ROBOTS IN ASSISTED
LIVING ENVIRONMENTS

UNOBTRUSIVE, EFFICIENT, RELIABLE AND MODULAR SOLUTIONS FOR INDEPENDENT AGEING

Research Innovation Action

Project Number: 643892 Start Date of Project: 01/04/2015 Duration: 36 months

DELIVERABLE 4.8

Integrated smart home with robotic platform extensions I

Dissemination Level	**Public**
Due Date of Deliverable	Project Month 18, 30 September 2016
Actual Submission Date	12 April 2017
Work Package	WP4: *Physical home architecture development and integration of cost-effective, reliable and power-efficient RADIO components for elder monitoring and caring*
Task	T4.4: *Smart home design and integration*
Lead Beneficiary	S&C
Contributing beneficiaries	NCSR-D, TWG, ROBOTNIK, S&C
Type	DEM
Status	Submitted
Version	Final

Project funded by the European Union's Horizon 2020 Research and Innovation Actions

Abstract

This deliverable is the RADIO Home prototype, a set of hardware devices and the corresponding software. The prototype demonstrates a complete RADIO Home, but without its interconnections with other RADIO Homes or other elements of the RADIO Ecosystem.

History and Contributors

Ver	Date	Description	Contributors
00	6 Jul 2016	Document structure	S&C and NCSR-D
02	15 Mar 2017	RADIO Home deployment and demonstration	NCSR-D, TWG, ROBOTNIK, S&C
03	5 Apr 2017	RADIO Home demonstration video	NCSR-D, S&C, TWG
04	11 Apr 2017	Cover document preparation	NCSR-D, TWG, S&C
05	12 Apr 2017	Internal review comments and corrections	RUB
Fin	12 Apr 2017	Addressing, peer review comments, final preparations and submission.	NCSR-D

Abbreviations and Acronyms

NCSR-D	National Centre for Scientific Research “Demokritos”
TWG	Technical Educational Institute of Western Greece
RUB	Ruhr Universitaet Bochum
ROBOTNIK	Robotnik Automation SLL
S&C	Sensing & Control Systems S.L.
AVN	AVN Innovative Technology Solutions Ltd.
FSL	Fondazione Santa Lucia
FHAG	Fundació Hospital Asil de Granollers
FZ	Frontida Zois
ADL	Activities of Daily Life
BLE	Bluetooth Low Energy
CPS	Cyber-Physical Systems
ICT	Information and Communications Technology
ROS	Robot Operating System

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

1.1 Purpose and Scope

This deliverable demonstrates the first RADIO Home prototype, integrating smart home sensing and automation with the RADIO Robot. Within the scope of this deliverable is to demonstrate the hardware prototypes and to publish the source code of the software developed for the Main Controller, the network gateways, and other software needed for integration purposes.

1.2 Approach

This deliverable is prepared within *Task 4.4: Smart home design and integration*. This task uses the architecture and components developed within the previous tasks in WP4 and complements them with commercially available smart home and home automation devices in order to design and integrate the specific smart home environment that will be deployed for the RADIO pilots. This task also implements the Main Controller that orchestrates the overall RADIO Home system and the gateways needed to make existing solutions interoperable. Work in Task 4.4 is in tandem with work in Task *4.1 Designing device interconnection and interfacing*, to update the physical architecture, a living document that documents the integrated system as it evolves and has a snapshot (D4.2) delivered based on D4.8.

1.3 Relation to other Work Packages and Deliverables

This deliverable is prepared following the *Architecture for extending smart homes with robotic platforms I* (D4.1) and integrates components developed within WP4 (D4.4 and D4.6). This integrated prototype is documented by the second version of the architecture, D4.2.

This deliverable will be the starting point for the final prototype, D4.9, due M30.

