Profile-driven Power Optimizations for AAL Robots

Maximizing Robots Idle Time by Offloading Monitoring Workload to Dedicated Hardware Components

Georgios Keramidas, Nikolaos Voros, Christos Antonopoulos
Embedded System Design and Application Laboratory, Technological Educational Institute of Western Greece, Antirio, Greece
{gkeramidas;voros;cantonopoulos}@teiwest.gr

Fynn Schwiegelshohn, Philipp Wehner, Diana Göhringer
Ruhr-University, Bochum, Germany,
{fynn.schwiegelshohn;philipp.wehner;diana.goehringer}@rub.de

Evaggelinos Mariatos
AVN Technologies, Cyprus
emariatos@avntechgroup.com

***Abstract*—The EU funded project RADIO brings forward a new health care paradigm according to which a mobile robot platform can act as assistant to an elderly person in his/her domestic environment. Under this context, unobtrusiveness is of paramount importance since the robot should be a natural participant of patients' daily life. However, robot assistance in everyday living still suffers from limited autonomy dictaded by the robot battery. This paper presents the approach taken in the project in order to reduce, to the extent possible, the usage of the power-hungry processing components. This minimizes the need for revisiting the robot charging station during the day, by using hardware accelerated ultra-low-power monitoring during the periods of inactivity.**

Keywords—Robot; Low Power; Monitoring; Hardware Acceleration; Ambient Assisted Living

I. INTRODUCTION

The RADIO consortium [1] develops an assisted living architecture around a low cost house robot. The robot acts mainly as a mobile sensor platform that monitors and guides an elderly or disable person throughout the entire day, using cameras and microphones as well as through direct access to the home automation infrastructure. The core concept and approach of the project has been presented in more details in [2].

However, to achieve its mandate, the robot has to constantly know where the person is and to be able to move if the person also moves to a new location or another room. While monitoring, the robot mechanical systems do not operate, thus saving battery life. But this is not enough; the power consumed by the main robot controller is significantly high, given that (even when not moving) a relatively powerful computing engine needs to be active. This holds true even for idle time e.g., when the person is sleeping or watching TV, as there is no way for the robot to know if and when the person intends to move. To reduce this idle-time power consumption, as part of the RADIO project, a HW FPGA-based implementation of event detection algorithms is set forward.

By carefully profiling power consumption in typical everyday activities, we are able to split the detection algorithms in a HW-only part that consumes a fraction of the normal idle power, an embedded SW part that runs on an ARM processor (co-located with the the FPGA fabric in a SoC module) and a high-complexity part which is involved only when really needed and executed on the main computer of the robot. We present our profile-driven approach to achieve low-power consumption at system-level, thus increase the battery life of the robot.

The rest of the paper is organized as follows. Section 2 describes main processing units and presents different modes of operation. Section 3 analyzes which processing unit is suitable for each mode of operation. Section 4 outlines our profiling-based approach for reduced system power consumption. Section 5 gives preliminary results and Section 6 concludes the work and presents future development plans.

II. THE ROBOT AND THE PROCESSING UNITS

The RADIO robot is built on the TurtleBot2 platform [3]. It is outfitted with two processing units: an Intel NUC which is responsible for controlling sensors and actuators and an Avnet PicoZed equipped with Xilinx Zynq-7000 all programmable System on Chip (APSoC) [4]. Additional devices are an Asus Xtion Pro camera and a Hokuyo laser scanner [3].

In general, there are two types of data processed in the system:

- High throughput streaming data that comes from continually receiving the output of a microphone (audio stream) or a camera (video stream)

2159-3477/17 $31.00 © 2017 IEEE
DOI 10.1109/ISVLSI.2017.72

- Event or control-like data with relatively small size, collected by sensors. Event/measurement data can also be the outcome of streaming data analysis, e.g., processing of video can lead to the generation of an “exit” event if the camera looks towards the door.

