Software Survey 2026

Team name

RoboEireann

Which division(s) are you applying for? If your used software differs between divisions, please fill out the survey once per division.

Middle Size (height < 125 cm, weight < 25 kg)

Is your software fully or partially OpenSource? If so, where can it be found?

Open source. See https://github.com/roboeireann/RoboEireannCodeRelease

Are you using any software developed by other teams? If so, list every component that you are reusing and the team that originally developed it.

Yes. Our currently released code was based on the B-Human CodeRelease 2021 and used all components from it except the following which our team developed: Coroutine Behaviour Engine, Event based Team Communications, Robot and Jersey detection, Visual referee gesture detector, Whistle detector, and text logging. Selected RL Behaviours based on an approach by WisTex United were also included. We are planning to use the B-Human CodeRelease 2025 with similar components replaced for 2026.

Are you using any datasets in your research? If you are using your own datasets, are they public?

Yes we are using datasets, both our own and 3rd party. Our robot detection dataset is public, see Yao, Z., Douglas, W., O'Keeffe, S., Villing, R. (2022). Faster YOLO-LITE: Faster Object Detection on Robot and Edge Devices. In: Alami, R., Biswas, J., Cakmak, M., Obst, O. (eds) RoboCup 2021: Robot World Cup XXIV. RoboCup 2021. Lecture Notes in Computer Science(), vol 13132. Springer, Cham. https://doi.org/10.1007/978-3-030-98682-7_19

Please list the scientific publications your team has made since the last application to RoboCup (or if not applicable in the last 2 years).

No scientific publications from the team in the last 2 years.

Are there any other contributions you would like to share with the RoboCup community?

Other than open source code (detailed above), open datasets (also linked from spl.robocup.org), and publications in the RoboCup symposium there are no other recent contributions for the team.

Which approach are you using to generate the robot walking motion?

Previously we used the walking engine in the B-Human 2021 code release. For the HSL we are developing an RL based walk for the Booster K1 for 2026.

Which approach are you using to generate other motions of the robot (e.g. kicking, standing up)?

For kicking we are also developing RL motions for 2026. We have not yet decided whether to use the Booster stand up motion or not.

Do you have a kinematic or dynamic model of your robot? If so, how did you create it (e.g. measure physical robot, export from CAD model)?

From the manufacturer and underlying B-Human code release.

What approaches are you using in your robot's visual perception?

Modified YOLO-based object detection for robot detection (team mates and opponents), MobileNet-based classifier for visual referee detection.

Are you planning with objects in Cartesian or image space? If you are using Cartesian space, how do you transform between the image space and cartesian space?

We plan in cartesian space and transform using the camera matrix which maintains a representation of the camera pose in space.

Do you have some form of active vision (i.e. moving the robots camera based on information known about the world)?

Only in a limited sense, e.g. keeping the ball in view and keeping nearby obstacles (robots) in view

What approach are you using to localize your robot?

Multi-hypothesis Unscented Kalman Filter (implemented in the underlying B-Human framework)

Is your team performing team communication? Which communication protocol are you using?

Custom packet format over UDP, conforming with 2025 SPL rules to limit the amount of data per packet and total number of packets per game.

What approach are you using for navigation? Are you avoiding obstacles?

Yes obstacles are detected and avoided using one of two walk planners (implemented by the underlying B-Human code release). At distance, the planner treats obstacles as cylinders and plans piecewise linear trajectories that bypass the obstacles.

How is the behavior of your robots structured? (e.g. Behaviour Trees)

We use a custom behaviour architecture based on stackless coroutines (C++) and our behaviour is implemented as a hierarchy of such coroutines which is ticked once per cognition cycle.

Are you simulating your robot? If so, which simulator are you using and for what purpose do you use simulations?

Yes we use simulation using the SimRobot simulator provided as part of the B-Human code release.

What operating system is running on your robot and which middleware are you using (for example Ubuntu 22.04 and ROS2 Galactic)?

For 2025 we used Ubuntu 20.04. For 2026 we will use Ubuntu 22.04 and 24.04. We do not use a middleware such as ROS currently.

Is there anything else you would like to share that did not fit any previous question?

Nothing for now.