
Robot2Robot SPL

ROS Sports

Aug 09, 2022

CONTENTS

1	Installation	3
1.1	Binary Installation	3
1.2	Source Installation	3
2	Tutorial	5
2.1	Sending from Robot 1	5
2.2	Receiving on Robot 2	6
3	Versions	9
3.1	SPL Standard Message (SPLSM)	9
3.2	Supported Distros	9

Robot2Robot SPL, abbreviated as R2R_SPL is a ROS2 package that handles Intra-Team Communication for the RoboCup Standard Platform League.

In RoboCup Standard Platform League, communication between teammates have strict rules. They must adhere to a specific data structure and be broadcasted on the field network via UDP on a port designated to the team.

This package provides a ROS2 topic interface for team communication, such that teams can simply publish to `r2r/send` to broadcast a message, and subscribe to `r2r/recv` to listen to messages from teammates. The UDP connection and conversion to and from raw bytes are handled in this package.

The project is hosted on [Github](#) by ROS Sports. **Issues and Pull Requests are welcome!**

INSTALLATION

Warning: This package targets **ROS2 Foxy onwards**. It won't compile on all ROS1 and older ROS2 distros.

To install the packages, do one of the following:

- a *Binary Installation*
- a *Source Installation*

1.1 Binary Installation

Binary installation is available for all active distros.

Source your ROS installation, then run:

```
sudo apt update
sudo apt install ros-${ROS_DISTRO}-r2r-spl-7
```

The most recent version of the SPL Standard Message is version 7, as defined in the SPL rulebook. If you intend on using an older SPL Standard Message format, replace 7 with the version you want to use. Currently the only available version is 7. In the future, packages will be added to all available distros to cover future competitions.

If this method does not work for your platform, perform the *Source Installation* instead.

1.2 Source Installation

1.2.1 Cloning repositories

Source your ROS2 installation, then in your ROS2 workspace, run:

```
git clone https://github.com/ros-sports/r2r_spl.git src/r2r_spl --branch ${ROS_DISTRO}
```

1.2.2 Building

In your ROS2 workspace, install the dependencies:

```
rosdep install --from-paths src -i
```

Build the package:

```
colcon build
```


TUTORIAL

In this tutorial, you'll learn to make two robots on the same network communicate with each other. Different `ROS_DOMAIN_ID` will be used to isolate communication between ROS nodes running on the two robots on the same network.

Note: This tutorial does not require two robots, you can run it on your PC.

2.1 Sending from Robot 1

First, we'll look at sending a message from the first robot to other robots on the network.

Open a new terminal window. This window will imitate robot 1.

Start the `r2r_spl` node, specifying the team number (0) and player number (1) of robot 1. Select the tab from below corresponding to the SPL Standard Message version you are using:

7

master

```
# Set ROS_DOMAIN_ID to 1
export ROS_DOMAIN_ID=1

# Run the node
ros2 run r2r_spl_7 r2r_spl --ros-args -p team_num:=0 -p player_num:=1
```

```
# Set ROS_DOMAIN_ID to 1
export ROS_DOMAIN_ID=1

# Run the node
ros2 run r2r_spl_master r2r_spl --ros-args -p team_num:=0 -p player_num:=1
```

Open a new tab and publish an example SPLSM message on the `r2r/send` topic, using the command below:

7

master

```
# Set ROS_DOMAIN_ID to 1
export ROS_DOMAIN_ID=1
```

(continues on next page)

(continued from previous page)

```
# Publish the message
ros2 topic pub r2r/send splsm_7/msg/SPLSM "
player_num: 1
team_num: 0
fallen: 0
pose:
- 1000.0
- 2000.0
- 1.5
ball_age: 10.0
ball:
- 200.0
- 0.0
data:
- 0
- 100
- 255
"
```

```
# Set ROS_DOMAIN_ID to 1
export ROS_DOMAIN_ID=1

# Publish the message
ros2 topic pub r2r/send splsm_master/msg/SPLSM "
player_num: 1
team_num: 0
fallen: 0
pose:
- 1000.0
- 2000.0
- 1.5
ball_age: 10.0
ball:
- 200.0
- 0.0
data:
- 0
- 100
- 255
"
```

2.2 Receiving on Robot 2

Open a second terminal window. This window will imitate the robot 2.

Start the `r2r_spl` node, specifying the team number (0) and player number (2) of robot 2.

```
7
```

```
master
```

```
# Set ROS_DOMAIN_ID to 2
export ROS_DOMAIN_ID=2

# Run the node
ros2 run r2r_spl_7 r2r_spl --ros-args -p team_num:=0 -p player_num:=2
```

```
# Set ROS_DOMAIN_ID to 2
export ROS_DOMAIN_ID=2

# Run the node
ros2 run r2r_spl_master r2r_spl --ros-args -p team_num:=0 -p player_num:=2
```

Open a new tab, and echo messages on the r2r/recv topic, using the command below:

```
# Set ROS_DOMAIN_ID to 2
export ROS_DOMAIN_ID=2

# Echo the messages
ros2 topic echo r2r/recv
```

You should see a message being received at 1Hz from the first robot, as following:

```
---
player_num: 1
team_num: 0
fallen: 0
pose:
- 1000.0
- 2000.0
- 1.5
ball_age: 10.0
ball:
- 200.0
- 0.0
data:
- 0
- 100
- 255
---
```

This indicates that the first robot is sending messages to the other robot, not using ROS topics, but over UDP packets as specified in the SPL rules!

VERSIONS

The RoboCup competition is hosted annually, with the competition usually around July. Additionally, there are regional RoboCup competitions such as the German Open, Asia Pacific, and Japan Open. The structure of the packets used to communicate between robots are restricted in SPL and frequently change with rule changes accompanied by a version number increment.

3.1 SPL Standard Message (SPLSM)

The structure of SPL Standard Message is defined in [GameController](#), and has an associated version number. Every time there is a rule change for the message, the version number is incremented and a new package named `r2r_spl_<VERSION_NUMBER>` (eg. `r2r_spl_7`) will be released for all active ROS2 distros.

3.2 Supported Distros

Packages will be released for all active ROS distributions at the time of competition. The table below lists the current availability of packages under different ROS distros and competitions.

Table 1: Package availability

	Foxy	Galactic	Humble	Rolling
<code>r2r_spl_7</code>	Yes	Yes	Yes	Yes
<code>r2r_spl_master</code>	No	No	No	No