

EDACup

2025

22.08 Algoritmos y Estructuras de Datos

EDABot Datasheet rev. 5



Resumen

EDABot es un robot móvil diseñado para competiciones de deportes robóticos. Cuenta con cuatro ruedas omnidireccionales (que proveen tres grados de libertad, dos de traslación y uno de rotación), un dribbler para controlar la pelota, un kicker para dispararla hacia adelante y un chipper que la impulsa hacia arriba a un ángulo de 45°.

Especificaciones técnicas

Dimensiones	↑ 140 mm, ø 180 mm
Peso	2.6 kg
Centro de masa	35 mm (respecto del piso)

Movimiento

El movimiento del robot se controla mediante un controlador PID que ajusta continuamente su posición y orientación para alcanzar los valores objetivo establecidos.

Dribbler

El dribbler consiste en un cilindro accionado por un motor que permite controlar la pelota frente al robot, atrayéndola mediante su rotación. La velocidad del dribbler es regulable de forma continua, desde 0 (apagado) hasta 1 (máxima velocidad).

Kicker y chipper

El kicker y el chipper funcionan mediante solenoides activados por pulsos eléctricos. La potencia de disparo se ajusta de forma continua con un valor entre 0 (apagado) y 1 (máxima potencia).

Entorno

Cada equipo dispone de una mesa con cuatro robots: dos del equipo blanco y dos del equipo negro.

- **Acceso a la simulación:** Disponible en <https://www.openreality.app/>.
- **Control de los robots:** Se realiza a través de la aplicación edacup. Para configurar los robots, es necesario modificar el archivo `edacup.json` con la siguiente estructura:

```
{
  "accessToken": "Your access token",
  "team1Binary": "Executable file for white team",
  "team2Binary": "Executable file for black team"
}
```
- **accessToken:** Código que permite a cada equipo acceder a sus robots.
- **team1Binary y team2Binary:** Archivos ejecutables que controlan los robots del equipo blanco y negro, respectivamente. Si no se desea controlar un equipo, el campo puede dejarse en blanco ("").

La comunicación entre edacup y los ejecutables se realiza mediante mensajes JSON transmitidos por `STDIN` y `STDOUT`, con cada mensaje en una línea de texto terminada por `\n`.

Mensajes enviados por edacup

- **Mensaje "start":** Indica que el juego se reanuda.

```
{"type": "start"}
```

- **Mensaje "stop":** Indica que el juego se detiene. Inicialmente, el partido está detenido.

```
{"type": "stop"}
```

- **Mensaje "state":** Enviado cada 50 ms para reportar el estado de los robots y la pelota.

```
{
  "type": "state",
  "data": {
    "homeBot1": {
      "position": [x, y, z],
      "rotation": [x, y, z],
      "velocity": [x, y, z],
      "angularVelocity": [x, y, z]
    },
    "homeBot2": {
      "position": [x, y, z],
      "rotation": [x, y, z],
      "velocity": [x, y, z],
      "angularVelocity": [x, y, z]
    },
    "rivalBot1": {
      "position": [x, y, z],
      "rotation": [x, y, z],
      "velocity": [x, y, z],
      "angularVelocity": [x, y, z]
    },
    "rivalBot2": {
      "position": [x, y, z],
      "rotation": [x, y, z],
      "velocity": [x, y, z],
      "angularVelocity": [x, y, z]
    },
    "ball": {
      "position": [x, y, z],
      "rotation": [x, y, z],
      "velocity": [x, y, z],
      "angularVelocity": [x, y, z]
    }
  }
}
```

El centro del sistema coordenado se ubica en el centro de la mesa. La coordenada x se mide a lo largo del eje largo de la mesa, de izquierda a derecha. La coordenada z se mide a lo largo del eje corto de la mesa, del fondo al frente. Las rotaciones se expresan en radianes, utilizando ángulos eulerianos. Las coordenadas del equipo "home" corresponden siempre al lado izquierdo.

Respuestas del ejecutable teamBinary

El ejecutable puede responder a cada mensaje "state" con un mensaje "set", que controla los robots del equipo (homeBot1 y homeBot2):

```
{
  "type": "set",
  "data": {
    "homeBot1": {
      "positionXZ": [x, z],
      "rotationY": y,
      "dribbler": 0-1,
      "kick": 0-1,
      "chirp": 0-1
    },
    "homeBot2": {
      "positionXZ": [x, z],
      "rotationY": y,
      "dribbler": 0-1,
      "kick": 0-1,
      "chirp": 0-1
    }
  }
}
```

Sólo es necesario enviar un mensaje "set" en respuesta a un mensaje "state". Cualquier otra respuesta será ignorada.

Los valores de "dribbler", "kick" y "chirp" deben estar entre 0 (apagado) y 1 (máxima potencia).