Mighty Thymio for University-level Educational Robotics*

Jérôme Guzzi[1], Alessandro Giusti[1], Gianni A. Di Caro[2] and Luca Maria Gambardella[1]
[1]Dalle Molle Institute for Artificial Intelligence (IDSIA), USI-SUPSI, Lugano, Switzerland
[2]Carnegie Mellon University in Qatar, Doha, Qatar

Abstract

Thymio is a small, inexpensive, mass-produced mobile robot with widespread use in primary and secondary education. In order to make it more versatile and effectively use it in later educational stages, including university levels, we have expanded Thymio's capabilities by adding off-the-shelf hardware and open software components. The resulting robot, that we call *Mighty Thymio*, provides additional sensing functionalities, increased computing power, networking, and full ROS integration. We present the architecture of Mighty Thymio and show its application in advanced educational activities.

Introduction

Thymio (Mondada et al. 2017) is a small (11 x 11.2 x 5.3 cm) commercial open-hardware mobile robot designed for teaching computational thinking (Magnenat et al. 2014) and mobile robotics. Thymio is part of a very successful package[1] that comprises the robot, different visual programming environments, and activities for schools at primary and early secondary stages of education.

Thymio is an affordable and sturdy platform made by: a translucent plastic body with attachments for Lego bricks, two differentially driven wheels, an internal battery, 9 proximity sensors (infrared), 5 buttons, an accelerometer, a microphone, a loudspeaker, an IR receiver, three large RGB LEDs to change the body color, and several smaller LEDs spread around the body. The open source firmware can be programmed using Aseba (Magnenat et al. 2011), a distributed, event-based programming language. A wireless (802.15.4) interface has been recently introduced.

Thymio has been adopted by schools, in particular in Switzerland and in France. However, it lacks some features that are important for educational activities at university levels, such as the option to add sensors (e.g., visual ones), integration with ROS (Quigley et al. 2009), and onboard computing power for complex perceptual and control tasks.

To overcome these limitations, we introduce *Mighty Thymio*, an incremental extension of Thymio to a Linux-based, ROS-compatible platform suitable for higher education. The resulting robot remains very affordable and compact, suitable to be carried at home by students. Alternative commercial solutions with a comparable software flexibility, such as the Turtlebot 3,[2] are significantly larger, heavier, and more expensive. We describe the upgrade in hardware and software components of the Mighty Thymio, and present its use in two educational activities.[3]

Platform Description

The hardware upgrade is tool-free and can be performed by the students themselves in a few minutes. A frame with Lego bricks is added directly on the Lego-compatible Thymio base. The frame carries the following hardware (Fig. 1-left):

Odroid C1: a powerful quad-core ARM Linux board with 1 GB SDRAM, 16 GB flash memory, 4 USB ports;

Power pack: an easily swappable USB power pack that powers the robot for several hours;

Odroid UPS: a Li-Po battery that provides power when swapping power packs;

Odroid 720p (web) camera: fixed by a tiltable clap;

Wi-Fi dongle: that provides wireless networking.

Parts are connected by common USB cables. In particular, the Thymio base is connected to one of the Odroid C1's USB ports to send commands and read sensors. The Odroid C1 exposes many other interfaces (Ethernet, UART, I^{2}C, GPIO, SPI, ADC) to connect additional sensors and actuators. The total cost of the add-ons is about 130 dollars.

Software We provide a file-system image that includes all software for the Odroid C1. The software is based on ROS. We extend the open-source Aseba-ROS bridge and Thymio ROS driver[4] to exposes all Thymio's sensors and actuators using standard ROS messages. Moreover, we add ROS interfaces for: camera, LED dynamic patterns (blinking/waving patterns that are very helpful to represent robot internal states), battery monitoring (the robot changes color

*This project has been partially supported by the Swiss National Center of Competence Research (NCCR) Robotics through the Swiss National Science Foundation.

[1] `https://www.thymio.org`

[2] `https://turtlebot3.readthedocs.io`

[3] Parts list, detailed building instructions, and code available at `https://github.com/jeguzzi/mighty-thymio`.

