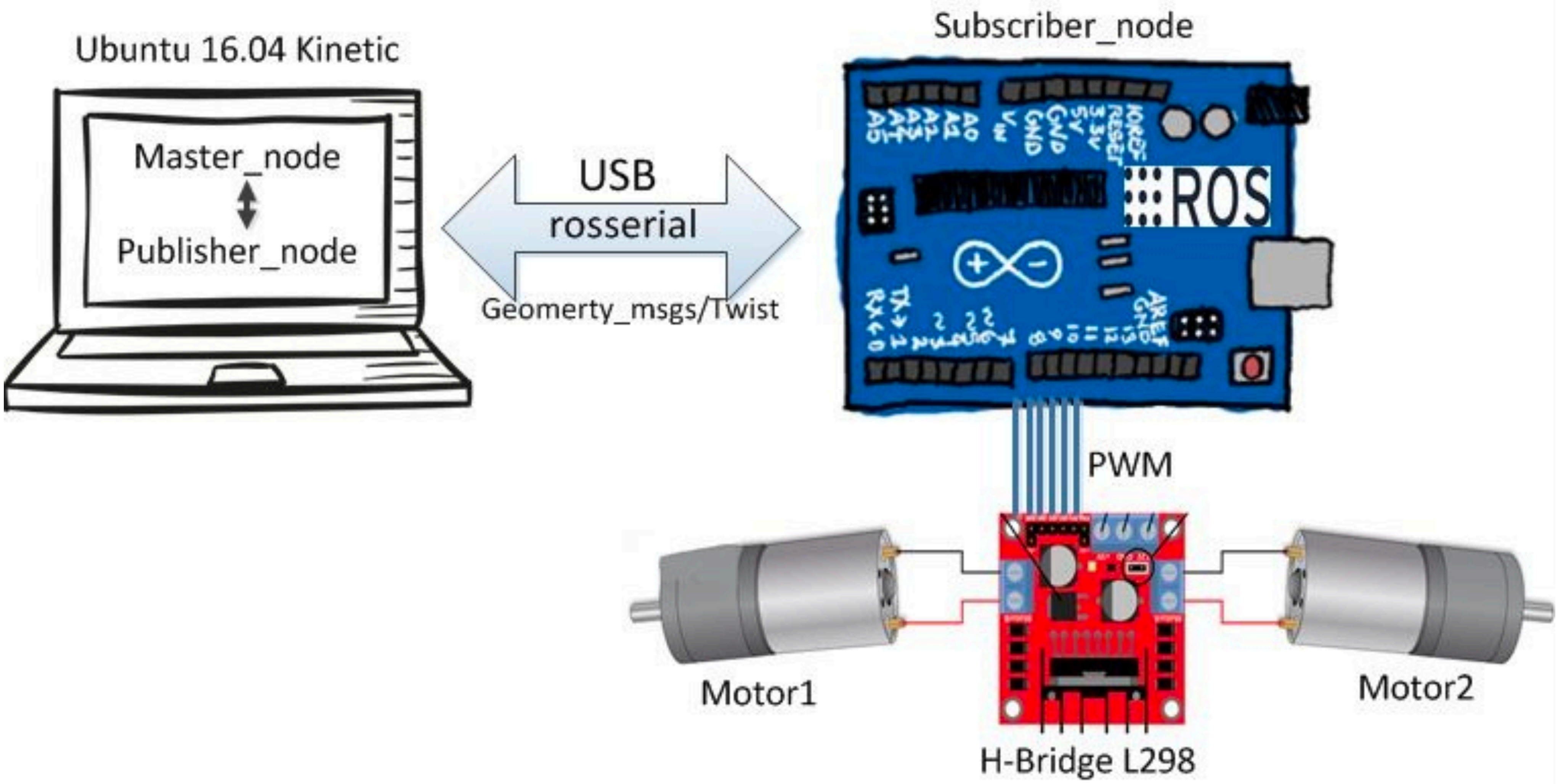
ROS 시리얼

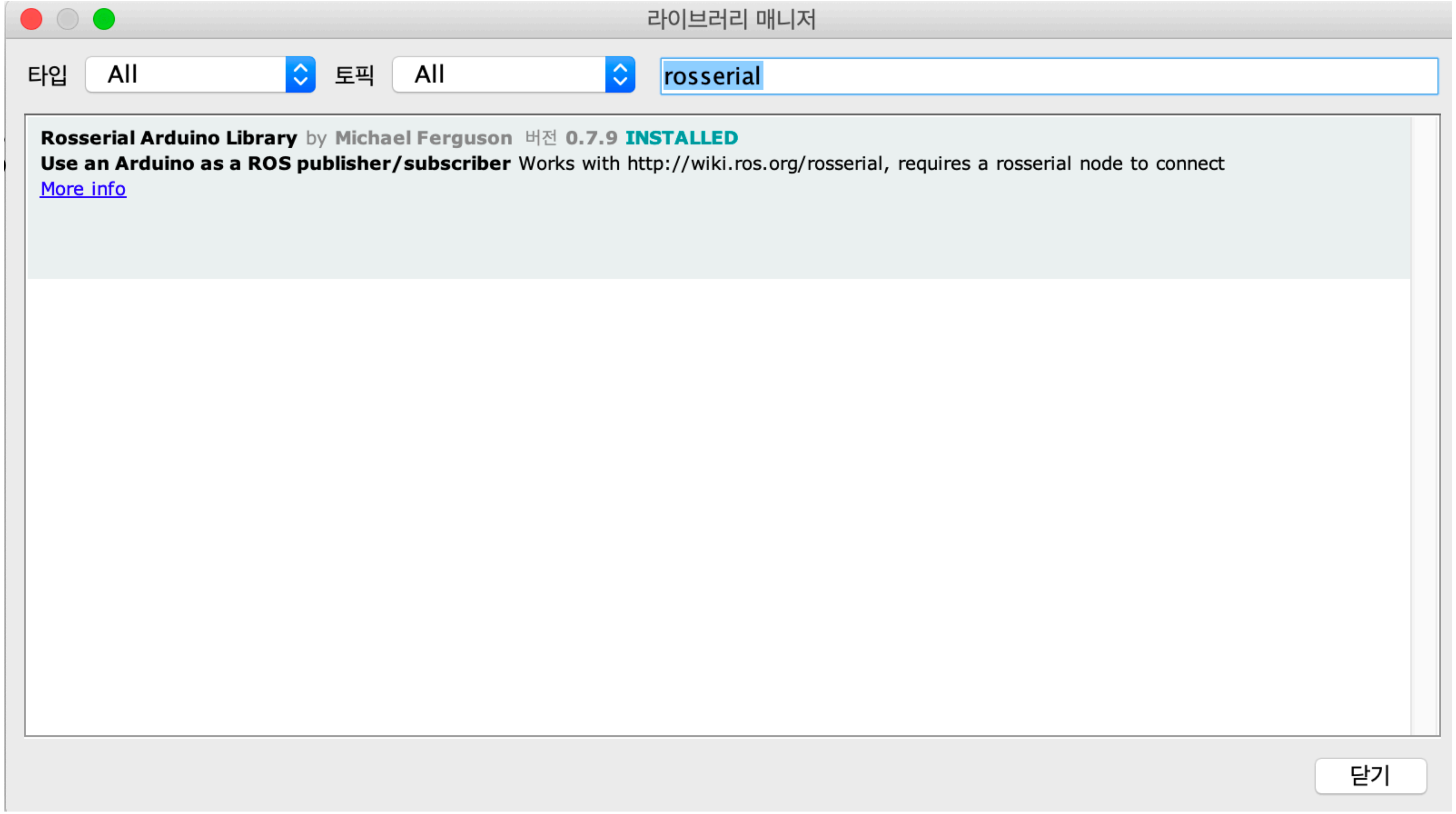
오로카

2019.07.12
최규남

MAKER SPACE Pangyo


Ubuntu 16.04 Kinetic
Master_node
Publisher_node
USB
rosserial
Geomerty_msgs/Twist
Subscriber_node
ROS
PWM
Motor1
H-Bridge L298
Motor2

1st Byte	Sync Flag (Value: 0xff)
2nd Byte	Sync Flag / Protocol version
3rd Byte	Message Length (N) - Low Byte
4th Byte	Message Length (N) - High Byte
5th Byte	Checksum over message length
6th Byte	Topic ID - Low Byte
7th Byte	Topic ID - High Byte
N Byte	Serialized Message Data
Byte N+8	Checksum over Topic ID and Message Data



Rosserial Arduino Library by Michael Ferguson 버전 0.7.9 **INSTALLED**
Use an Arduino as a ROS publisher/subscriber Works with http://wiki.ros.org/rosserial, requires a rosserial node to connect
More info