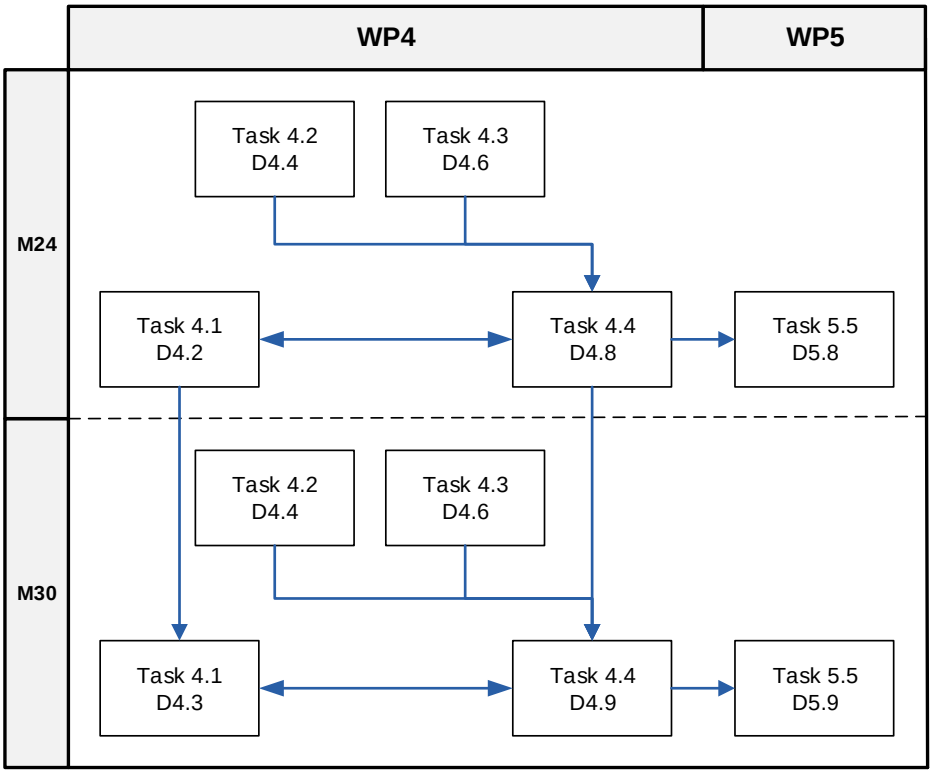

Figure 1: Relation to other Work Packages and Deliverables

2 PROTOTYPE HARDWARE

The first RADIO Home prototype integrates the hardware components listed below.

Integration base	Component
ROS/Wifi	First RADIO Robot prototype [D4.6].
BLE	BLE Beacons and tags for localization.
ZWave	Multisensor 6 for general comfort sensors (e.g., temperature).
ZWave	Presence sensor and pressure sensor for detecting presence and using furniture (chairs, bed).
ZWave	Magnetic sensors for detecting door opening.
ZWave	Smart switch for controlling lights
ZWave	Smart motors for controlling blinds.
ZWave	Smart plugs for controlling and detecting usage of electrical appliances (kettle, microwave, TV, etc.)
ZWave	Cooktop power consumption sensor for detecting usage of cooker
ROS/Wifi/BLE/ZWave	The RADIO Home Cluster, a stack of three RaspberryPi devices, executing the Main Controller and the gateways to the BLE and ZWave networks.