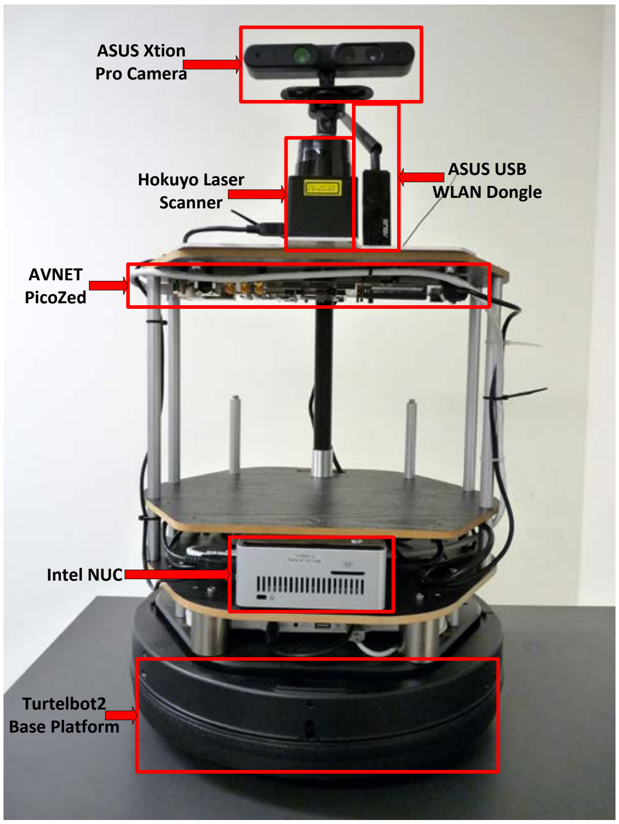

Figure 1. RADIO robot platform

External interfacing is achieved through RF networks with emphasis on low power and minimal usage of the RF spectrum. These networks are not suitable for conveying real time streaming data like audio or video, leaving no other option but to perform the processing of the multimedia workloads in the robot.

The assisted living approach heavily relies on the collection and processing of audio and video streams for analyzing and recognizing activities of daily life (ADL) [5] [6] [7]. Recognizing the emotional status of patients and the identification of emergency situations, such as the detection of falls, are also of high importance [8] [9] [10]. As a result, the main processing engine(s) of robot are designed to operate as a power-efficient architecture for streaming data processing.

Apart from the FPGA fabric, the Xilinx PicoZed platform includes also an ARM Cortex A9 dual core processor (equipped with a Neon co-processor) interfaced with programmable logic.

The processing of the data streams needs to be:

- *Responsive:* in some cases the data collected on the robot and the measured/detected quantities and events are correlated centrally with data from the smart home sensors or other sources. For example, detection of exiting the room can be correlated with a motion sensor mounted on the room walls.
- *Efficient,* in terms of energy consumption: as the robot relies on battery, the processing of data should be limited to what is necessary and executed on this node which provides the lowest overall power consumption

Figure 2 illustrates the streaming data flow through the robot processing elements and their interfaces. The camera and audio data streams are continuously monitored and when activity is detected the corresponding algorithms (which can analyse and recognise the activity) are triggered.

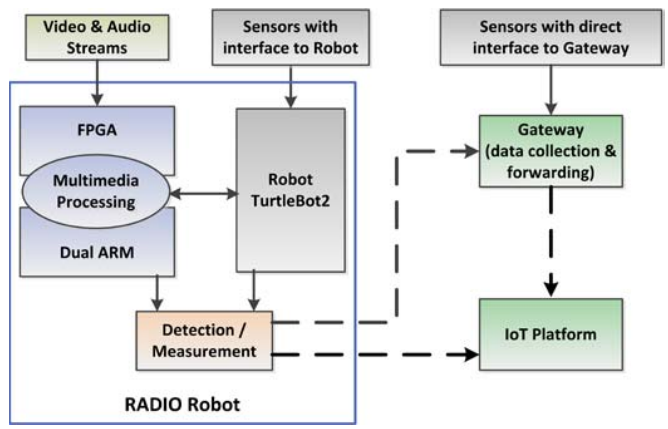

Figure 2. The main data flow scenario

Depending on the specific combination of algorithms that get triggered, some or all computational tasks may be executed in the i) NUC, ii) Zynq ARM processor, or iii) accelerated by hardware components in the FPGA. To put this into perspective, an example state-diagram of the robot is presented in Figure 3.

