ROBOTICS AND THE INTERNET OF THINGS: FIRST STEPS, THE RIOT CAR, AND FUTURE PERSPECTIVES

Kailash Mohapatra[1], Satyaban subudhi[2]

[1]Professor, Department of Mechanical Engineering, Raajdhani Engineering College, Bhubaneswar, Odisha
[2]Assistant Professor, Department of Mechanical Engineering, Raajdhani Engineering College, Bhubaneswar, Odisha

ABSTRACT

We demonstrate a four-wheeled, mobile mini-robot that is boosted by the cloud and is built using open-source software and inexpensive, commercial hardware components. These building elements give new applications that combine robotics and the Internet of Things a reusable and extendable foundation.

Keywords: *IoT, Robotics, Cloud Computing*

1. INTRODUCTION

Due to technological advancements in embedded system hardware, software, and communication, the Internet of Things (IoT) has experienced a surge in interest and innovation in recent years. Low-end Internet of Things (IoT) devices are a new class of computers that have emerged as a result of the growing availability of small, affordable, power-efficient microcontrollers and peripherals. The majority of low-end IoT devices have enough resources to run more modern operating systems [4] and cross-platform application code, despite the fact that such devices cannot run conventional operating systems (such as Linux and equivalents) due to severely constrained memory, CPU, and power resources. Additionally, recent advances in network technology and protocol standardisation have made it possible for these devices to communicate in novel ways, such as through low-power, end-to-end IPv6 based networking. Simultaneously, robotics is experiencing a dramatic growth, not only in their traditional applications, such as in- dustrial 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, an- other class of robots (and applications) is expected to consist of mini-robots [10] approximatively of the size and comput- ing capabilities of current IoT devices. Mini-robots are ex- pected to become commodity and

1000 times cheaper than currently available robots (see for instance the AFRON Chal- lenge [7]). Leveraging a number of emerging techniques, such as 3D printed robots (see for instance Poppy [8]), and network connectivity enabling new paradigms ranging from fog computing [3] to cloud robotics [6], such robots are likely to be massively deployed in a variety of application do- mains in the near future. The encounter of IoT and robotics thus promises to open a fascinating new field.

2. IoT meets Robotics

A new class of mini-robots will inherit the same limits as present Internet of Things (IoT) devices (such as actuators), such as extremely constrained memory, finite computing capacity, and severe energy limitations. The following focuses on three crucial facets of IoT robotics: network, software, and hardware factors.

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 avail- ability and such a significant price drop for these compo- nents, that mini-robots under $10 are becoming a reality [7]. Popular examples of structural components include *Lego*, while 3D printers allow virtually anyone to conveniently cre- ate custom parts with a high precision. The scale modelling community helped make a variety of actuators accessible, while the Arduino community helped make a broad range of reasonably priced sensors (from inertial measurement units to full-fledged laser distance scanners) available. Currently, the market for low-cost, low-power micro-controllers is booming. Basing robots on these low-power platforms makes it simple to employ common off-the-shelf batteries or, in situations where there is intermittent activity, modest solar panels and other kinds of energy gathering.

Software Perspective: Control software, communication software, hardware abstraction and device drivers, and a systems layer that connects all of these components make up the software that runs on IoT mini-robots. The Robot Operating System is the most widely used software platform for robots (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

ISSN: 2278-4632
Vol-11 Issue-01 March 2021

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* [2] provides real-time capabilities, hardware abstraction, multi-threading, and full IPv6 networking while fitting the tight memory constraints of micro-controllers typ- ically 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 [1] to *RIOT*. This task is simplified by RIOT providing common developer APIs, such as BSD sockets or POSIX thread (*pthread*).

