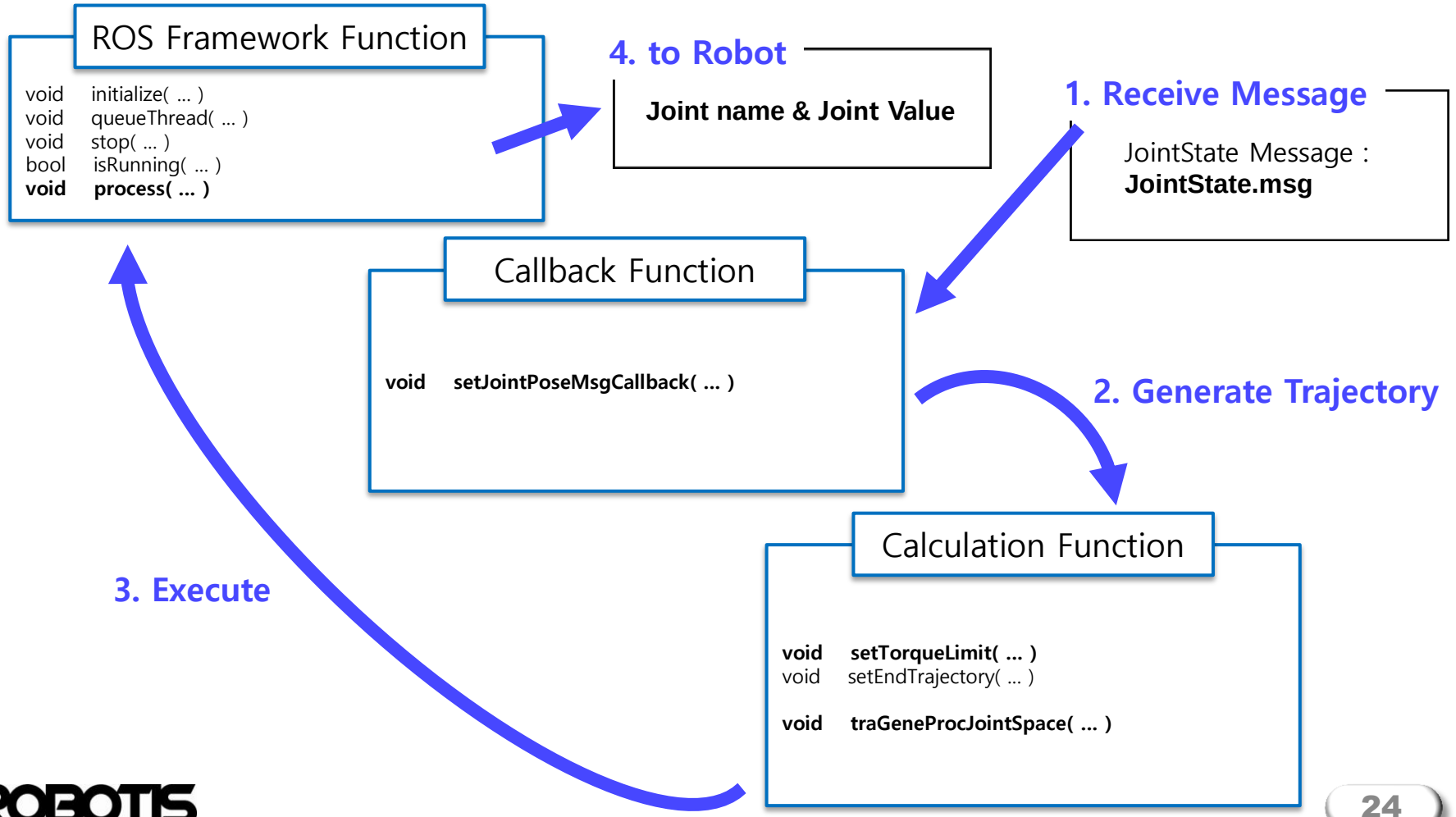
- **Topic List**

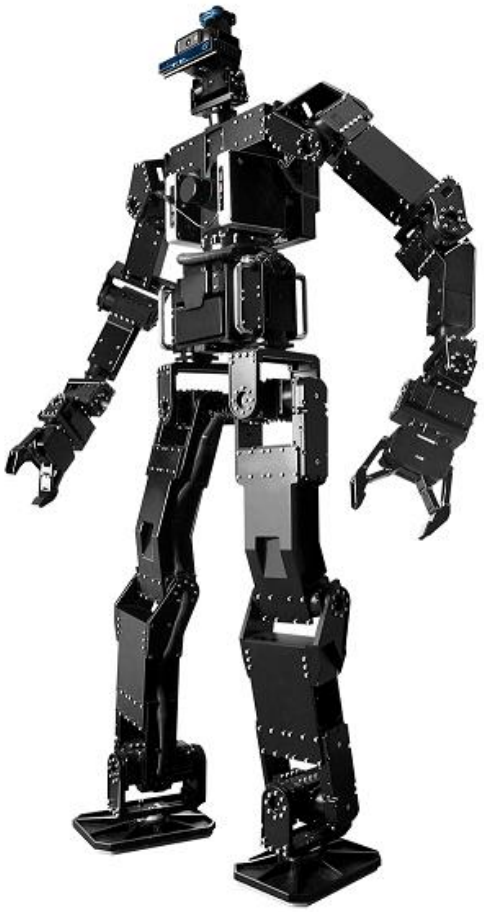
	Name	Description
Topic (Publish)	`/robotis/status`	publisher to send status
	`/robotis/movement_done`	publisher to send movement done
	`/robotis/sync_write_item`	Publisher to set goal torque limit
Topic (Subscribe)	`/robotis/gripper/joint_pose_msg`	command for moving to gripper