표 1-1

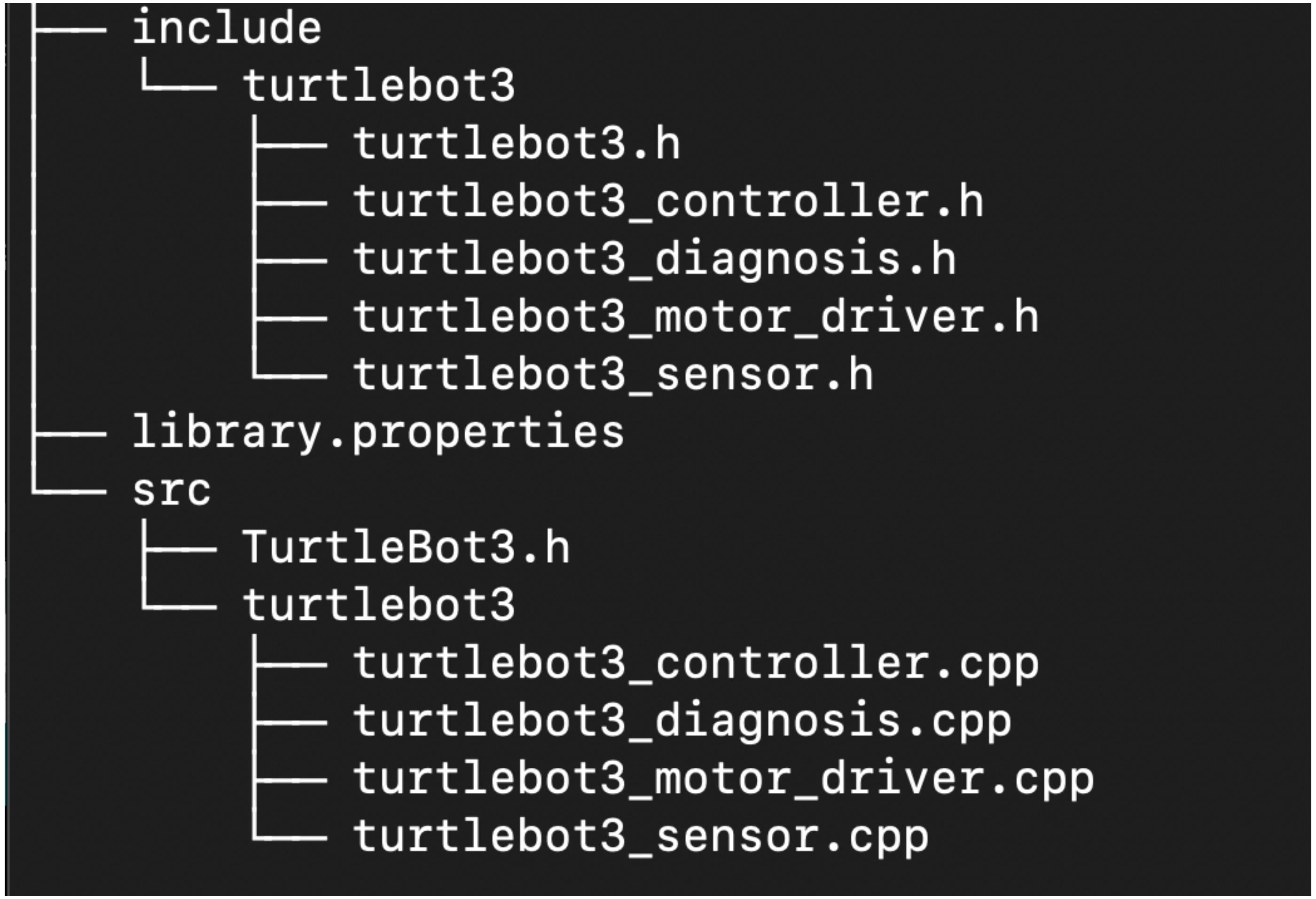
#define	CONTROL_MOTOR_SPEED_FREQUENCY	30	//hz
#define	CONTROL_MOTOR_TIMEOUT	500	//ms
#define	IMU_PUBLISH_FREQUENCY	200	//hz
#define	CMD_VEL_PUBLISH_FREQUENCY	30	//hz
#define	DRIVE_INFORMATION_PUBLISH_FREQUENCY	30	//hz
#define	VERSION_INFORMATION_PUBLISH_FREQUENCY	1	//hz
#define	DEBUG_LOG_FREQUENCY	10	//hz

표 1

nh.subscribe(cmd_vel_sub)	• The desired linear and angular x,y,z of the base
nh.subscribe(sound_sub)	Sound
nh.subscribe(motor_power_sub)	Motor torque
nh.subscribe(reset_sub)	Reset gyro, Reset odm

표 1

nh.advertise(sensor_state_pub)	
nh.advertise(version_info_pub)	Version Info
nh.advertise(imu_pub)	IMU data
nh.advertise(cmd_vel_rc100_pub)	Bluetooth Remote Controller
nh.advertise(odom_pub)	Odm data
nh.advertise(joint_states_pub)	
nh.advertise(battery_state_pub)	battery state
nh.advertise(mag_pub)	Mag sensor data



```
├── include
│   └── turtlebot3
│       ├── turtlebot3.h
│       ├── turtlebot3_controller.h
│       ├── turtlebot3_diagnosis.h
│       ├── turtlebot3_motor_driver.h
│       └── turtlebot3_sensor.h
├── library.properties
└── src
    ├── TurtleBot3.h
    └── turtlebot3
        ├── turtlebot3_controller.cpp
        ├── turtlebot3_diagnosis.cpp
        ├── turtlebot3_motor_driver.cpp
        └── turtlebot3_sensor.cpp
```