Network Challenges: IoT mini-robots require both innovative/holistic network topologies and enhanced network-side algorithms and protocols. Network technologies have challenges as a result of the aforementioned demands on hardware and software. These technologies must operate with a small memory footprint, little power consumption, high wireless dependability, and internet compatibility. 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 in- crease reliability. But these are not designed to accommo- date mobility, temporal loss of connectivity and topological changes, in addition to the classical radio interference, mul- tipath fading: they should be extended and adapted (see [11] for instance). Furthermore, IoT robotics combines embed- ded system constraints with the extreme complexity of some tasks IoT robot may have to carry out (e.g. grasping an un- known object/environment); thus, it will be necessary to de- port some of the logic and/or processing for robot control to remote server(s) i.e., the cloud. Elements of such an architec- ture already exist (protocols such as [5], publish/subscribe in *ROS*, or rosserial). 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 architec-ture, from IoT mini-robots up to the cloud.

3.DEMONSTRATION

We will present a four-wheeled, mobile mini-robot (see Fig. 1) we have built assembling low-cost, off-the-shelf com- ponents including a low-power MCU (ARM Cortex-M0+), DC drive motor, power stage, steering server, and ultrasonic distance sensor. The behavior of the mini-robot will be (i) reprogrammable on the fly from the cloud, (ii) simultane- ously subject to local and cloud-based control loops. For local control the mini-robot will run RIOT, an open source real-time

ISSN: 2278-4632
Vol-11 Issue-01 March 2021

operating system which fits resource constrained and low-cost micro-controller platforms. The mini-robot will use RIOT's default network stack to provide end-to-end connection while combining low-power wireless (IEEE 802.15.4) and IP protocols for communication with the cloud. A straightforward REST-based daemon that uses CoAP to connect with the robot makes up the cloud component. The web is used for user interaction (such changing the robot's behaviour).

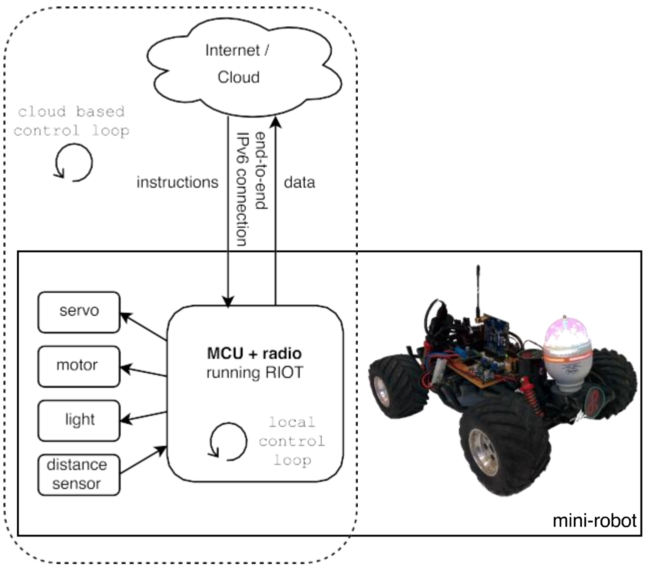

Figure 1 shows the high-level architecture and picture of a low-cost, 4-wheeled minirobot that is cloud-controlled Using low-power wireless, IPv6, and RIOT.

4. CONCLUSION AND FUTURE WORK

Using open-source software and readily available hardware, the example we present breaks down into neatly divided building components. It was especially important to make it simple to: I replace the mini-local robot's real-time control loop with more sophisticated motor control, local sensor data fusion, and short-term decision making; (ii) relocate and/or enhance the cloud-based control loop with more sophisticated computational offloading for sensor data processing; and (iii) add/substitute sensors and actuators on the mini-robot. As a result, our work is easily reusable and extendable for a variety of new IoT and robotics applications. Our future work will concentrate on compute offloading techniques, efficient portable software operating on Internet of Things minirobots, and standard IoT protocol optimization for reliability in the face of mobility and multihop over low-power wireless.

ISSN: 2278-4632
Vol-11 Issue-01 March 2021

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 Mar- shalling. In IEEE IROS, 2010.

[6] B. Kehoe et al. A Survey of Research on Cloud Robotics and Automa- tion. 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 Experi- ments 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 Thousand- Robot Swarm. Science, 2014.

[11] A. Tinka et al. A decentralized scheduling algorithm for time synchro- nized channel hopping. In Ad Hoc Networks, Springer. 2010.