- **Programming guide**
  - Send desired pose
    - Joint pose

Gripper Module

- **Programming guide**
  - Send desired pose





THORMANG3

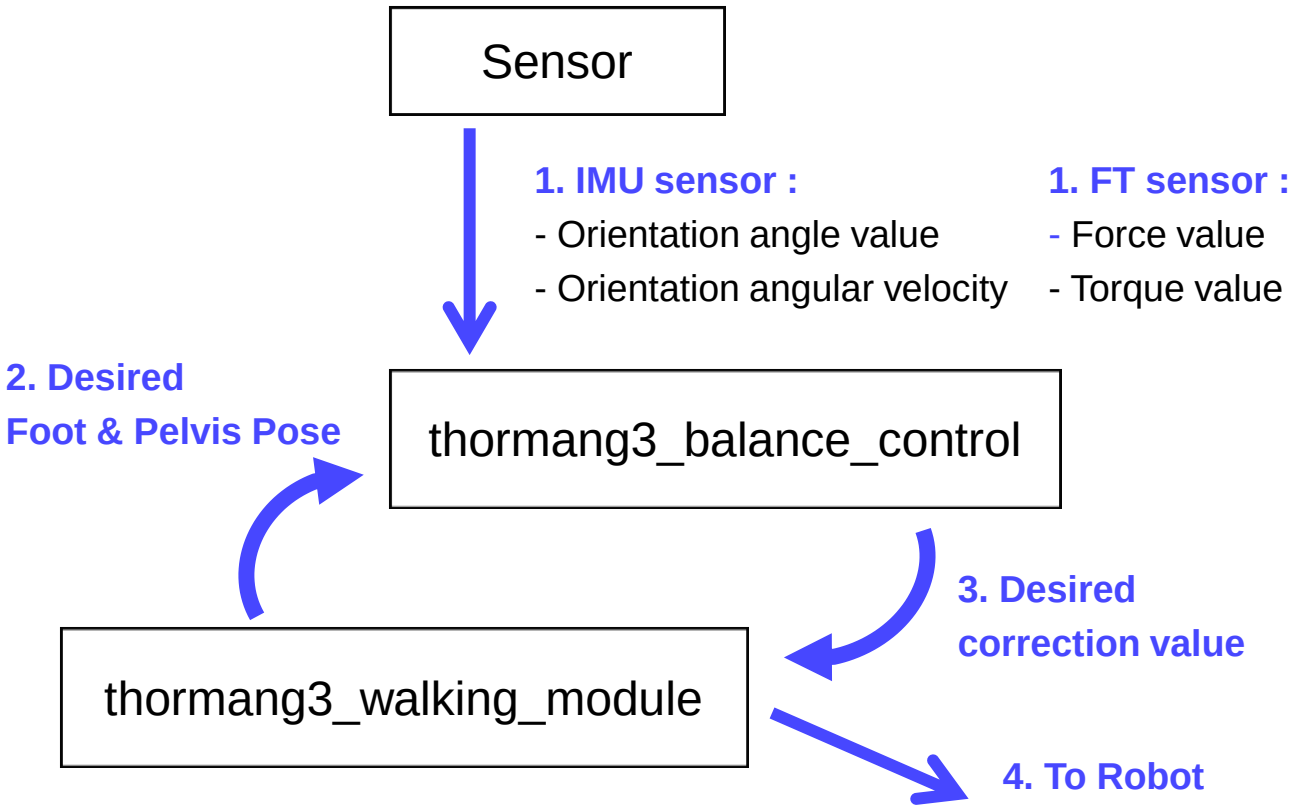
THORMANG3 Tutorial

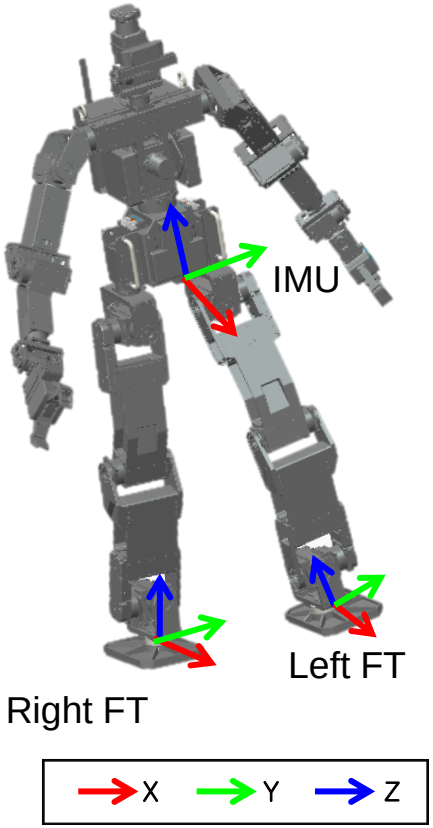
Walking

Agenda

- **Robotis Library**
  - thormang3_balance_control
- **Robotis ROS Package**
  - thormang3_foot_step_generator
- **Walking Module**
  - Overview
    - Structure
    - Files
  - Messages
  - Topic List
  - Programming Guide

- **thormang3_balance_control**
  - This library indicates sensory feedback algorithm for THORMANG3.
  - In sensory feedback, FT sensors and IMU sensor are used.
  - Our balance algorithm is based on PD control.



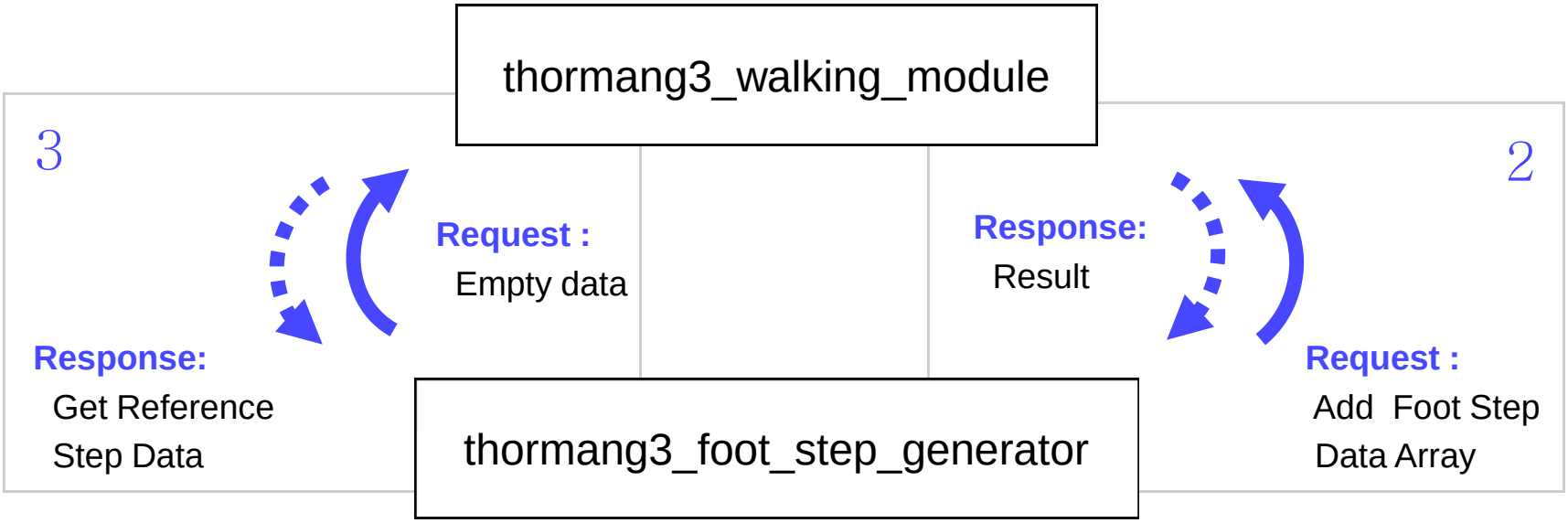


- **thormang3_foot_step_generator**
  - This package is used to generate foot step for thormang3's walking module

1. Make step data message

2. Send step data

3. Request reference step data



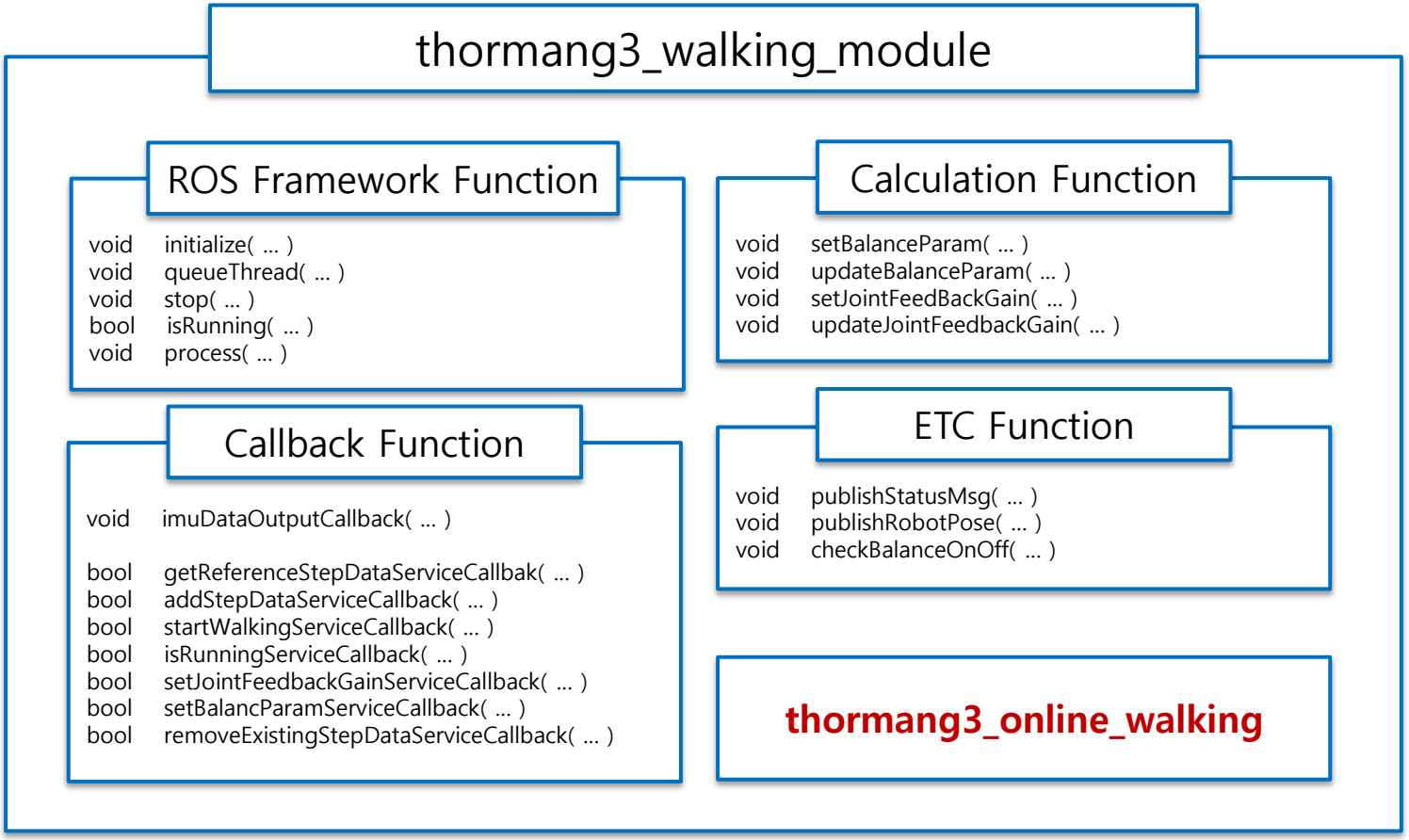
1. Calculation process

Robotis ROS Package

- **thormang3_foot_step_generator**
  - Messages
    - FootStepCommand.msg
    - Step2D.msg
    - Step2DArray.msg