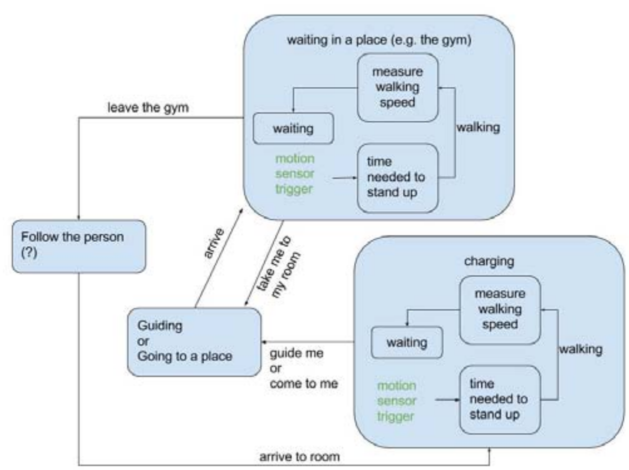

Figure 3. State diagram highlighting different modes

In this diagram, there are a number of states that involve heavy data processing and we have to ensure that this is either very low power or executed when the robot is stationed on its charging station. This is further analyzed in the next section.

III. PROCESSING SUBSYSTEM USAGE AT EACH STATE

The RADIO robot is a unit which has many energy-hungry subsystems. These are:

- Main processor to control robot movement (NUC)
- FPGA to accelerate ADL recognition methods
- Sensors, especially the image sensor (camera)
- Mechanical subsystem (motors)
- Wireless subsystem (network)

If all subsystems are always active, the RADIO robot will need to be recharged every few hours, which results in long periods of robot non-availability. As a first step, we had to understand how each subsystem is used and if it indeed needs to be active at each use case. Table I provides an overview, assuming that robot activity can be classified in the following states:

- *Waiting:* at this state, the robot is not moving; neither is it processing sensor data. It will be triggered by some external event
- *Moving:* when leading the way or following a person
- *Monitoring:* at this state, the robot is not moving but it is processing sensor input data in order to detect an ADL or understand patient's mood

For some of these states, there is a difference on whether the robot is on its charging station or away of it e.g., in another room.

TABLE I. ROBOT SUBSYSTEMS ENERGY USAGE

State	CPU	FPGA	Sensors	Motors	Network
Waiting	Used	Not used	Not used	Not used	Used
Moving	Used	Used	Used	Used	Used
Monitoring/ Away	Used	Used	Used	Not used	Used
Monitoring/ Charging	Used	Used	Used	Not used	Used

The energy consumed at each state by each subsystem is not the same. For example, the CPU while waiting can be clocked at lower frequency, drastically reducing the required power. Also, sensors and FPGA can perform only basic data capture and processing when monitoring away from the charging dock and revert to full-power processing mode when this power is available.

Although a number of such techniques are used, their impact on power consumption is not drastic. To cope with this problem, our view is to develop dedicated hardware components that allow the robot to turn-off complete subsystems in some cases; turning them on only on demand and just for the short period when they are really needed. The goal is to have an improved energy profile. The results of this analysis are depicted in Table II.

A HW accelerator component is a specially designed circuit which is implemented in FPGA and is connected directly to the other subsystems. The component is processing signals from sensors, so that simple decisions on whether other subsystems have to be employed or not can be devised. Typically, this component is equipped with the following functionality:

- Triggering mechanism, which initiates sensor data capture and processing
- Local Memory, which holds processed sensor data so that the main system RAM does not have to be used
- Signal processing acceleration functions in FPGA
- Control interfaces to turn-on and notify (or get notified by) other subsystems

TABLE II. ENERGY PROFILE BY USING HW ACCELERATORS

State	CPU	FPGA	Sensors	Motors	Network
Waiting	On demand	Used	Not used	Not used	On demand
Moving	Used	Used	Used	Used	Used
Monitoring/ Away	On demand	Used	Used	Not used	On demand
Monitoring/ Charging	Used	Used	Used	Not used	Used

IV. EXPERIMENTAL PROFILING

To prototype and experiment with the alternative approach discussed in this paper, we selected a small number of ADLs as target use cases for the monitoring state of the robot. The selected ADLs are the ones which detect:

