Android Mobile Application for Controlling Arexx Robots

Szilárd-Gábor Mátis*, Károly Simon**, Levente Szélyes***
* Codespring LLC, Babeş-Bolyai University, Cluj-Napoca, Romania
** Codespring LLC, Babeş-Bolyai University, Cluj-Napoca, Romania
*** Codespring LLC, Cluj-Napoca, Romania
matis.szilard@codespring.ro
simon.karoly@codespring.ro
szelyes.levente@codespring.ro

Abstract—**The ASURO project is presented, the used hardware devices and software development tools, methodologies and technologies are described.**

The aim of the project is to create algorithms and software modules for controlling Arexx robots. Two robot types are considered, the ARX-03 ASURO and the AAR-04 ASURINO robot. The AAR-04 robot is the improved version of ARX-03, with Arduino platform, which makes the development of software modules much easier. In the framework of the presented project the robots are extended with special hardware modules (sensors and Bluetooth communication modules).

The project contains an Android mobile application, which provides the user interface for controlling the robots. The robots are equipped with a WiFi camera, the camera image can be visualized in real time with the mobile application, together with sensor information. The robots can be controlled by rotating the mobile device or with virtual buttons on the display.

I. Introduction

The Asuro project is working with Asuro and Asurino robots created by the Arexx Company. The project includes controller software modules for these robot types, and an Android mobile application. The main purpose of the project is to control the Arexx robots with the mobile application. The robots can also be controlled using proper desktop applications (e.g. HyperTerminal). The goal is to create a control mechanism which is independent from the controller device chosen by the user.

The robots can be commanded to move in any direction. They must change their speed increasingly, avoiding the "jerking" or "square" movements. For example: the robot moves forward with an x speed, and then receives a stop or a slowdown command. This is not performed immediately, but gradually reducing the speed, until the desired speed is reached. Similarly, in the case of acceleration the speed is also increased gradually.

The robots should inform the user in the case of a sensor event (collision, object detection with the ultrasonic distance detector sensor, events received from the minesweeper sensor etc.). The robot must react intelligently on some sensor events. For example, in the case of a collision the robot must stop. Using the distance detector sensor three distance levels can be differentiated. In the case of the first two levels (green and yellow levels) the robot only informs the user about the events. In the case of the third (red) level the robot will interrupt its movement. The movement remains suspended, while there is a risk of collision. The robot should also inform the user about minesweeper sensor events.

A WiFi camera is installed on the robot. The image of the camera can be visualized in real time by the Android mobile application. The Android application also shows the sensor information on its screen. This information is displayed on the camera image.

The Android application communicates with the robots via a Bluetooth connection. The application must automatically connect to the robot, and to the camera. In the case of a connection error the user should be informed by the application. The user can choose from multiple control modes: controlling the robots by rotating the mobile device or controlling the robots using virtual buttons displayed on the screen of the application (above the camera image). The settings of the application can be changed from the configuration screen.

II. Hardware Devices

Within the framework of the Asuro project two robot types are considered: the ARX-03 ASURO and the "smarter" AAR-04 ASURINO Arexx robots. The central part of the Arexx robots is the microcontroller. The ASURO ARX-03 robot is equipped with an Atmel Atmega8 microcontroller [1], while the AAR-04 robot has an Atmel Atmega328P [2] microcontroller.

The standard equipment of the ASURO [3] ARX-03 robot (Fig. 1) contains two separately controllable motors, six collision sensor (on the front of the robot), a line tracking sensor and three LED diodes. The robot can be equipped with optional sensors and tools (e.g. distance detector, minesweeper sensor, etc.).

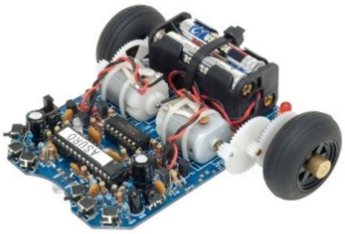

Figure 1. The ASURO robot with standard equipment

The standard version of the ASURO robot can be programmed and controlled via an infra port. In the presented project the robots are equipped with a Bluetooth module (Fig. 2), which helps to control them from a greater distance.