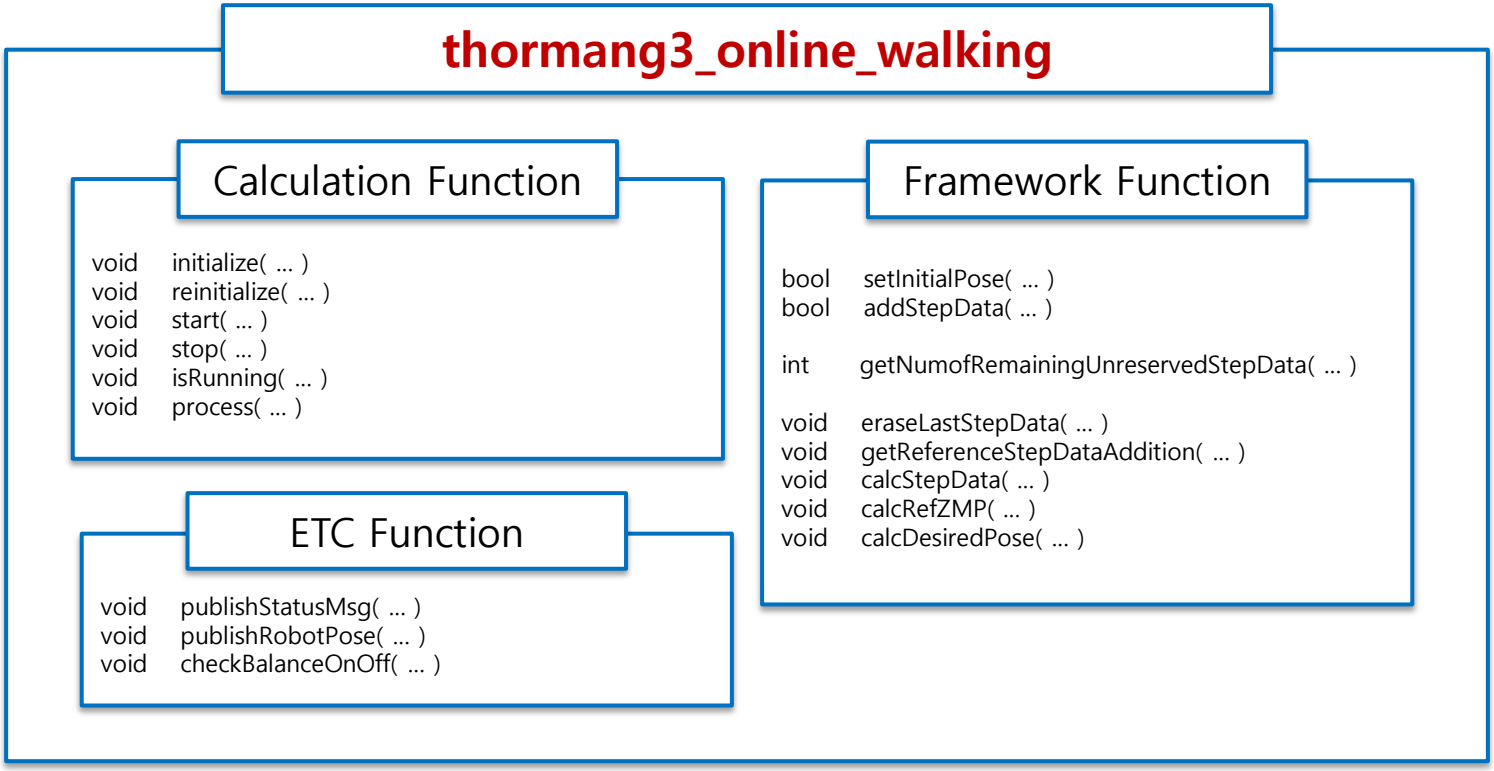
Walking Module

- **Overview**
  - Structure



Walking Module

- **Overview**
  - Structure
    - thormang3_online_walking is generally called as Dr. Kajita's preview walking control which is based on cart-table model



Walking Module

- **Overview**
  - Files
    - ./src/walking_module.cpp
    - ./src/thormang3_online_walking.cpp
    - ./include/thormang3_walking_module/walking_module.h
    - ./include/thormang3_walking_module/thormang3_online_walking.h

Walking Module

- **Messages**
  - manipulation_module_msgs
    - msg
      - BalanceParam.msg
      - DampingBalanceParam.msg
      - JonitFeedBackGain.msg
      - PoseXYZRPY.msg
      - PoseZRPY.msg
      - RobotPose.msg
      - StepData.msg
      - StepPositionData.msg
      - StepTimeData.msg
    - srv
      - AddStepDataArray.srv
      - GetReferenceStepData.srv
      - IsRunning.srv
      - RemoveExsitingStepData.srv
      - SetBalanceParam.srv
      - SetDampingBalanceParam.srv
      - SetJointFeedBackGain.srv
      - StartWalking.srv

Walking Module

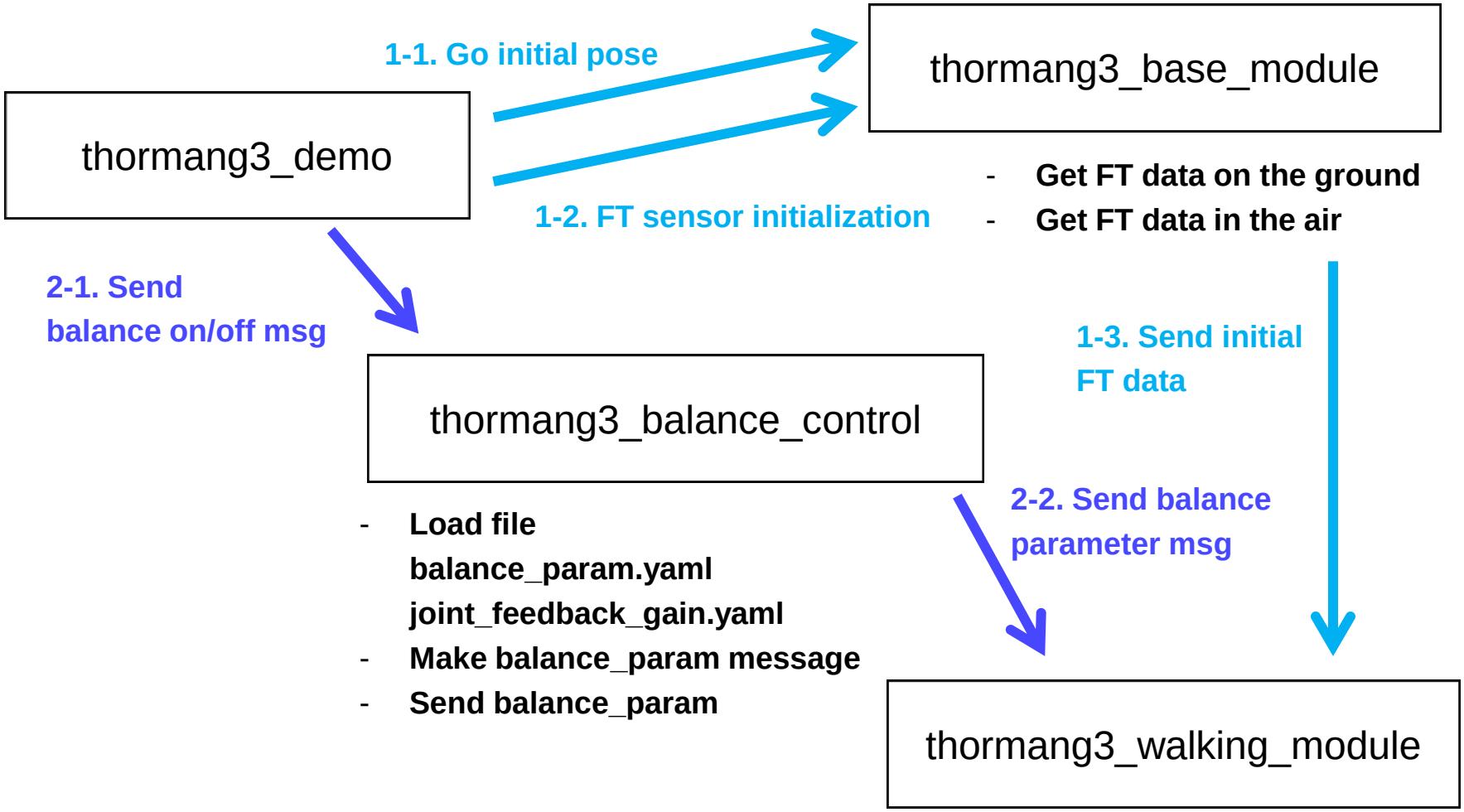
- **Topic List**

	Name		Description
Topic (Publish)	/robotis/walking/status_message		The status message from walking module
Service (Server)	/robotis/walking/get_reference_step_data	req	Empty
		res	Reference Step Data
	/robotis/walking/add_step_data	req	"Auto Start" and "Step Data Array"
		res	Processing Result for Request
	/robotis/walking/walking_start	req	Empty
		res	Processing Result for Request
	/robotis/walking/remove_existing_step_data	req	Empty
		res	Processing Result for Request
	/robotis/walking/set_balance_param	req	All of Desired Balancing Parameter
		res	Processing Result for Request
	/robotis/walking/is_running	req	Empty
		res	Running or Not