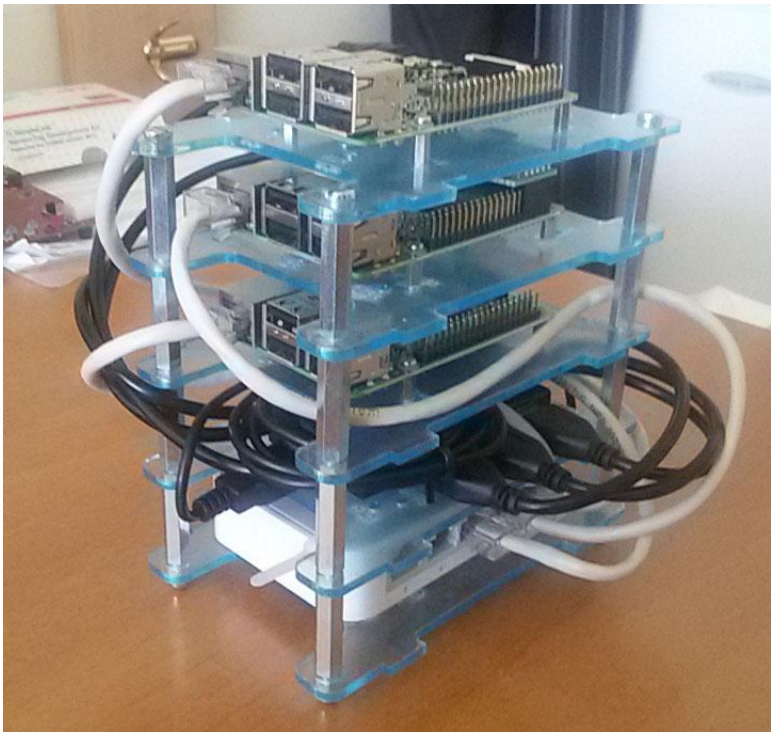

Figure 2: The RADIO Home Cluster

Figure 2 depicts the integrated Home RADIO Gateway, comprised of the following components:

- Main Controller: Raspberry pi 3 taking advantage of the on board WiFi wireless interface
- ZWave Gateway: Raspberry pi 2 which used the onboard WiFi wireless interface in conjunction with an integrated Z-Wave wireless interface
- BLE Gateway: Raspberry pi 3 using the onboard WiFi and BLE wireless interfaces.
- Ethernet switch

Effectively the this is a self-contained Raspberry cluster, requiring only one Ethernet connection to connect all components to the internet. The cluster is very efficient, with all components powered through an active 4-port USB3 hub, with a maximum power consumption of 18W.

3 PROTOTYPE SOFTWARE

The project has registered a Github organization (https://github.com/RADIO-PROJECT-EU) in order to gather and integrate all the Git repositories used for development. When adapting and extending existing software, either by one of the RADIO beneficiaries or by third parties, the original repository is forked into RADIO-PROJECT-EU and then the RADIO fork is used to track development.

3.1 Robot Packages

The robot software stack, delivered as *Integrated robotic platform* (D4.6), is repeated here for completeness.

Package name and description	Source code repository and release used for this deliverable	
turtlebot: The basic drivers for running and using a TurtleBot with ROS. Forked from the official Turtlebot repository and modified and adapted for the RADIO Robot, creating a new model of the robot and sensors.	https://github.com/radio-project-eu/turtlebot	
turtlebot_apps: A group of demos and examples used as templates for TurtleBot/ROS packages. Forked from the official Turtlebot repository and modified and adapted, creating configuration files specific to the RADIO Robot.	https://github.com/radio-project-eu/turtlebot_apps	
kobuki: Software controllers for the Kobuki mobile base. Forked from the official Turtlebot repository and slightly modified for the RADIO Robot.	https://github.com/radio-project-eu/kobuki	
kobuki_core: Software controllers for the Kobuki mobile base. Forked from the official Turtlebot repository and modified for the auto-docking procedure of the RADIO robot.	https://github.com/radio-project-eu/kobuki_core	
turtlebot_radio_bringup: Bringup files for the robot. It contains all the configuration and launch files to run all the RADIO Robot components.	https://github.com/radio-project-eu/turtlebot_radio_bringup	
robotnik_msgs: Definition of messages and services used by the core packages.	https://github.com/radio-project-eu/robotnik_msgs	
turtlebot_radio_emergency: Node that implements safety stop. It manages the emergency button of the robot and disables any movement.	https://github.com/radio-project-eu/turtlebot_radio_emergency	
marker_mapping: To avoid manual initialization, this node localizes visual QR markers onto the map and uses them inversely to calculate the correct pose of the robot.	https://github.com/radio-project-eu/marker_mapping	
HumanPatternRecognition: Recognizes human walking patterns in laser scans and tracks walking.	https://github.com/radio-project-eu/HumanPatternRecognition	v2.0

Table continues on next page.

Table continues from next page.

