

# GPRS-based Real-Time Remote Control of MicroBots with M2M Capabilities

Diego López de Ipiña<sup>1</sup>, Iñaki Vázquez<sup>2</sup>, Jonathan Ruiz de Garibay<sup>3</sup> and David Sainz<sup>3</sup>

Faculty of Engineering, University of Deusto, Bilbao, Spain

<sup>1</sup>dipina@eside.deusto.es, <sup>2</sup>ivazquez@eside.deusto.es, <sup>3</sup>joruiz@tecnologico.deusto.es and <sup>4</sup>dsainz@tecnologico.deusto.es

**Abstract.** Machine to Machine (M2M) communication is gathering momentum. Many network operators deem that the future of the business on data transmission lies on M2M. In parallel, the application of robotics is progressively becoming more widespread. Traditionally, robotics has only been applied to industrial environments, but lately some more exoteric (e.g. domestic) robots have arisen. Anyhow, those robots usually offer very primitive communication means. Few researchers have considered that a solution to this issue would be to combine those two emerging fields. This paper describes our experiences combining M2M with robotics to create a fleet of MicroBots, which are remotely controllable through GPRS connection links. Those robots can be used in dangerous environments to gather material samples or simply for surveillance and security control. A sophisticated 3-tier architecture is proposed that combined with a purposely built protocol, optimized for wireless transmission, makes feasible the real-time control of remote devices.

## 1 Introduction

Telemetry [1], i.e. remote monitoring and control of machines, is increasable becoming popular. For instance, telemetry allows the operation of a whole factory to be controlled from a single computerised station, accessible even remotely from the factory premises.

The possibilities of Machine to Machine (M2M) [2] communication are ample. Some application areas are: telemetry, data collection, remote control, robotics, maintenance and security, logistics or telemedicine. For example, M2M is very useful for vending machine providers to control when the stock in their machines needs refilling. In fact, mobile telephony operators consider that wireless M2M may definitely foster the consumption of data over their networks.

Robotics is slowly moving from exclusively industrial application domains to more esoteric fields, such as domestic [3] or air space [4] ones. The Mars Rovers, unmanned land vehicles for exploration of the planet Mars, are probably the most popular robots in the world. Those NASA produced robots posses sophisticated communication mechanisms which have enabled them to send us pictures of the

surface of Mars. Unfortunately, most conventional robots have very primitive (if any) communication mechanisms. The technology used for data transmission by the NASA robots is not commonplace. Thus, it would be interesting to correct this deficiency by making use of readily off-the-shelf technology such as M2M and GSM/GPRS.

This paper describes our experiences combining the promising fields of robotics and M2M, in order to achieve real-time telemetry over a GPRS data network. We have named the deliverable of this research COMMunication MicroBots (COMMBOTS, from now on).

The structure of this paper is as follows. Section 2 explains the COMMBOTS concept. Section 3 describes the sophisticated multi-layer architecture designed for COMMBOTS. Section 4 describes the MicroBot Protocol (MP) designed for the wireless communication among Controls Stations and MicroBots. Section 5 offers some performance results of the system. Section 6 reviews some related work. Finally, Section 7 concludes the paper and suggests some further work.

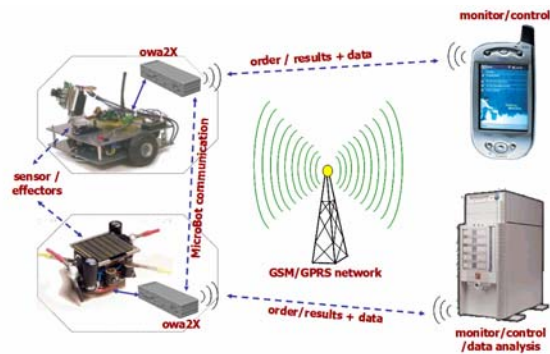


Fig. 1. COMMBOTS concept.

## 2 COMMBOTS

The COMMBOTS project aims to create a group of MicroBots equipped with mobile communication and location (GPS) facilities. Those robots can be applied to carry out emulations and activities in different environments, such as dangerous environments where they can gather data and materials, or for surveillance and alarm control in security zones.

Fig. 1 shows the two component types that constitute COMMBOTS: (1) MicroBots and (2) Control Stations.

The MicroBots are devised to carry out diverse tasks, such as: obtaining and measuring data by sensing, collecting samples, or recording and transferring images to a central Control Station. A MicroBot is bundled with a mobile communication module (GSM/GPRS), by means of which it can receive commands from the control centres and send responses.