- **Programming guide**
  - Set Balance Parameter
  - Send Walking Command

Walking Module

- **Programming guide**
  - Set Balance Parameter



- **Programming guide**
  - balance_param.yaml

```
####### cob_offset #######
cob_x_offset_m : -0.015
cob_y_offset_m :  0.0

####### FeedForward #####
hip_roll_swap_angle_rad : 0.0

########## Gain ########
# by gyro
foot_roll_gyro_p_gain : 0.5
foot_roll_gyro_d_gain : 0.0

foot_pitch_gyro_p_gain : 0.5
foot_pitch_gyro_d_gain : 0.0


# by imu
foot_roll_angle_p_gain : 1.0
foot_roll_angle_d_gain : 0.1

foot_pitch_angle_p_gain : 1.0
foot_pitch_angle_d_gain : 0.1
```

```
# by ft sensor
foot_x_force_p_gain : 0.1
foot_x_force_d_gain : 0.0

foot_y_force_p_gain : 0.1
foot_y_force_d_gain : 0.0

foot_z_force_p_gain : 0.02
foot_z_force_d_gain : 0.0

foot_roll_torque_p_gain : 0.0015
foot_roll_torque_d_gain : 0.0

foot_pitch_torque_p_gain : 0.0015
foot_pitch_torque_d_gain : 0.0
```

```
########## CUT OFF FREQUENCY ##########
# by gyro
roll_gyro_cut_off_frequency  : 50.0
pitch_gyro_cut_off_frequency : 50.0

# by imu
roll_angle_cut_off_frequency  : 50.0
pitch_angle_cut_off_frequency : 50.0

# by ft sensor
foot_x_force_cut_off_frequency : 40.0
foot_y_force_cut_off_frequency : 40.0
foot_z_force_cut_off_frequency : 40.0
foot_roll_torque_cut_off_frequency : 40.0
foot_pitch_torque_cut_off_frequency : 40.0
```

- **Programming guide**
  - joint_feedback_gain.yaml

```
r_leg_hip_y_p_gain : 1.0        l_leg_hip_y_p_gain : 1.0
r_leg_hip_y_d_gain : 0.0        l_leg_hip_y_d_gain : 0.0

r_leg_hip_r_p_gain : 1.5        l_leg_hip_r_p_gain : 1.5
r_leg_hip_r_d_gain : 0.0        l_leg_hip_r_d_gain : 0.0

r_leg_hip_p_p_gain : 0.15       l_leg_hip_p_p_gain : 0.15
r_leg_hip_p_d_gain : 0.0        l_leg_hip_p_d_gain : 0.0

r_leg_kn_p_p_gain  : 0.15       l_leg_kn_p_p_gain  : 0.15
r_leg_kn_p_d_gain  : 0.0        l_leg_kn_p_d_gain  : 0.0

r_leg_an_p_p_gain  : 0.05       l_leg_an_p_p_gain  : 0.05
r_leg_an_p_d_gain  : 0.0        l_leg_an_p_d_gain  : 0.0

r_leg_an_r_p_gain  : 0.05       l_leg_an_r_p_gain  : 0.05
r_leg_an_r_d_gain  : 0.0        l_leg_an_r_d_gain  : 0.0
```

Walking Module

- **Programming guide**
  - Send Walking Command

thormang3_demo

1. Send Message

FootStepCommand.msg or Step2DArray.msg

thormang3_foot_step_generator

- **Make step data msg**
- **Send step data msg**
- **Request reference step data**

2. Add foot step data

thormang3_walking_module

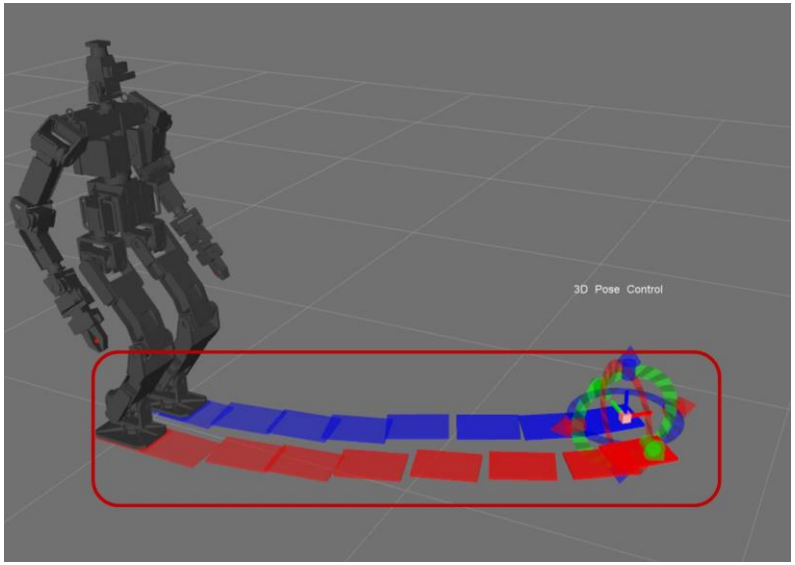
- **Programming guide**
  - Send Walking Command
    - FootStepCommand.msg
      - string command -> walking direction
      - int32 step_num -> step number
      - float64 step_length -> step length
      - float64 side_step_length -> side step length
      - float64 step_angle_rad -> step angle when the robot turn left or right

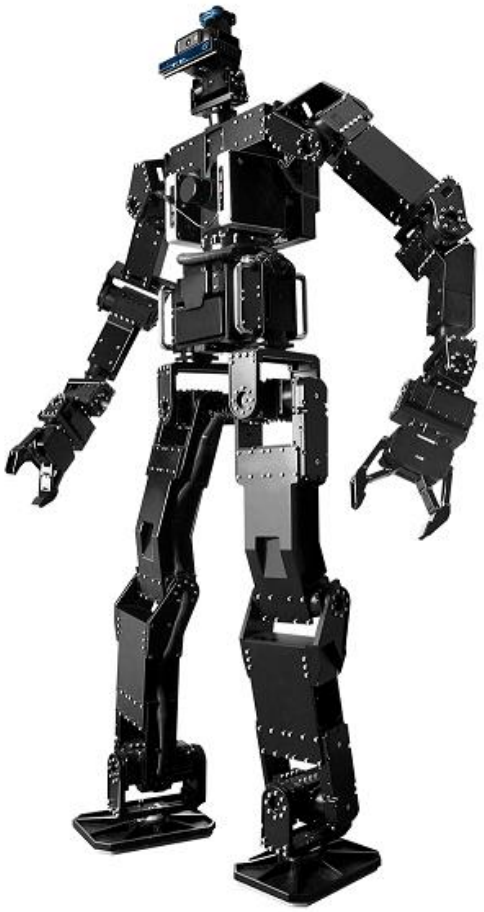
Programming example : go forward

```
thormang3_foot_step_generator::FootStepCommand msg;
msg.command = “forward”;
msg.step_num = 3;
msg.step_length = 0.05;
msg.side_step_length = 0.0;
msg.step_angle_rad = 0.0;
```

Walking Module

- **Programming guide**
  - Send Walking Command
    - Step2DArray.msg
      - Step2D[] footsteps_2d -> foot step array
    - Step2D.msg
      - geometry_msgs/Pose2D step2d -> foot step pose
      - unit8 moving_foot -> moving state
      - unit8 LEFT_FOOT_SWING = 1
      - unit8 RIGHT_FOOT_SWING = 2
      - unit8 STANDING = 3





THORMANG3

THORMANG3 Tutorial

Head Control

Agenda

- **Robotis Library**
  - robotis_math
- **Robotis ROS Package**
  - thormang3_sensors
- **Head Control Module**
  - Overview
    - Structure
    - Files
  - Messages
  - Topic List
  - Programming Guide

Robotis Library

- **robotis_math**
  - This library includes basic operations which are used in motion modules.
  - These files describes the function related to coordinate transformation.
    - robotis_math_base.cpp
    - robotis_linear_algebra.cpp

Eigen::Vector3d getTransitionXYZ(double position_x, double position_y, double position_z)
Eigen::Matrix3d convertRPYToRotation(double roll, double pitch, double yaw)

