# **Speed Dreams Roborace API**

## Introduction

Speed Dreams is an open source simulator developed by a community of enthusiasts. It’s being maintained and regularly updated. The “vanilla” version, community board and other useful information can be found on the following website: [http://www.speed-dreams.org/.](http://www.speed-dreams.org/)

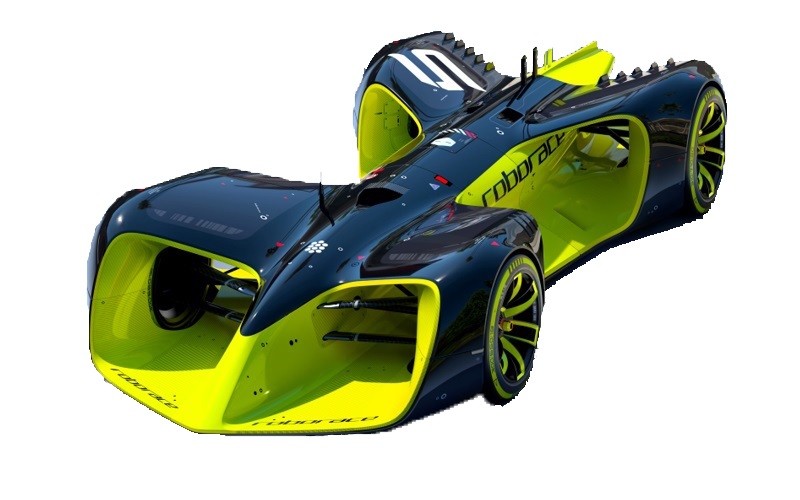
The simulator was adopted by Roborace to create a standalone solution for early control systems testing. Roborace is not planning to utilize Speed Dreams as a final platform for virtual testing, qualifying and other official events - rather it’s a quick temporary solution that allows teams to start testing their control systems immediately.

Repository with additional modules developed by Roborace is located here: <https://github.com/vadbut/Roborace-Dreams>

## Vehicle Model

We aimed to define a physical vehicle model that accurately represents performance of the actual vehicle. However, the Speed Dreams game engine is not well suited for modeling electrical vehicles with aerodynamic configuration of Robocar. Thus, one should be aware that the performance of the vehicle in the simulator environment might significantly differ from the behavior of the real vehicle. Robocar visual and physical model parameters are defined in the following folder: ~\data\cars\models\Robocar.

Motion of the vehicle is measured with respect to an absolute coordinate system (CS) attached to the ground: X – East, Y – North, Z – Up. In addition to the absolute CS, a local (vehicle) CS is defined. Vehicle CS axes are depicted in the image below. The local CS origin is at the vehicle center of gravity (CG).



**Z**

**Y**

**X**

**CG**

## UDP Interface

Speed Dreams was augmented with an interface that allows controlling robot cars. The interface is implemented using UDP protocol. Each Roborace robot driver has an assigned incoming (to the sim) and an outgoing (from the sim) ports:

|  |  |  |
| --- | --- | --- |
| Robot name | In port | Out port |
| udp 1 | 3001 | 4001 |
| udp 2 | 3002 | 4002 |
| … | .. | .. |
| udp 20 | 3020 | 4020 |

Currently, UDP interface is configured for broadcasting for the both in and out ports. The protocol configuration can be customized in the following file: ~\src\drivers\udp\src\driver.cpp.

