

IOT Meets Robotics - First Steps, RIOT Car, and Perspectives

Dinesh Reddy, Lokesh Reddy, Dr K K Sharma
Department Of Electronics & Communication Engineering
Lingaya's University

Abstract-We present a cloud-enhanced, four-wheeled, mobile minirobot, assembled from low-cost, off-the-shelf hardware parts, and open-source software building blocks. These building blocks provide a reusable & extensible base for emerging applications mixing robotics with the Internet of Things.

Index Terms- IoT, Robotics, Cloud Computing

I. Introduction

The recent years have seen wide interest and innovation in the field of Internet of Things (IoT), triggered by technological advances in embedded systems hardware, software, and connectivity. The increasing availability of tiny, cheap, power-efficient micro-controllers and peripherals has spun a new category of computers: low-end IoT devices. Even though such devices cannot run traditional operating systems (e.g. Linux and equivalents) due to very constrained memory, CPU, power resources, most low-end IoT devices have enough resources to run newer operating systems and cross-platform application code. Furthermore, recent network technology and protocol standardization efforts have enabled new interconnection capabilities for such devices, such as low-power, end-to-end IPv6 based networking.

Simultaneously, robotics is experiencing a dramatic growth, not only in their traditional applications, such as industrial automation, but also in other domains such as self-driving cars, and personal robots such as drones, vacuum cleaning robots, and other types in the making. While ever smaller robots are targeted by the field of nanorobotics, another class of robots (and applications) is expected to consist of mini-robots approximately of the size and computing capabilities of current IoT devices. Mini-robots are expected to become commodity and 1000 times cheaper than currently available robots (see for instance the AFRON Challenge). Leveraging a number of emerging techniques, such as 3D printed robots (see for instance poppy), and network connectivity enabling new paradigms ranging from fog computing to cloud robotics, such robots are likely to be massively deployed in a variety of application domains

in the near future. The encounter of IoT and robotics thus promises to open a fascinating new field.

II. IoT meets Robotics

An emerging class of mini-robots will inherit from the same constraints as current IoT devices (e.g. actuators) including very limited memory, finite processing power, and strong energy limitations. In the following, we focus on three important aspects in IoT robotics: hardware aspects, software aspects, and network aspects. **Hardware Aspects:** From a hardware perspective, a robot consists in (i) structural and mechanical components, e.g. carcass, frame, wheels, (ii) sensor and actuators, e.g. motors, distance sensors, (iii) computational elements and electronics, e.g. micro-controllers, motor controllers, and (iv) power supply, e.g. batteries. Recently, the rise of open source hardware and the maker scene lead to increased availability and such a significant price drop for these components that mini-robots under INR 500 are becoming a reality. Popular examples of structural components include *Lego*, while 3D printers allow virtually anyone to conveniently create custom parts with a high precision. The *Arduino community* lead to the availability of a wide range of affordable sensors (from inertial measurement units to full-blown laser distance scanners), and a variety of actuators became available due to the *scale modelling community*. The market around low-power, low-cost micro-controllers is currently booming. Basing robots on these low-power platforms conveniently allows the use of standard off-the-shelf batteries, or, in intermittent activity scenarios, of small solar panels and other means of energy harvesting solutions. **Software Perspective:** The software running on IoT mini-robots consists in (i) hardware abstraction and device drivers, (ii) control software, (iii) communication software, and (iv) a systems layer that glues together all these elements. The most popular software base for robots is the Robot Operating System (*ROS* [9]) a set of libraries and tools running on top of a host operating system (i.e. a traditional OS such as Linux, Windows). *ROS* is thus not intended to run on mini-robot hardware, whose constrained resources (memory, CPU, power) won't match traditional OS resource requirements. Instead, newer and more compact operating systems

[4] must be used as base on such hardware. For instance, *RIOT* provides real-time capabilities, hardware abstraction, multi-threading, and full IPv6 networking while fitting the tight memory constraints of micro-controllers typically found on low-end IoT devices. However, contrary to *ROS*, *RIOT* does not provide specific libraries targeting robotics. Nevertheless, this shortcoming could be overcome by porting light-weight robotics libraries to *RIOT*. This task is simplified by *RIOT* providing common developer APIs, such as BSD sockets or POSIX thread (*pthread*).

Network Challenges: On the network side, IoTminirobots need (i) enhanced algorithms and protocols, and (ii) novel/holistic network architectures. Aforementioned constraints on software and hardware translate into challenges for network technologies, which are expected to operate with low memory footprint, low energy consumption, and high reliability over wireless, and to interoperate with the Internet. For instance, the IETF is currently standardizing the use of IPv6 (with protocols as 6LoWPAN, RPL, CoAP) over low-power wireless link layers in IoT, e.g. BLE, or IEEE 802.15.4 using TDMA and frequency hopping to increase reliability. But these are not designed to accommodate mobility, temporal loss of connectivity and topological changes, in addition to the classical radio interference, multipath fading: they should be extended and adapted (see [11] for instance). Furthermore, IoT robotics combines embedded system constraints with the extreme complexity of some tasks IoT robot may have to carry out (e.g. grasping an unknown object/environment); thus, it will be necessary to deport some of the logic and/or processing for robot control to remote server(s) i.e., the cloud. Elements of such an architecture already exist (protocols such as, publish/subscribe in *ROS*, or *roscerial*). But convergence/adaptation is needed between such elements and standard IoT protocols in the making, such as CBOR, COAP, MQTT, or ICN. The goal being to provide a fully integrated communication architecture, from IoT minirobots up to the cloud.