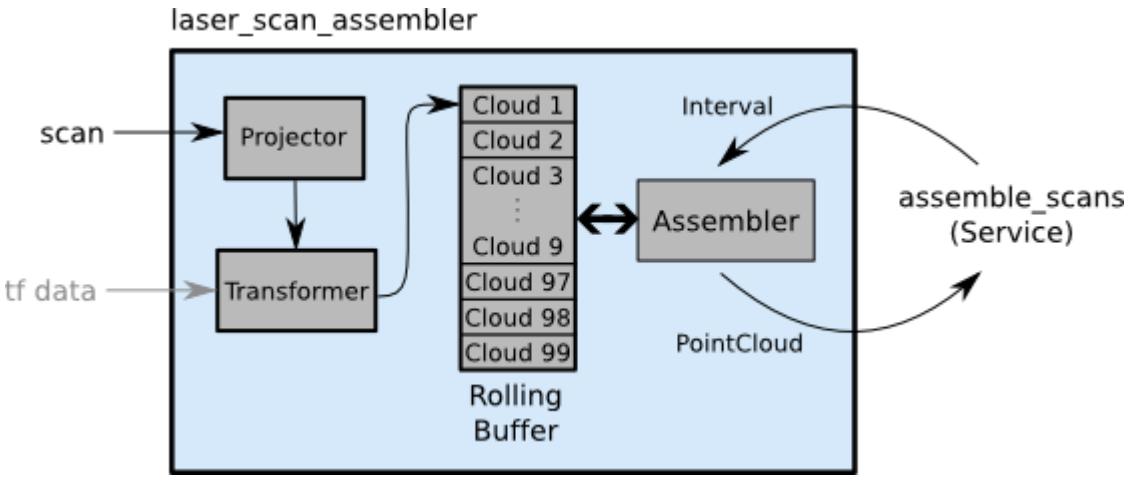
  - These files illustrate the function for trajectory calculation.
    - robotis_trajectory_calculator.cpp
    - bezier_curve.cpp
    - fifth_order_polynomial_trajectory.cpp
    - simple_trapezoidal_velocity_profile.cpp

Eigen::MatrixXd calcMinimumJerkTra(double pos_start, double vel_start, double accel_start,
double pos_end, double vel_end, double accel_end,
double smp_time, double mov_time)

  - This function is used in walking module for step data
    - step_data_define.cpp

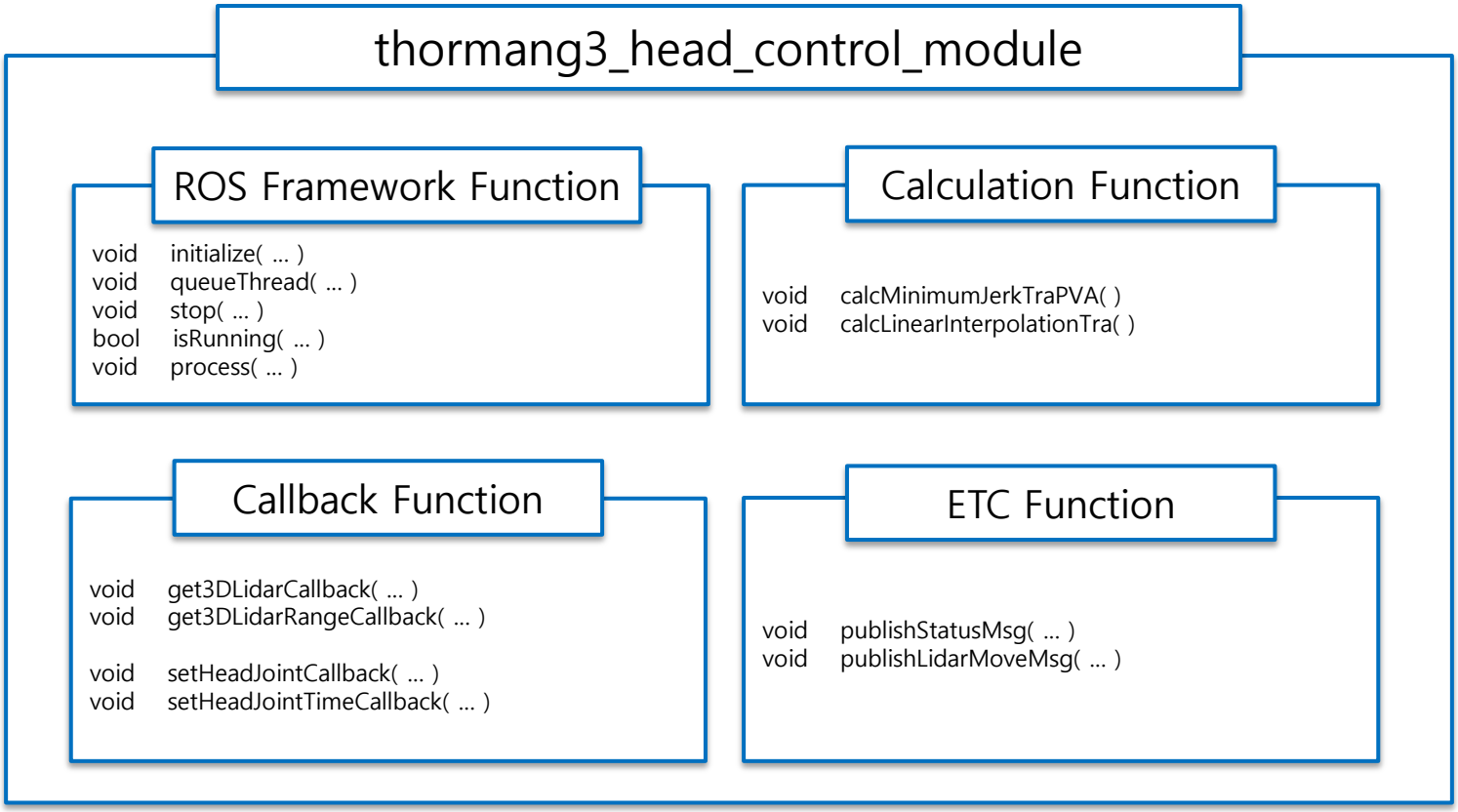
- **thormang3_sensor**
  - This package is used to assembling Laserscan(LaserScan to PointCloud)

1. Receive start message
2. Store time of getting start message
3. Receive end message
4. Assemble LaserScan using [laser_scan_assembler node]



Head Control Module

- **Overview**
  - Structure



Head Control Module

- **Overview**
  - Files
    - ./src/head_control_module.cpp
    - ./include/thormang3_head_control_module/head_control_module.h

Head Control Module

- **Messages**
  - head_control_module_msgs
    - msg
      - HeadJointPose.msg
  - sensor_msgs
    - msg
      - JointState.msg
  - std_msgs
    - msg
      - String.msg
      - Float64.msg

Head Control Module

- **Messages (msg)**
  - HeadJointPose.msg
    - float64 mov_time -> time to move
    - sensor_msgs/JointState angle -> desired joint value of head [rad]
  - JointState.msg
    - String[] name -> joint name
    - float64[] position -> desired joint value [rad]
    - float64[] velocity -> not used
    - float64[] effort -> not used