- The time needed by the patient to get out of bed: This ADL is based on image processing algorithms that observe the patient while getting out of bed. The image processing algorithms can be parallelized availing themselves from the acceleration within the FPGA hardware. This algorithm divides the image into different regions. If the centre of mass of moving pixels over succeeding images lies in one of these defined regions, an event is triggered. Thus, this algorithm is able to detect if a person is sleeping, awake (but not going out of bed), and awake and standing up
- Picking up medication cups: The image processing methods used to detect this ADL benefit from the acceleration through the FPGA hardware as they rely on a computational intensive algorithm

The acceleration does not involve the complete method; but rather focuses on early detection of a high-possibility for an event so that SW-based processing can be invoked. Specifically, in the context of the above mentioned ADLs:

- For the time-to-stand-up ADL, the hardware component will collect and calculate data from all regions, providing a trigger to software components when a given activity threshold is crossed
- For the cup-detection ADL, since this is manually triggered by the operator, hardware acceleration is not related to the recognition but to the stabilization and centering of the image. It has been observed through field trials that the robot can slightly move while waiting and this movement can issue false

positives. An always running HW component will be monitoring such small movements and constantly re-centre the view

In order to determine the SW-HW co-design of the FPGA-ARM system, extensive profiling of the image processing algorithms is needed. We profiled three flow options, so that the expected benefits of various optimization approaches can be quantified, allowing focusing on these solutions that are more beneficial (in terms of power consumption) in each case.

The three analyzed options are:

- No offloading i.e., all processing is performed on the robot's main processing unit (NUC)
- Offload on embedded ARM core of the FPGA (no HW acceleration)
- Offload on dedicated low-level hardware blocks in the FPGA (ARM core can be power down)

To make a realistic profiling, we identified typical activity use cases with the help of non-technical partners of the RADIO consortium. Each typical activity is depicted as a combination of five states for the robot subsystem:

- *Moving*, where the robot is actually moving and uses its motor, sensors and camera
- *Monitoring*, where the robot is waiting for an event to be triggered by what it can see
- *Sensing*, where the robot is using its onboard sensors or communicates with smart home
- *Processing*, where heavy processing to analyze sensor and camera input is required
- *Idle*, where the robot is on but is doing nothing of the above

It is important to understand that a specific human activity (e.g., having lunch) will combine more than one of the above states (e.g., looking, sensing, and processing).

By accumulating the energy needs at each activity, we take a daily activity profile in terms of energy consumption as shown in Figure 4.

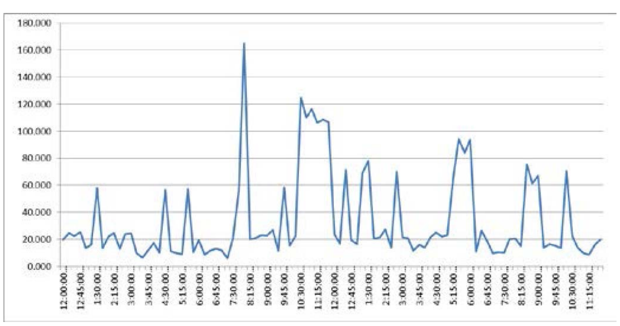

Figure 4. Dialy activity profile

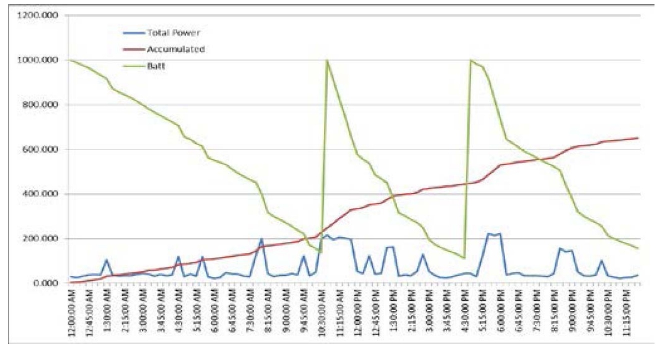

Figure 5. No offloading, all processing is performed on main unit (NUC)