III. Demo

We will present a four-wheeled, mobile minirobot (see Fig. 1) we have built assembling low-cost, off-the-shelf components including a low-power MCU (ARM Cortex-M0+), DC drive motor, power stage, steering server, and ultrasonic distance sensor. The behavior of the minirobot will be (i) reprogrammable on the fly from the cloud, (ii) simultaneously subject to local and cloud-based control loops. For local control the minirobot will run *RIOT*, an open source real-time operating system which fits resource constrained and low-cost micro-controller platforms. For communication with the cloud, the minirobot will combine low-power wireless (IEEE 802.15.4) and IP protocols, providing end-to-end connectivity via standard access points, using *RIOT*'s default network stack. The cloud

component consists in a simple REST-based daemon using CoAP to communicate with the robot, and the web for user interaction (e.g. changing the robot's behavior).

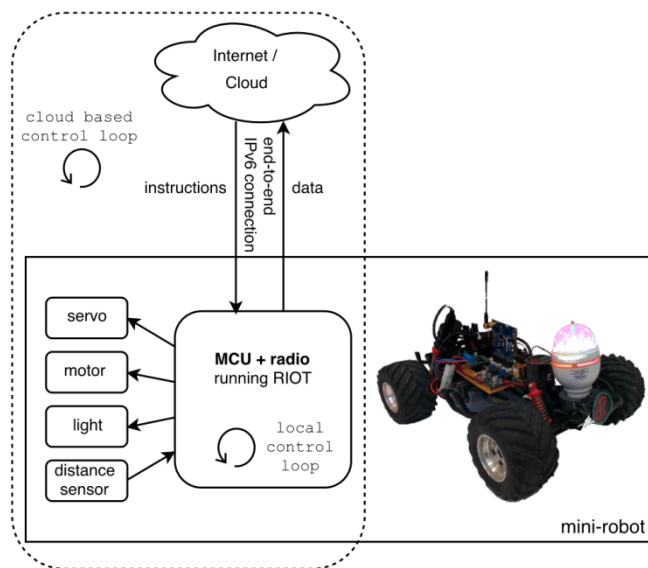


Figure 1. High-level architecture and image of low-cost 4-wheeled minirobot, cloud-controlled, using low-power wireless, IPv6 and *RIOT*.

IV. Conclusion and Future Work

The demo we present decomposes into cleanly separated building blocks, using open-source software and off-the-shelf hardware. In particular, care was taken to make it straightforward to (i) substitute the local real-time control loop on the minirobot with more advanced motor control, local sensor data fusion and short-term decision making, (ii) relocate and/or enhance the cloud-based control loop with more advanced computational offloading for sensor data processing, remote decision making, and mid- to long-term planning, or (iii) add/substitute sensors and actuators on the minirobot. Thus, this work can be easily reused and extended for a wide range of emerging applications mixing IoT and robotics. Our future work will focus on efficient portable software running on IoTminirobots, computation offloading schemes, and optimizing standard IoT protocols for reliability in face of mobility and multihop over low-power wireless.

ACKNOWLEDGEMENTS

We would like to express our special thanks of gratitude to **DR K. K. SHARMA** (Senior Professor) who was always willing to help and give his best suggestions. Special thanks to the reviewers for pointing out ways to improve the presentation of this paper.

References

- [1] Aversive C++ Library. <http://aversiveplusplus.com>.
- [2] E. Baccelli et al. *RIOT OS: Towards an OS for the Internet of Things*. In *IEEE INFOCOM*, 2013.

- [3] F. Bonomi et al. Fog Computing and its Role in the Internet of Things. In *ACM Mobile Cloud Computing Workshop*, 2012.
- [4] O. Hahm et al. Operating Systems for Low-End Devices in the Internet of Things: a Survey. *IEEE Internet of Things Journal*, 2016.
- [5] A. S. Huang et al. LCM: Lightweight Communications and Marshalling. In *IEEE IROS*, 2010.
- [6] B. Kehoe et al. A Survey of Research on Cloud Robotics and Automation. *IEEE Trans on Automation Science and Engineering*, 2015.
- [7] G. A. Korash et al. African Robotics Network and the 10 Dollar Robot Design Challenge. *IEEE Mag. on Robotics & Automation*, 2013.
- [8] M. Lapeyre et al. Poppy: Open Source 3D-printed Robot for Experiments in Developmental Robotics. In *IEEE ICDL*, 2014.
- [9] M. Quigley et al. ROS: an Open-Source Robot Operating System. In *ICRA Workshop on Open-Source Software*, 2009.
- [10] M. Rubenstein et al. Programmable Self-assembly in a ThousandRobot Swarm. *Science*, 2014.

IJSER