The protocol for the simulator out stream is defined below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Signal | Start Byte | Type | Units | Description |
| LapTime | 0 | float | seconds | current lap time |
| DistStart | 4 | float | meters | distance from start line |
| LastLapTime | 8 | float | seconds | last lap time |
| RacePos | 12 | int32 | - | race position |
| RPM | 16 | float | rev/min | RPM |
| EnergyLevel | 20 | Float | TBD | battery level |
| GearPos | 24 | int32 | - | gear position |
| WheelVelFR | 28 | float | rad/sec | wheel speed front right |
| WheelVelFL | 32 | float | rad/sec | wheel speed front left |
| WheelVelRR | 36 | float | rad/sec | wheel speed rear right |
| WheelVelRL | 40 | float | rad/sec | wheel speed rear left |
| AbsPosX | 44 | float | meters | position of vehicle CG in abs CS |
| AbsPosY | 48 | float | meters |
| AbsPosZ | 52 | float | meters |
| AngX | 56 | float | rad | roll (X), pitch (Y) and yaw (Z) angles of the vehicle |
| AngY | 60 | float | rad |
| AngZ | 64 | float | rad |
| AngRateX | 68 | float | rad/s | angle rates |
| AngRateY | 72 | float | rad/s |
| AngRateZ | 76 | float | rad/s |
| AbsVelX | 80 | float | m/s | absolute speeds along the axis of the global CS |
| AbsVelY | 84 | float | m/s |
| AbsVelZ | 88 | float | m/s |
| AccelX | 92 | float | m/s2 | vehicle acceleration in the local CS |
| AccelY | 96 | float | m/s2 |
| AccelZ | 100 | float | m/s2 |
| SteerAngle | 104 | float | deg | current road wheel angle |
| WheelReactionFL | 108 | float | N | front left wheel reaction |
| WheelReactionFR | 112 | float | N | front right wheel reaction |
| WheelReactionRR | 116 | float | N | rear right wheel reaction |
| WheelReactionRL | 120 | float | N | rear left wheel reaction |
| Collision | 124 | int32 | - | collision state (0-no collision) |
| V2xN | 128 | uint8 | - | number of V2x objects |
| V2xType | 129 | uint8 [39] | - | type of the objects:  0 – unknown, 1 – ai racing car  2 – human driver, 3 - slow car, 10 - static object |
| V2xXPos | 168 | float [39] | meters | position of the objects in the absolute CS |
| V2xYPos | 324 | float [39] | meters |
| V2xSpeed | 480 | float [39] | m/s | speed of the object |
| V2xYaw | 636 | float [39] | rad | yaw angle of the object |
| SimStatus | 792 | uint8 | - | 3 – active, 5 - stopped |
| SimCounter | 793 | uint8 | - | increments every send cycle |
| Total size: 794 bytes | | | | |

The protocol for the simulator in stream is defined below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Signal | Start Byte | Type | Units | Description |
| SteerAng | 0 | float | deg | “+” – left, “-“ - right |
| AccelRequest | 4 | float | N∙m or % gas | Overall torque request or gas pedal percentage (depending on CtrlMode) |
| BrakeRequest | 8 | float | % brake | brake pedal request (0 to 1) |
| GearPos | 12 | int32 | - | gear position: -1 – reverse,  0 – neutral, 1 - drive |
| CtrlMode | 16 | uint8 | - | control mode:  3 – torque request, N∙m  11 –gas pedal request, % |
| Ctr | 20 | uint8 | - | the simulator checks if controller is alive using this counter |

## Additional useful Information

• The following file contains definitions of the simulation and robot time steps: ~\src\interfaces\Raceman.h

#define RCM\_MAX\_DT\_SIMU 0.002 // <- timestep of the simulation

#define RCM\_MAX\_DT\_ROBOTS 0.010 // <- timestep of robot implementation

• One can perform a race between udp robots and a human driver at the same time. Currently, the driver is configured to send out a data stream at port 5001 (broadcasting). The human driver data steam interface can be reconfigured in the following file:

~\src\drivers\human\human.cpp

The table below defines a list of signals send out from the human driver.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Signal | Start Byte | Type | Units | Description |
| V2xType | 0 | float | - | equals 2 (human driver) |
| PosX | 4 | float | meters | global position of the human car CG |
| PosY | 8 | float | meters |
| Speed | 12 | float | m/s | absolute speed of the car |
| YawAngle | 16 | float | rad | yaw angle of the car |