Head Control Module

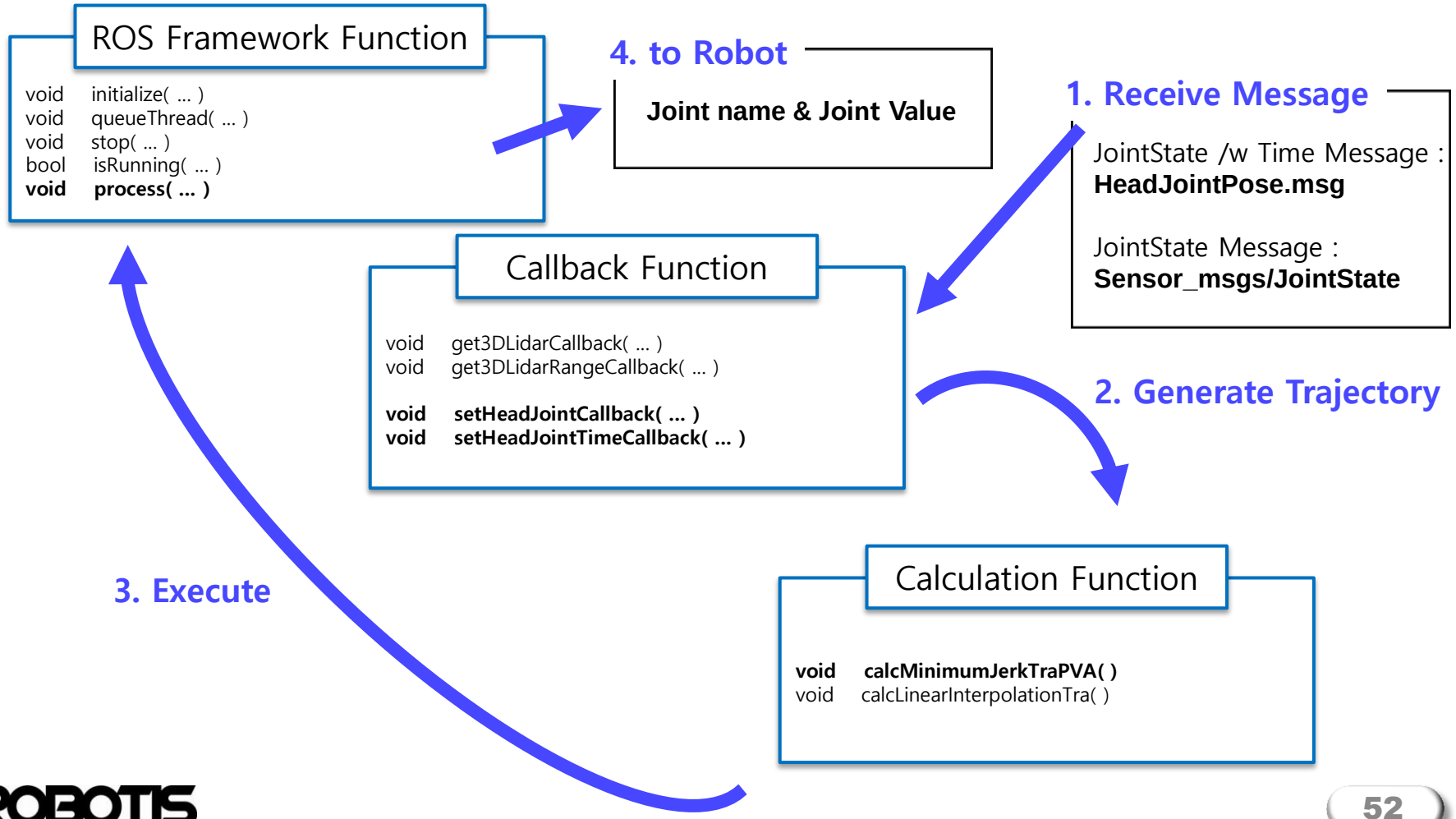
- **Topic List**

	Name	Description
Topic (Publish)	/robotis/sensor/move_lidar	Send a topic(start/end time of movement) to the assemble_lidar_node
Topic (Subscribe)	/robotis/head_control/move_lidar	Command to move head pitch joint by a given amount in order to make pointcloud
	/robotis/head_control/move_lidar_with_range	Command to move head pitch joint in order to make pointcloud
	/robotis/head_control/set_joint_states	Command to move joints of head
	/robotis/head_control/set_joint_states_time	Command to move joints of head in given time

- **Programming guide**
  - Send desired pose of head joints
    - Set head joints
    - Set head joints with moving time
  - Send command to make pointcloud
    - Make pointcloud of full range
    - Make pointcloud of given range