Description	Source code repository and release used for this deliverable	
HPR Wrapper: Uses HPR output to recognize and time "walked 4m" events.	https://github.com/radio-project-eu/hpr_wrapper	v1.0
ROSVisual: Tracks moving objects in the RGB/depth modality and classifies motion as bed or chair transfer.	https://github.com/radio-project-eu/ros_visual	v1.1
ROSVisual Wrapper: Uses the output from ROSVisual to time chair and bed transfer events and to recognize and time "walked 4m" events.	https://github.com/radio-project-eu/ros_visual_wrapper	v1.0
Motion Analysis: Recognizes motion and classifies it as "bed transfer" and "pill intake" events.	https://github.com/radio-project-eu/motion_analysis	v1.2
Motion Analysis Wrapper: Uses the output from motion analysis to time the bed transfer event.	https://github.com/radio-project-eu/motion_analysis_wrapper	v1.0
AUROS: Recognizes talking, watching TV, listening to music, and doing housework by acoustic analysis of the audio modality.	https://github.com/radio-project-eu/AUROS	v1.0
BLE Localization: Estimates position observing RSSI values. Implements filters, like Kalman filter, for normalizing RSSI values and predicting position.	https://github.com/radio-project-eu/robot_ble_localization	
Map convergence: Consumes robot pose messages from the localization package and converts them to the corresponding RADIO Home model coordinates.	https://github.com/radio-project-eu/map_convergence	

Further to these, the packages below have been developed specifically for the purpose of integrating the RADIO Robot into the RADIO Home.

Package name and description	Source code repository and release used for this deliverable	
Node manager: The robot-side node of the Node Manager [D4.2, Section 7.2].	https://github.com/radio-project-eu/radio_node_manager	v1.0
Anti-theft alarm: Identifies being picked up or moved from the robot's IMU, and sends alarm notifications to the RADIO backend system.	https://github.com/radio-project-eu/anti_theft_alarm	
REST/ROS bridge: Software interface between a REST service and ROS MoveBase, used for interfacing the first GUI prototype [D5.4, Section 2.3.2]. This has been deprecated, as the new GUI prototype directly accesses ROS.	https://github.com/radio-project-eu/radio_actions_manager	v1.0

3.2 Main Controller

The RADIO Main Controller is the main orchestrator of the behaviours of the RADIO Home and the main keeper of the information collected and analysed by the various RADIO Home systems. The packages that implement the functionalities of the main controller are listed below.

Package name and description	Source code repository and release used for this deliverable
Node manager: The home computer-side node of the Node Manager [D4.2, Section 7.2].	https://github.com/radio-project-eu/radio_node_manager_main_controller
Influx data service: The database that provides the RADIO Home's data services [D4.2, Section 7.5].	https://github.com/radio-project-eu/maincontroller
Report generator: The aggregator that goes through the events log in the data service to generate ADL reports [D4.2, Section 7.4].	https://github.com/radio-project-eu/radio_report_generator

The Main Controller also bridges between the ROS middleware/wifi network and the two other communication infrastructures present in the RADIO Home. The packages that implement these bridges are listed below.

Package name and description	Source code repository and release used for this deliverable
The REST API to the ZWave network of home automation sensors and actuators, via the S&C Gateway	https://github.com/radio-project-eu/snc_sensors_publisher
The MQTT middleware used by the BLE network	https://github.com/radio-project-eu/room_status_publisher

3.3 BLE Localization

Besides the robot-side packages listed above, the following packages have also been developed to support BLE localization and composite services over BLE localization.

Package name and description	Source code repository and release used for this deliverable
Multihop localization: Firmware code for the CSR µEnergy® CSR1010™ Development kit. The nodes are distinguished in fixed infrastructure and mobile nodes. Fixed infrastructure nodes inform the BLE Gateway of the objects that are in their vicinity and the value of the RSSI.	https://github.com/radio-project-eu/relative_multihop_localization
BLE Gateway: interface which connects IP networks with Heterogeneous Sensor Networks. Utilizes the MQTT for the communication between cloud applications, gateway applications, and all these applications to communicate with the WSN. Implementing functionalities, like local storages, routing, local data processing, applications execution.	https://github.com/radio-project-eu/ble_gateway
Indoor relative localization: A BLE Gateway application that communicates with BLE Gateway to estimate relative locations of BLE tagged objects. The application receives from the fixed nodes packets that record the surrounding Radio-enabled mobile BLE nodes along with the RSSI.	https://github.com/radio-project-eu/indoor_relative_localization

4 DEMONSTRATION

The first prototype of the RADIO Home was deployed in March 2017 at FHAG premises, in preparation for the first round of the piloting study.

A demonstration is at https://vimeo.com/211673690