The Control Stations in COMMBOTS are available in two forms: (1) fixed stations operated from a PC and (2) mobile ones operated from a PDA or mobile phone. These Control Stations allow a user to monitor and control at *anytime* and *anywhere* the whereabouts and actions of a fleet of MicroBots. Likewise, the MicroBots may notify events of interest to the Control Stations (temperature too high, movement detected) or delegate the processing of the data captured to third parties.

The major contribution of COMMBOTS is to combine the latest developments in M2M and MicroBotics, to design a real-time remote control system operated over public data networks (GPRS).

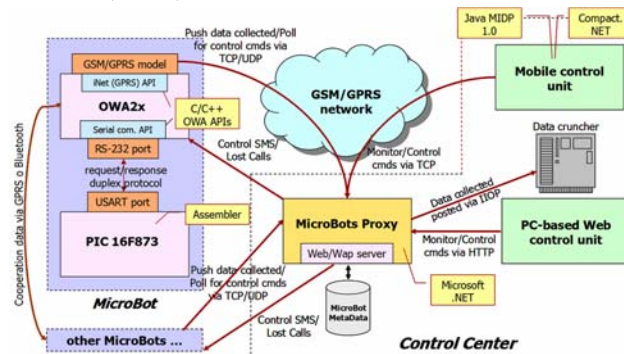


Fig. 2. COMMBOS concept.

### 3 The COMMBOTS Architecture

COMMBOTS presents a *client/server/client* architecture aimed to ease end-to-end communication between Control Stations and MicroBots. Fig. 2 shows the different components of this architecture together with the data and control flows exchanged among them. In what follows, we detail those components: (1) MicroBots, (2) MicroBots Proxy and (3) Control Stations.

#### 3.1 MicroBots

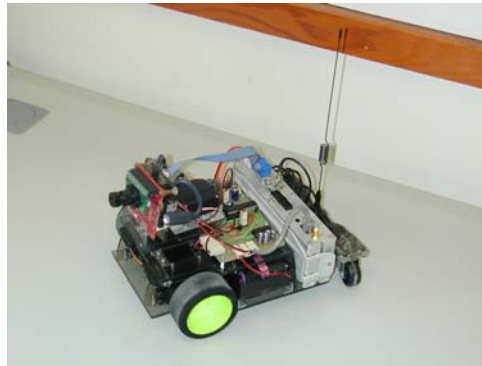
Following the human body analogy, a MicroBot in COMMBOTS is composed of (see Fig.3):

- *Extremities*, i.e. the wheels together with the engines that move them and give mobility to the bot.
- *Senses*, i.e. sensors to capture information from its environment: luminosity, temperature, images, and so on.
- *Body*, i.e. the framework that gives weight and stability to the bot and acts as container of its different hardware components.

- *Brain*, i.e. the microcontrollers that manage the engines and sensors bundled with the MicroBot, together with a communications module, such as the owa22A[2], which allows data transmission with other robots and stations.

The control system of the MicroBot is composed by two entities which coordinate their operation: a PIC16F873 [5] microcontroller, manufactured by MICROCHIP, and the owa22A communication module, manufactured by OWASYS [2]. The coordination between those two devices is achieved by a specially designed protocol. This protocol uses 1 or 2 bytes long messages which are exchanged through the RS232 ports of both devices.

Given the limited processing power of the microcontroller, the MicroBot's CPU intensive tasks are undertaken by the owa22A, equipped with a 60 MIPS CPU ARM7 and 2 MB RAM. The tasks undertaken by this module are: (1) communication between the MicroBot and the MicroBots Proxy and (2) capture of digital images from the attached CMUcam2 camera [6] and their compression before delivery. The microcontroller undertakes simple tasks that would unnecessary delay the owa22A module, such as control commands over the engines, sensors and effectors integrated in the MicroBot.



**Fig. 3.** The COMMBOTS MicroBot

### 3.2 MicroBots Proxy

The creation of this component which intermediates between MicroBots and Control Stations is justified by the following reasons:

- The MicroBots operate under battery. Thus, it is convenient to reduce their power consumption by avoiding having their communications modules actively listening to requests.
- Many telephony operators assign NAT (Network-Address-Translation) addresses to GPRS connected devices. Those IP addresses are not accessible outside the network operators LAN.
- Frequently, the mobile operators raise firewalls and other obstacles to avoid connections to port 80 of GPRS mobile devices.

- Common functionality shared by both the static and mobile Control Stations can be factored out and placed within the business logic of a single component, namely the MicroBots Proxy. Thus, this component:
  - Provides a cache to avoid redundant communications with the remote MicroBots. The proxy caches the current status of the MicroBot, i.e. its latest sensor values. So, the communication through GPRS will be reduced in one step, minimizing the cost and reducing the latency.
  - Ensures the fault tolerance of the MicroBot communications. In a transparent fashion to the Control Stations, it is able to restart communications among itself and the MicroBots.