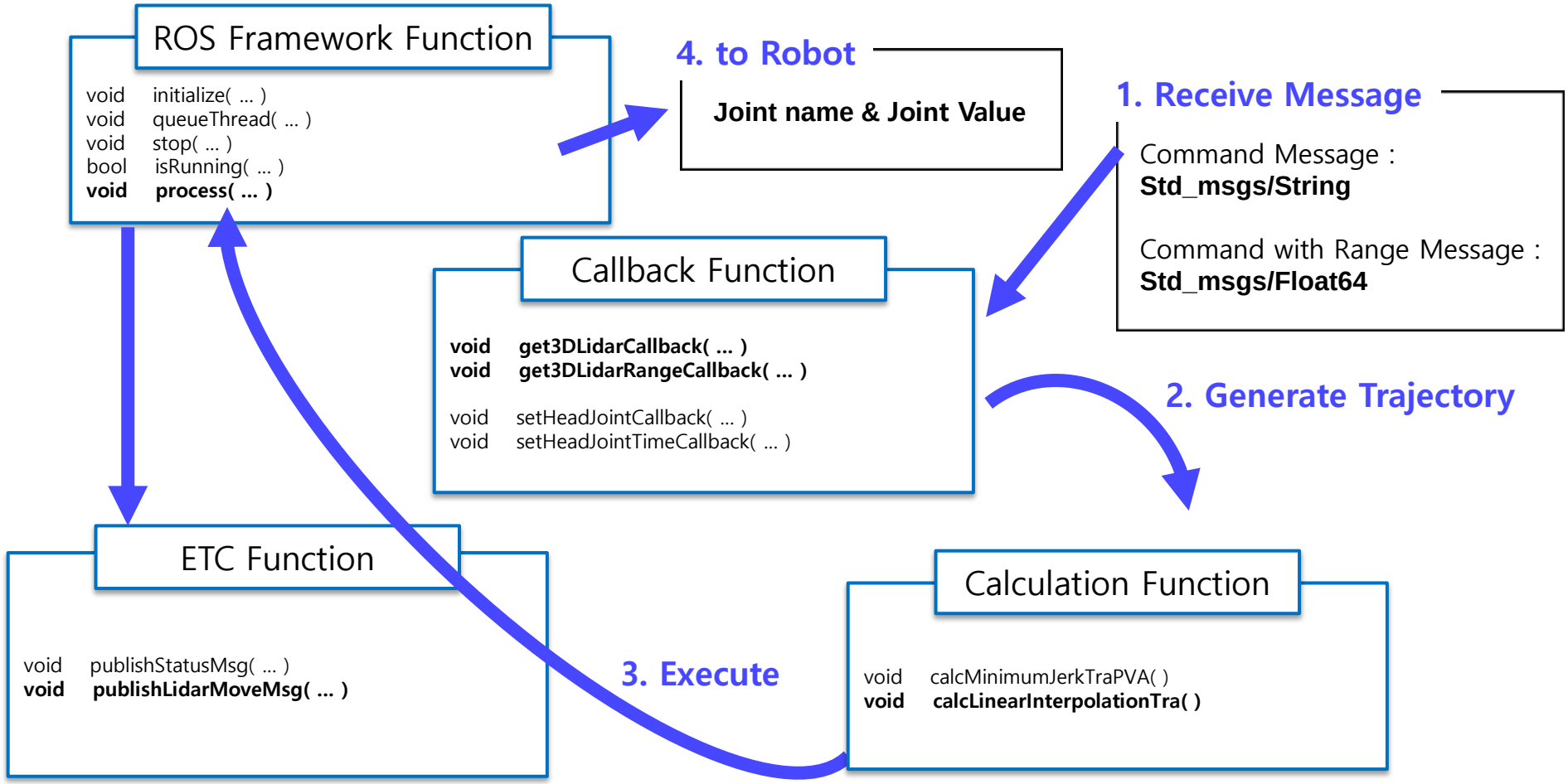
Head Control Module

- **Programming guide**
  - Send desired pose of head joints



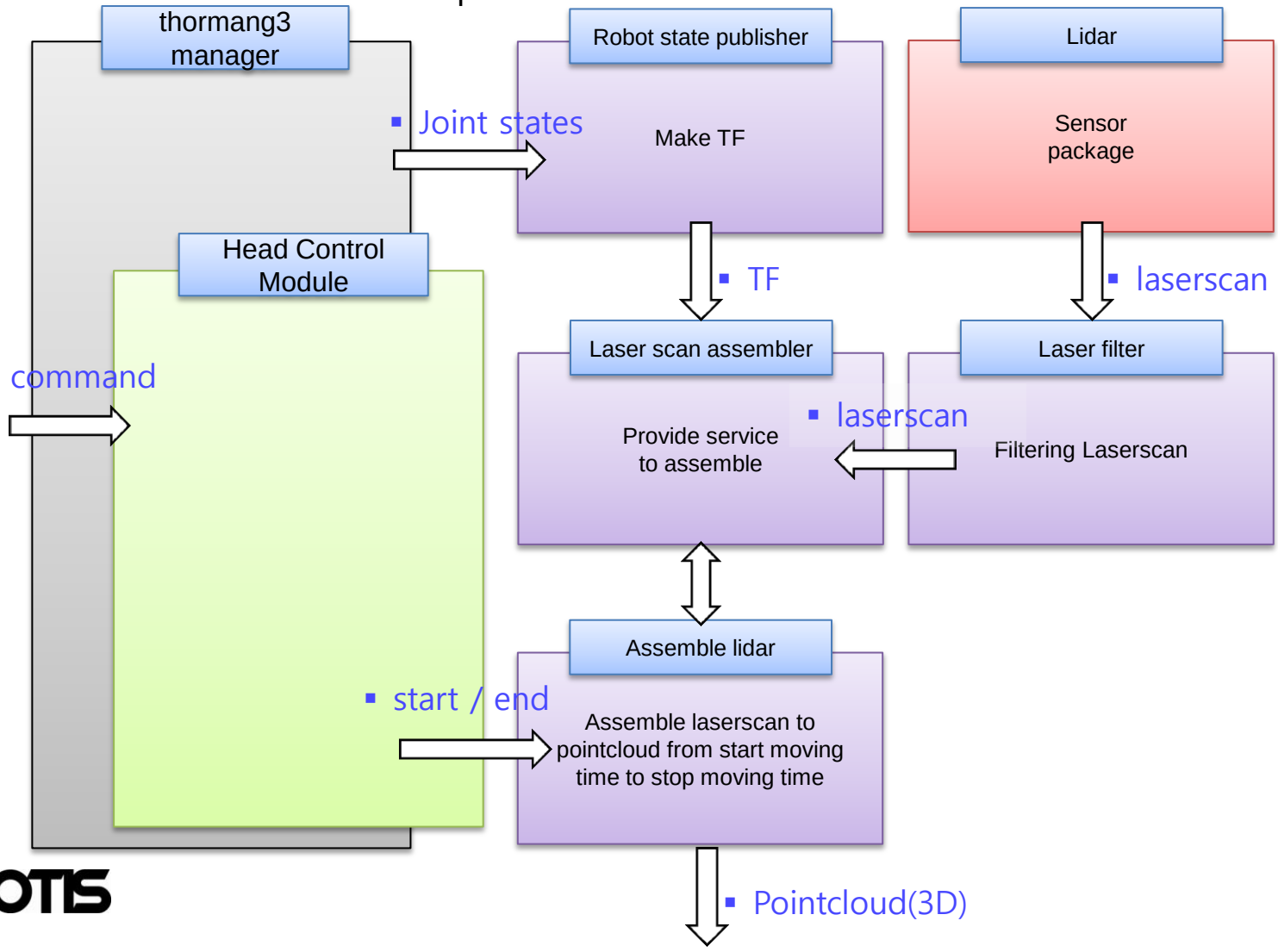
Head Control Module

- **Programming guide**
  - Send command to make pointcloud



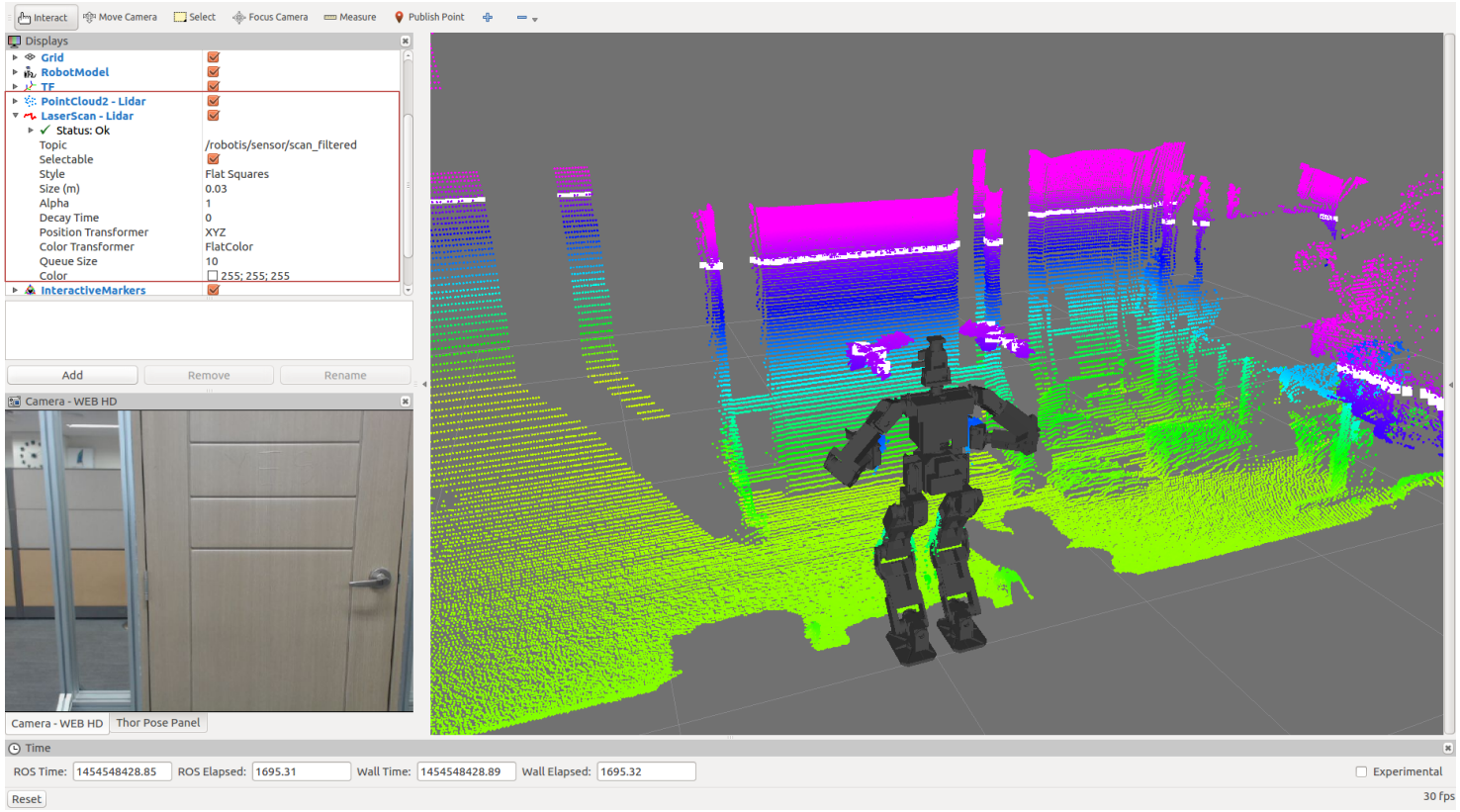
Head Control Module

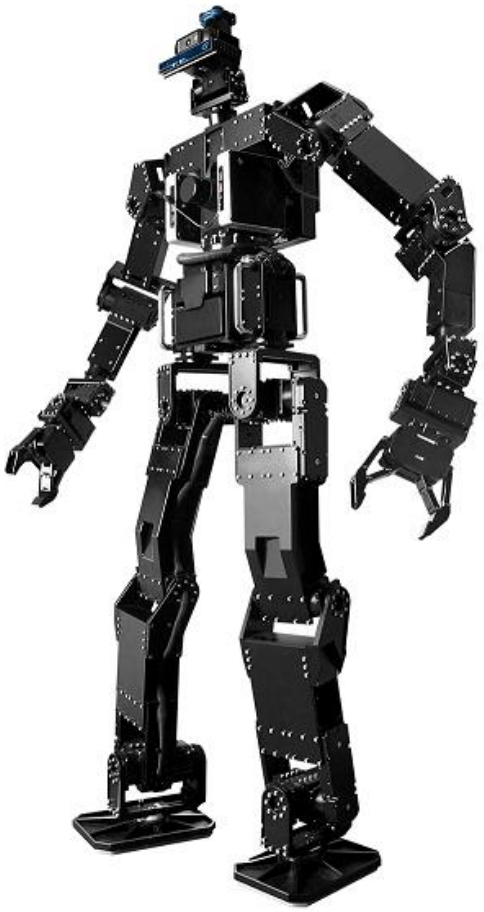
- **Programming guide**
  - Send command to make pointcloud



Head Control Module

- **Programming guide**
  - Send command to make pointcloud
    - White line : LaserScan (lidar)
    - Colorful line : PointCloud





THORMANG3

THORMANG3 Tutorial

Feet FT

Agenda

- **Feet FT Module**
  - Hardware
  - Overview
    - Structure
    - Files
  - Messages
  - Topic List
  - Programming Guide

