

# ETSA02

# **Robot Communication Protocol**

# 2018

v1.0

Developed by Markus Borg

Approved by the Regulatory Body

## Revision history

Name	Date	Description	Version
Markus Borg	2018-03-23	Initial version of the communication protocol.	1.0

# Fundamentals of Robocode team communication

Robots that extend TeamRobot can communicate with team members. If you want to implement this feature, please study the Robocode API:

<http://robocode.sourceforge.net/docs/robocode/robocode/TeamRobot.html>

## **Broadcasting message to all team members:**

```
public void broadcastMessage(Serializable message)
```

## **Sending messages to one specific team member:**

```
TeamRobot.sendMessage(String name, Serializable message)
```

## **When a robot receives a message, the following method is called:**

```
onMessageReceived(MessageEvent)
```

The MessageEvent always contains the name of the sending robot and a specific message.

# Evolution of ETSA02 Robot Communication Protocol

To allow communication between robots developed by different suppliers, the ETSA02 Regulatory Body has standardized a communication protocol. All messages are to be transmitted as plain text strings, with one command per line. The ETSA02 Robot Communication Protocol will be version controlled, and groups that decide to implement communication should carefully specify in their requirements specifications which version of the protocol is implemented.

The ETSA02 Robot Communication Protocol will evolve during the course, i.e., the Regulatory Body will release new versions. All groups are welcome to provide feedback and requests to the standardization work done by the Regulatory Body. To do so, the project manager should submit a "Change Request" to the Regulatory Body. The Change Request should carefully describe what changes are proposed, and motivate why this request should be considered.

# ETSA02 Robot Communication Protocol

A message string should have the following overall structure, including the new lines.

```
[0-1] leadership;[followMe|leadMe]
[0-1] teamMode;[offensive|defensive]
[0-1] myPos;x;y
[0-*] enemyPos;x;y
[0-1] targetEnemy;name
[0-1] targetPos;x;y
[0-1] moveTo;x;y
```

Example message:

```
teamMode;offensive
myPos;100;100
enemyPos;50;50
enemyPos;200;150
targetPos;50;50
```

The commands shall have the following meaning:

leadership

Tells the receiving robot(s) that the sender takes on leadership (`followMe`) or reports as a follower (`leadMe`)

teamMode

Tells the receiving robot(s) that the team shall enter offensive mode (`offensive`) or defensive mode (`defensive`)

myPos

Tells the receiving robot(s) its current position on the 2D battlefield

enemyPos

Tells the receiving robot(s) the position of an identified enemy robot

targetPos

Tells the receiving robot(s) to fire at a specific position on the 2D battlefield

targetEnemy

Tells the receiving robot(s) to target a specific enemy robot

moveTo

Tells the receiving robot(s) to move to a specific position on the 2D battlefield