The MicroBot communication module always initiates the connection with the proxy. Between a communication module and a proxy the following three connections are established:

1. *Connection for the reception of MP commands and the delivery of responses.* The owa22A module, once the connection is opened, blocks waiting for the arrival of commands.
2. *Connection for the delivery of captured images.* Once the MicroBot camera is activated, the owa22A module continuously transmits images to the proxy, where they are cached.
3. *Connection for the delivery of alerts.* Changes produced in the state of the MicroBot (temperature, luminosity, battery status, etc.) are notified to the MicroBots Proxy by means of this connection.

In order to prevent the MicroBot from having to keep opened a GPRS connection, with the associated waste of battery power, even when a MicroBot is not being used (controlled), the MicroBots Proxy allows for the delivery of control text messages, which activate/deactivate and parameter the connection opened with the MicroBot (e.g. the longest period a connection can remain opened). Moreover, the owa22A modules have been programmed to connect/disconnect to the MicroBots Proxy when they receive a lost call from the proxy phone number. The SMS messages and lost calls are generated by a pair of web services we have made available at <http://www.ctme.deusto.es>.

In essence, the MicroBots Proxy, acts as router of the requests arriving from the Control Stations towards the MicroBots and, at the same time, propagates the data generated by the MicroBots to the Control Stations.

### 3.3 Control Stations

The following varieties of Control Stations have been designed:

- The *Web/WAP Control Stations* transmit data through HTTP/WSP in XHTML or WML format. The interface of these stations is generated dynamically by the web component of the MicroBots Proxy. This web component uses the ASP.NET Mobile Web Controls framework [7], which adapts the mark-up pages generated to the target user agent (PC, PDA or mobile). Fig. 4 shows the same interface displayed on a mobile WAP browser, a web browser in a PDA and a web browser in a PC, respectively. Given that the browser in a Pocket PC emits the HTTP

Accept-Encoding: gzip, deflate header, i.e. it accepts XHTML content in compressed format, the MicroBots Proxy web component compresses the information before delivering it over GPRS.

- The *Pocket PC Control Stations* use TCP to transmit in binary MP commands to the MicroBots Proxy. This component has been implemented with Compact.NET [8], a .NET API for the development of applications in mobile devices. The left hand side of Fig. 5 shows the appearance of the interface for Pocket PC.
- The *J2ME Control Stations* [9] run in every phone compatible with MIDP 1.0, i.e., in the majority of the currently sold mobile phones. The communication with the proxy uses TCP sockets which transfer MP command and responses in binary format. The right hand side of Fig. 5 shows the J2ME interface.



Fig. 4. Interfaces WAP (mobile) and web (PDA/PC) of a Control Station



Fig. 5. Compact.NET front-end for a PDA (left) and J2ME interface (right) for Control Station.

#### 4 The MP Protocol

The data sent between the MicroBots Proxy and Control Stations, as well as between the owa22A modules and the proprietary Control Stations (those not using a web browser), conforms to the MicroBot Protocol (MP). Its design has followed a double objective: (1) use the smallest size messages and (2) avoid every confirmation or

message communication not strictly necessary. This is to (1) incur in the minimum possible GPRS costs and (2) to reduce the required bandwidth as well as the communication latency.

MP follows the WBXML (WAP Binary XML) [10] principles to reduce the size of the messages. The command/response latency is improved by minimizing the messages delivered. A well known fact is that TCP in wireless environments behaves better with few bigger segments than many small ones [11]. TCP was designed for wired networks where latency is not a paramount factor, unlike the wireless case.

The MP commands are classified in three categories:

1. *Movement commands.*

- Nine different movement actions (left forward, forward, right forward, left, stop, right, left backward, backward and right backward) are available.
- The speed can be set to fast, normal or slow.

2. *Sensor commands.*

- *Temperature:* commands to retrieve the current temperature and to set a notification alarm when the temperature moves outside a range. The user may activate the automatic temperature control by setting an upper and lower range.
- *Luminosity:* commands to retrieve the current luminosity level and to set the automatic luminosity control.
- *Anti-collision:* commands to activate/deactivate the collision detection control, which prevents the MicroBot from colliding against walls and obstacles
- *Batteries:* commands to retrieve the current battery value and receive alerts when the value is under a threshold.

3. *Effector commands.*

- *Digital camera:* commands to retrieve the images captured by the CMUcam2 camera attached to the RS-232 port of the owa22A, and to move it around. The MicroBot can move the camera up, left, center, right and down.
- *Lights:* commands to explicitly or implicitly (when the luminosity is below a threshold) switch on/off the lights of the MicroBot.