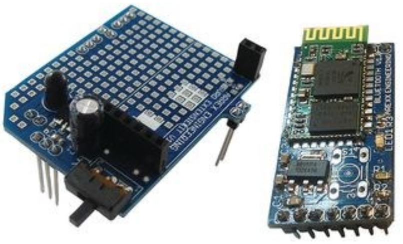

Figure 2. Bluetooth module

The ASURO ARX-03 robot is also equipped with an Arexx Engineering ARXULT10 Ultrasonic module and with an Arexx Minesweeper Kit (Fig. 3). With this equipment the robot can detect objects in front metals under itself. Due to some hardware limitations, the line tracker and the minesweeper sensor cannot be used at the same time.

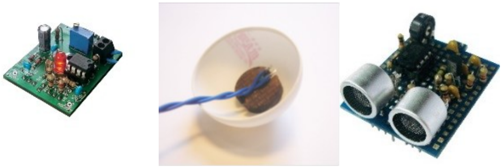

Figure 3. Arexx Minesweeper Kit and Arexx Engineering ARXULT10 Ultrasonic module

The ASURO robot can be programmed in C programming language. The GNU C-Compiler can be used for compiling the code. The programs can be uploaded via the infra port using the ASUROFlashTool application.

The ASURINO (AAR-04) robot (the official name is Arexx Arduino Robot [4]) (Fig. 4) is also designed by the Arexx Company and it is compatible with the Arduino platform. Using the Arduino platform the robot can be programmed more easily. It also provides some basic libraries, which helps to create the controllers in a more effective manner, the AAR-04 can be programmed and controlled via USB - by default, but the robot is also compatible with the ASURO extension kits, and therefore it can also be controlled via Bluetooth.

As standard equipment, the AAR-04 robot has a wheel sensor and a line tracker sensor. It does not have collision sensors.

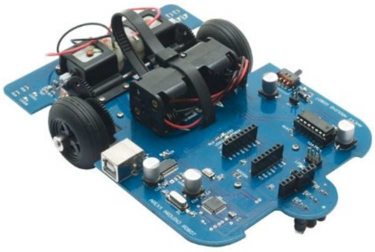

Figure 4. The ASURINO robot

For image recording, an EGO WiFi Model 727 camera (Fig. 5) was installed on the robots. It is a WiFi camera manufactured by the Liquid Image Company, and it can take pictures and videos in HD quality.

Figure 5. EGO WiFi camera

By default, both robot types are equipped with AAA accumulators. Using this power supply, Bluetooth communication problems were encountered. As a consequence, within the framework of the Asuro project the AAA accumulators are replaced by AA accumulators.

III. DEVELOPMENT TOOLS AND METHODS

The Asuro project development process is based on Agile [5] principles, using elements from the Scrum methodology [6]. In the first step of the development a product backlog was created with user stories describing the main functionalities. This list was managed by the product owner during the implementation.

The source code is managed by a Mercurial distributed version control system. Tortoise HG is used as a Mercurial desktop client application. The RhodeCode system is used for central repository management.

Redmine is used as a project management and issue tracking system. XWiki is used as a wiki software platform for information and documentation sharing between the developers.

For developing the Android client application the Eclipse IDE is used, together with the Android ADT Eclipse plugin. The controller software for the Asuro robot is developed in Atmel Studio 6, an IDE based on Visual

Studio. The ASURINO controller software is also created in Atmel Studio, using the Arduino IDE plugin as an extension.

IV. USED TECHNOLOGIES

The controller software for the Asuro robot is developed in C programming language. The manufacturer of the robot provides some libraries for implementing basic functionalities (e.g.: initial settings, motor controller methods, sensor functions). Within the framework of the Asuro project these libraries are modified, and some other function libraries are created for providing the required new functionalities.