[4] `https://github.com/ethz-asl/ros-aseba`

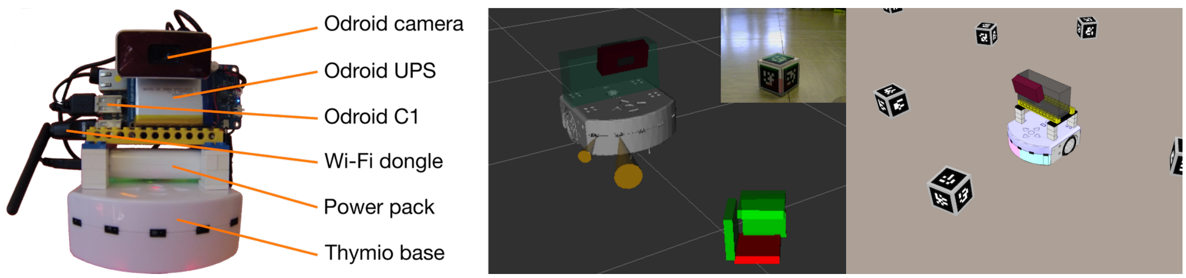

Figure 1: Left: The Mighty Thymio. Center: Robot visualization in rviz during a real-world experiment: proximity sensors (brown cones), camera (top-right image) and detected AR marker (red/green marker). Right: VREP simulation.

when a battery swap is needed), and wireless configuration. Robot's buttons are used to change Wi-Fi from hot-spot to managed mode (to connect to a router) and to safely power-off the onboard computer.

We provide client ROS libraries that: (a) expose the robot's configuration as URDF to visualize the state of all sensors in rviz; (b) interface with the VREP Simulator (Rohmer, Singh, and Freese 2013) by emulating the ROS-Aseba bridge.

Usage We briefly describe the typical interaction of a student with a single robot.[5] The robot acts as the ROS master. The student connects to the robot's Wi-Fi hotspot and runs ROS nodes, on his machine, to control and inspect the robot (Fig. 1-center). When the student's code is complete, it can be ported without effort to the robot's onboard computer. The same ROS driver can control both the real robot and the simulated robot, which allows the student to prototype algorithms in simulation (Fig. 1-right).

Educational Activities

We present two activities about robot perception, that exploit the camera's video stream and the onboard processing of the Mighty Thymio. Both activities were successfully completed by students.

As an **assignment for a graduate course in mobile robotics**, students are provided with a robot and several paper cubes with AR markers on the faces (Fig. 1-right). The student uses a ROS library[6] to detect, in a camera frame, the id and world pose of visible cubes. Then, they explore landmark-based SLAM techniques to incrementally build a map of all cubes, while piloting the robot from the keyboard; once the map is built, they use RRT to plan a path to a given point. The assignment is first solved in simulation, then tested with the real robots, for which landmark detection is affected by errors. The educational content includes direct and inverse kinematics, feedback-based control, vision-based SLAM, path planning.

As a **Bachelor project on machine learning** (Toniolo et al. 2018), the student's goal is to train a deep convolutional network to detect obstacles in front of the robot from camera images. In a first step, the student implements a controller that randomly explores an environment and logs camera frames along with the corresponding outputs from the 5 front proximity sensors. Resulting data from multiple environments is collected in a large training dataset and used to train a CNN that predicts proximity sensor outputs given a camera image. Finally, the student implements a controller that avoids obstacles only relying on camera images, and demonstrates it by masking the proximity sensors with tape.

Conclusions

We presented hardware and software extensions that enable the use of the Thymio robot in higher education. Software is based on ROS. Hardware is composed of affordable and easy to assemble parts. We are currently using the robot in Switzerland for promotional and educational activities, from secondary educational stages to robotics graduate courses.

References

Magnenat, S.; Rétornaz, P.; Bonani, M.; Longchamp, V.; and Mondada, F. 2011. Aseba: A modular architecture for event-based control of complex robots. *IEEE/ASME Transactions on Mechatronics* 16(2):321–329.

Magnenat, S.; Shin, J.; Riedo, F.; Siegwart, R.; and Ben-Ari, M. 2014. Teaching a core cs concept through robotics. In *Proc. of the ACM Conference on Innovation and Technology in Computer Science Education*, 315–320. ACM.

Mondada, F.; Bonani, M.; Riedo, F.; Briod, M.; Pereyre, L.; Retornaz, P.; and Magnenat, S. 2017. Bringing robotics to formal education: The thymio open-source hardware robot. *IEEE Robotics Automation Magazine* 24(1):77–85.

Quigley, M.; Conley, K.; P Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; and Y Ng, A. 2009. Ros: an open-source robot operating system. In *Proc. of Int. Conf. on Robotics and Automation (ICRA), Workshop on Open Source Software*.

Rohmer, E.; Singh, S. P. N.; and Freese, M. 2013. V-REP: a versatile and scalable robot simulation framework. In *Proc. Int. Conf. on Intel. Robots and Systems (IROS)*, 1321–1326.

Toniolo, S.; Guzzi, J.; Giusti, A.; and Gambardella, L. M. 2018. Learning an image-based obstacle detector with automatic acquisition of training data. In *Proc. of the Thirty-Second AAAI Conf. on Artificial Intelligence (AAAI-18 Demo Track)*, to appear.

[5] control of multiple robots is possible with a wireless router

[6] `http://wiki.ros.org/ar_track_alvar`