THORMANG3 FT Sensor

- **Hardware**

1 FT Sensor

↓ Analogue Signal

2 DGPIO Board — For amplification

↓ Analogue Signal

3 Dynamixel's external port

Ankle roll : 4 lines

Ankle pitch : 2 lines

THORMANG3 FT Sensor

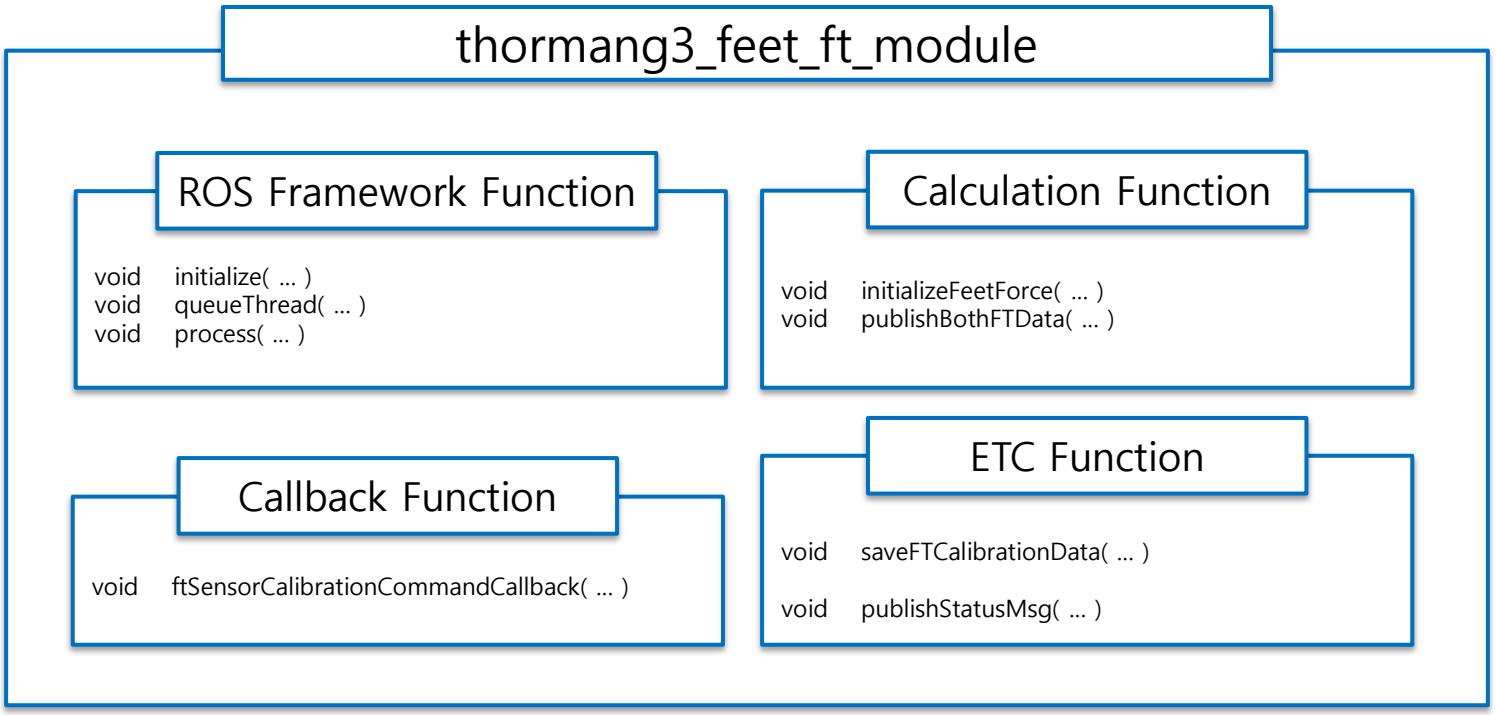
- **Hardware**
  - thormang3_manager/config/THORMANG3.robot

```
[ port info ]
# PORT NAME   | BAUDRATE | DEFAULT JOINT
/dev/ttyUSB0 | 2000000  | r_arm_sh_p1
/dev/ttyUSB1 | 2000000  | l_arm_sh_p1
/dev/ttyUSB2 | 2000000  | r_leg_hip_y
/dev/ttyUSB3 | 2000000  | l_leg_hip_y

[ device info ]
# TYPE     | PORT NAME     | ID  | MODEL           | PROTOCOL | DEV NAME      | BULK READ ITEMS
dynamixel  | /dev/ttyUSB0  | 1   | H54-100-S500-R  | 2.0      | r_arm_sh_p1   | present_position, present_voltage
dynamixel  | /dev/ttyUSB1  | 2   | H54-100-S500-R  | 2.0      | l_arm_sh_p1   | present_position, present_voltage
dynamixel  | /dev/ttyUSB0  | 3   | H54-100-S500-R  | 2.0      | r_arm_sh_r    | present_position, present_voltage
dynamixel  | /dev/ttyUSB1  | 4   | H54-100-S500-R  | 2.0      | l_arm_sh_r    | present_position, present_voltage
dynamixel  | /dev/ttyUSB0  | 5   | H54-100-S500-R  | 2.0      | r_arm_sh_p2   | present_position, present_voltage
dynamixel  | /dev/ttyUSB1  | 6   | H54-100-S500-R  | 2.0      | l_arm_sh_p2   | present_position, present_voltage
dynamixel  | /dev/ttyUSB0  | 7   | H54-100-S500-R  | 2.0      | r_arm_el_y    | present_position, present_voltage
dynamixel  | /dev/ttyUSB1  | 8   | H54-100-S500-R  | 2.0      | l_arm_el_y    | present_position, present_voltage
dynamixel  | /dev/ttyUSB0  | 9   | H42-20-S300-R   | 2.0      | r_arm_wr_r    | present_position, present_voltage
dynamixel  | /dev/ttyUSB1  | 10  | H42-20-S300-R   | 2.0      | l_arm_wr_r    | present_position, present_voltage
dynamixel  | /dev/ttyUSB0  | 11  | H42-20-S300-R   | 2.0      | r_arm_wr_y    | present_position, present_voltage
dynamixel  | /dev/ttyUSB1  | 12  | H42-20-S300-R   | 2.0      | l_arm_wr_y    | present_position, present_voltage
dynamixel  | /dev/ttyUSB0  | 13  | H42-20-S300-R   | 2.0      | r_arm_wr_p    | present_position, present_voltage
dynamixel  | /dev/ttyUSB1  | 14  | H42-20-S300-R   | 2.0      | l_arm_wr_p    | present_position, present_voltage
dynamixel  | /dev/ttyUSB2  | 15  | H54-100-S500-R  | 2.0      | r_leg_hip_y   | present_position, present_voltage
dynamixel  | /dev/ttyUSB3  | 16  | H54-100-S500-R  | 2.0      | l_leg_hip_y   | present_position, present_voltage
dynamixel  | /dev/ttyUSB2  | 17  | H54-200-S500-R  | 2.0      | r_leg_hip_r   | present_position, present_voltage
dynamixel  | /dev/ttyUSB3  | 18  | H54-200-S500-R  | 2.0      | l_leg_hip_r   | present_position, present_voltage
dynamixel  | /dev/ttyUSB2  | 19  | H54-200-B500-R  | 2.0      | r_leg_hip_p   | present_position, present_voltage
dynamixel  | /dev/ttyUSB3  | 20  | H54-200-B500-R  | 2.0      | l_leg_hip_p   | present_position, present_voltage
dynamixel  | /dev/ttyUSB2  | 21  | H54-200-S500-R  | 2.0      | r_leg_kn_p    | present_position, present_voltage
dynamixel  | /dev/ttyUSB3  | 22  | H54-200-S500-R  | 2.0      | l_leg_kn_p    | present_position, present_voltage
dynamixel  | /dev/ttyUSB2  | 23  | H54-200-B500-R  | 2.0      | r_leg_an_p    | present_position, present_voltage, external_port_data_1, external_port_data_2
dynamixel  | /dev/ttyUSB3  | 24  | H54-200-B500-R  | 2.0      | l_leg_an_p    | present_position, present_voltage, external_port_data_1, external_port_data_2
dynamixel  | /dev/ttyUSB2  | 25  | H54-200-S500-R  | 2.0      | r_leg_an_r    | present_position, present_voltage, external_port_data_1, external_port_data_2, external_port_data_3, external_port_data_4
dynamixel  | /dev/ttyUSB3  | 26  | H54-200-S500-R  | 2.0      | l_leg_an_r    | present_position, present_voltage, external_port_data_1, external_port_data_2, external_port_data_3, external_port_data_4
dynamixel  | /dev/ttyUSB0  | 27  | H54-100-S500-R  | 2.0      | torso_y       | present_position, present_voltage
dynamixel  | /dev/ttyUSB1  | 28  | H42-20-S300-R   | 2.0      | head_y        | present_position, present_voltage
dynamixel  | /dev/ttyUSB1  | 29  | H42-20-S300-R   | 2.0      | head_p        | present_position, present_voltage
dynamixel  | /dev/ttyUSB1  | 30  | GRIPPER         | 2.0      | l_arm_grip    | present_position, present_voltage
dynamixel  | /dev/ttyUSB0  | 31  | GRIPPER         | 2.0      | r_arm_grip    | present_position, present_voltage
```

Feet FT Module

- **Overview**
  - Structure



Feet FT Module

- **Overview**
  - Files
    - ./src/feet_force_torque_sensor_module.cpp
    - ./include/thormang3_feet_ft_module/feet_force_torque_sensor_module.h

Feet FT Module

- **Messages**
  - msg
    - BothWrench.msg

Feet FT Module

- **Messages (msg)**
  - BothWrench.msg
    - string name -> ft sensor value on the ground or in the air
    - geometry_msgs/Wrench right -> right foot's ft sensor value
    - geometry_msgs/Wrench left -> left foot's ft sensor value

Feet FT Module

- **Topic List**

	Name	Description
Topic (Publish)	/robotis/status	publisher to send status
	/robotis/feet_ft/both_ft_value	publisher to send ft sensor's value
Topic (Subscribe)	/robotis/feet_ft/ft_calib_command	command for ft sensor calibration