The software for the ASURINO robot is created using the Arduino platform [7]. The platform creates a higher abstraction level and makes the development easier. The platform also provides some general functionalities, working in the same way on different developer boards. A simplified C++ programming language is used for development. The Arduino platform also provides a cross-platform development environment, the Arduino IDE (developed in Java).

The mobile application is developed for the Android platform [8].

V. THE ASURO PROJECT

A. *Architecture*

The project contains two main components: the controller modules for the Arexx robots and the Android mobile application (Fig. 6). The Android application communicates with the robots and it also communicates with the EGO Wifi camera installed on the robots.

The robot controller software module is separated into four sub-components. The first part contains the ASURO/Arduino libraries, providing the basic functionalities. The second part is the ASURO sensor module, which provides functionalities related to the sensors. The ASURO Movement Service provides functionalities for motion calculation and message decryption. The central layer is the ASURO Main component. This layer contains the code for initializations and the main program. The movement actions and sensor operations are also scheduled within this layer.

The second component is the Android mobile application. It contains the GUI, a Service layer, a Robot Control layer and a Robot Communication API.

The UI communicates directly with the Service layer, which is separated into two parts: a Camera Service and a Robot Service. The Robot Control layer is responsible for processing sensor information, and for sending the proper movement commands to the robot. The Robot Communication API is responsible for establishing the connection with the robot and for managing the communication. The Bluetooth API is used within this layer for creating and maintaining the Bluetooth connection.

The Otto Event Bus technology is used for event handling, propagating events from the lower levels to the higher levels.

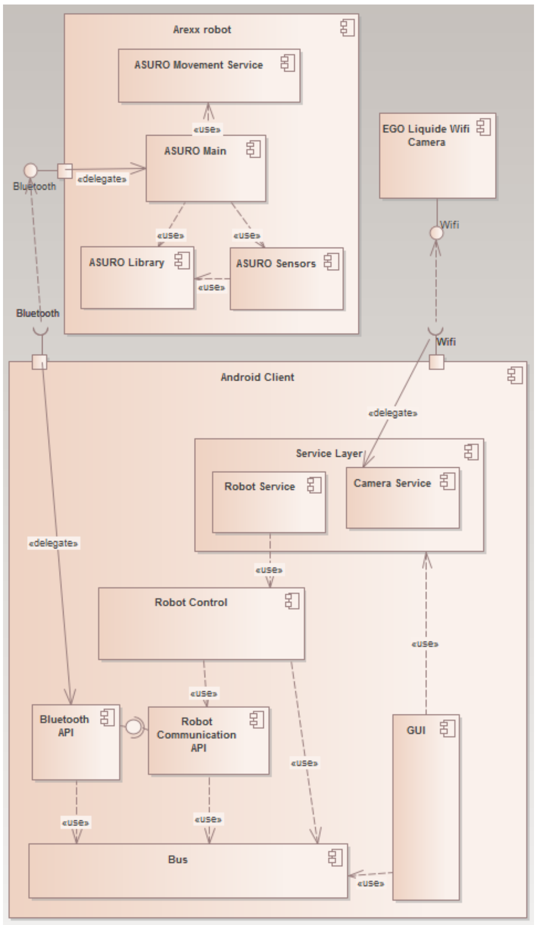

Figure 6. The ASURO project architecture

A communication protocol has been created for the robots. The most important aspect of the protocol's design is simplicity. The aim is to create an effective method for exchanging messages between the robots and the controller clients, and to provide an easy way for encrypting and decrypting messages. The robot uses a two-way communication with the connected client, they exchange messages related to movements and sensors.

B. *Using the Asuro mobile application*

After launching the application, the software tries to connect automatically to a saved robot instance. If the robot is not in range, then a popup window will inform the user about the problem. In this case a connection cannot be established and the user cannot switch to the control screen.