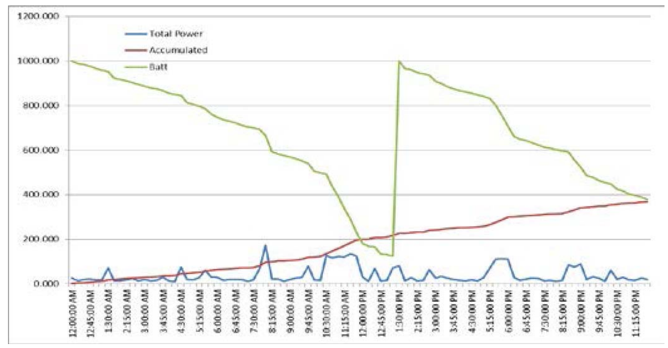

Figure 6. Offload on embedded ARM core (no HW acceleration)

V. RESULTS

The three options listed in Section IV are then tested on these profiles. The main reason for doing this activity is to identify if there is potential for maximizing battery life, i.e., reducing the number of required re-charges during one day – thus we added a battery load calculation in our results as shown in Figures 5, 6, and 7 (indicated by the green line).

The obtained results are very promising showing a major improvement in power consumption (saving 1 or 2 recharges during the day) by offloading parts of the NUC processing on the APSoC. A smaller but still important gain is further achieved when we move the wake-up decision in the FPGA fabric of the APSoC, allowing also the ARM core to sleep during periods of inactivity.

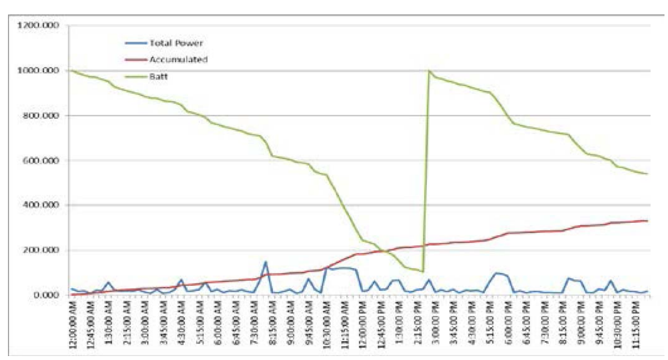

Figure 7. Offload on dedicated low-level hardware blocks in the FPGA.

VI. Conclusions and Further Work

This paper presented a profile-driven, system-level approach to increase the autonomy of the robot battery in AAL environments. The presented approach is benchmarked in experimental conditions via use case profiling. The next step is to apply this method on the actual field trials, thus validating the results in unforeseen scenarios and confirming that the reduced recharging need improves usability and autonomy of the full system.

Acknowledgment

This study is part of the collaborative project RADIO which is funded by the European Commission under Horizon 2020 Research and Innovation Programme with Grant Agreement Number 643892.

References

[1] EU project RADIO. Robots in Assisted Living Environments: Unobtrusive, Efficient, Reliable and Modular Solution for independent Ageing. http://www.radio-project.eu

[2] C. P. Antonopoulos, G. Keramidas et al. Robots in Assisted Living Environments as an Unobtrusive, Efficient, Reliable and Modular Solution for Independent Ageing: The RADIO Perspective. Proc. of Intl. Sympocium in Applied Reconfigurable Computing, 2015.

[3] TurtleBot2 Robot Development Kit. http://www.turtlebot.com

[4] Xilinx Inc. Zynq-7000 All Programmable SoC Overview, 2015.

[5] D. Sgouropoulos, T. Giannakopoulos et al. Clothes change detection using the Kinect sensor. Proc. of Intl. Conference on Signal Processing and Multimedia Applications, 2014.

[6] C. Zhang, Y. Tian. RGB-D Camera-based Daily Living Activity Recognition. Journal of Computer Vision and Image Processing, 2012.

[7] H.M. Hondori, M. Khademi et al. Monitoring intake gestures using sensor fusion (microsoft kinect and inertial sensors) for smart home telerehab setting. Proc. of Annual Healthcare Innovation Conference, 2012.

[8] M. Vacher, A. Fleury et al. Complete sound and speech recognition system for health smart homes: application to the recognition of activities of daily living. New Developments in Biomedical Engineering, 2010.

[9] S. Koolagudi, K.S. Rao. Emotion recognition from speech: a review. Journal of Speech Technology, 2012.

[10] Z. Zeng, M. Pantic et al. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. Transactions on Pattern Analysis and Machine Intelligence, 2009.