The application can be configured using the settings screen (e.g. camera settings, the Bluetooth ids of the robots, control settings).

The application can be configured using the settings screen (e.g. camera settings, the Bluetooth ids of the robots, control settings).

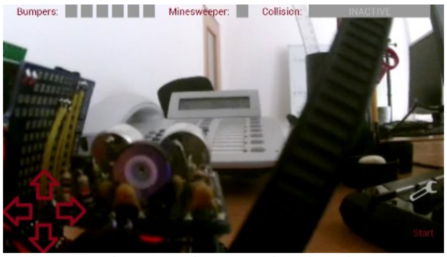

Figure 7. The control screen showing the camera image and sensor information

In the case of any exceptions the user will be notified with a proper error message.

On the top of the control screen the sensor information can be visualized: six collision sensors, the minesweeper sensor and the distance detector sensor information. If a sensor event occurs, then the corresponding indicator color will be changed on the screen. The distance detector sensor shows four distance levels with different colors:

- Gray: no object detected
- Green: an object is detected, the distance is approximately 10 cm
- Yellow: the detected object is closer, the distance is between 4 and 5 cm
- Red: the detected object is very close, the distance is between 0 and 3 cm, and the movement will be suspended to avoid collision

The robots can be controlled by rotating the mobile device or by using the virtual buttons displayed on the bottom-left side of the screen. Only one control mode can be active at the same time, and the button control has a higher priority. On the bottom-right side of the control screen a start/stop button is displayed (for suspending/resuming the robot control).

VI. CONCLUSIONS AND FURTHER DEVELOPMENT

Within the framework of the Asuro project controller modules were implemented for Arexx robots and a mobile application was developed for controlling these robots.

During the development several problems and difficulties were encountered (e.g. camera picture delay, Bluetooth communication problems, etc.). For resolving the mentioned problems and for including new functionalities there are further development and improvement possibilities:

- The project can be extended with AI modules. The robots could react intelligently in different situations. For example: using the ultrasonic distance detector the robots could automatically bypass blockages. Another possibility for AI-based reactions is using the camera picture and image recognition for orientation and object detection. For this functionality the delay of the camera picture must be eliminated or minimized (for example by using other camera device).
- The project can be extended on hardware level with a smarter control unit, for example a Raspberry Pi. By using a Raspberry Pi an abstraction level can be created between the robots and the mobile application. The Raspberry Pi can help the robot in some calculations, and on the other hand it can be extended with a camera, in this way the camera information could be processed without delay.
- Using the described hardware multiple difficulties were encountered. These problems could be resolved by the construction of a new robot hardware, controlled for example by a Raspberry Pi.

ACKNOWLEDGMENT

The authors would like to thank Codespring LLC for supporting the development process and for ensuring the required infrastructure.

REFERENCES

[1] ***, The Atmel official site [Online], Atmega8 microcontroller description, Available: http://www.atmel.com/devices/atmega8.aspx

[2] ***, The Atmel official site, Atmega328p microcontroller description [Online], Available: http://www.atmel.com/devices/atmega328p

[3] ***, Assembly and Operational MANUAL Model ARX-03, Arexx Engineering, 2011.

[4] ***, Arexx Arduino Robot Manual: AAR-04, Arexx Engineering, 2012.

[5] Robert C. Martin, Agile Software Development, Principles, Patterns, and Practices, Pearson Education, October 15, 2002.

[6] Jeff Sutherland, Scrum: The Art of Doing Twice the Work in Half the Time, Crown Business, September 30, 2014.

[7] Brian W. Evans, Arduino programing Notebook, August 2007.

[8] Zigurd Mednieks, Laird Dornin, G. Blake Meike, Masumi Nakamura, Programming Android, 2nd edition, Sebastopol, California, O'Reilly Media, September 2012.