The following MP commands are sent between the Control Stations and the MicroBots Proxy, and between this latter one and the MicroBots:

- `MODULE , STATION:` it allows a client (module or station) to connect with the proxy. The telephone number of the module connecting or to connect to will be specified. For instance: `MODULE +34609421898.`
- `LIST:` it enumerates all the sensors and effectors allowed in the MicroBot.
- `GET <control-element> [<param>(,<param>)*]:` indicates the sensor to obtain information from. For instance, `GET TEMP_LOW_VALUE` or `GET MV_MC_UP.`
- `SET <control-element> [<param>(,<param>)*]:` indicates the element to control and the values to assign. For instance, `SET TEMP_LOW_VALUE 20`

On the other hand, the MicroBots respond to commands in the following format:

```
<response-code> <response-msg> CR-LF
Content-Length: <response-size> CR-LF
message-body
```

Noticeably, the MP format is very similar to HTTP [12]. However, the body of the message will be delivered in binary format for performance reasons. It is not required to specify the MIME type of the response, because each request waits for a response in a predefined manner. Only the Content-Length header is required to facilitate processing. The <response-code> in the answer indicates whether the request was satisfied or not.

MP is a very simple and easily extensible protocol. By means of the LIST message, all the elements which can be configured (SET command) and queried (GET command) can be retrieved. The LIST <control-element> command enumerates all the parameter types allowed to configure and retrieve data from a control element.

## 5 Test Results

Table 1 illustrates the average latencies encountered while issuing MP movement commands to a MicroBot from three different Control Stations: (1) the WAP browser of a Nokia 6600 mobile, (2) the proprietary client running on a TSM 500 Pocket PC and a (3) MIDP client running on a Nokia 6600. Observe that the best results are obtained by using the MIDP client in the mobile phone, probably due to a more efficient implementation of the TCP/IP stack in the mobile phone than in the PDA. The worst results are obtained by the web client. Obviously, this is due to the fact that apart from the commands and their responses, it is necessary to deliver the mark-up of the pages to visualize. Moreover, the data transfer goes over HTTP, a level higher in the protocol stack than TCP (used by the proprietary clients).

On the other hand, Table 2 compares the latency encountered in the transfer of an image from the MicroBots proxy to the proprietary Control Stations via TCP and to the web client through HTTP. Evidently, the best results correspond to the proprietary clients using the MP protocol over TCP rather than the less efficient HTTP.

**Table 1.** Delivery of movement commands (50 bytes in size) between Control Stations and MicroBots.

| Device                      | Mean (secs) | Standard Deviation (secs) |
|-----------------------------|-------------|---------------------------|
| Mobile Phone MIDP 1.0 (TCP) | 2.68        | 1.23                      |
| PDA Compact .NET (TCP)      | 4           | 1.67                      |
| PDA Web (HTTP)              | 5.54        | 1.36                      |

**Table 2.** Delivery of a 2.3K image between the MicroBots Proxy and the Control Stations.

| Device                      | Mean (secs) | Standard Deviation (secs) |
|-----------------------------|-------------|---------------------------|
| Mobile Phone MIDP 1.0 (TCP) | 3.89        | 2.05                      |
| PDA Compact .NET (TCP)      | 3.724       | 2.722                     |
| PDA Web (HTTP)              | 6           | 1.91                      |



## 6 Related Work

The PocketCERO [13] project has devised PDA interfaces that allow service robots, those that assist people with special needs (e.g. elderly people), control the elements in their environment. The COMMBOTS system was thought with a more industrial purpose in mind. However, we are currently working on the definition of a new kind of COMMBOTS, denominated CareBot, which will fulfil a similar care taking role.

The special requirements of the mobile communications with robots have been studied thoroughly by other authors [14], which have also considered the use of WAP-enabled phones for robot telemetry [15]. We have followed their guidelines in the design of our MP protocol.

The KDDI mobile operator has also considered the combination of mobile phones and microbots with Pirkus [16], a Bluetooth mobile phone controlled robot. In COMMBOTS, we use GPRS instead, since it allows the control of microbots from remote locations. This approach was previously followed by Fujitsu's Maron [17], an internet-enabled home robot capable of undertaking remote household monitoring, remote control of household appliances, or to serve as a hands-free phone.

## 7 Conclusion and Further Work

With the implementation of COMMBOTS we have experienced the enormous potential brought forward by the combination of two emerging fields: Robotics and M2M communication. COMMBOTS has demonstrated the feasibility of adding flexible, off-the-shelf, always available communication mechanisms to MicroBots.

Our research prototype should now be transferred to industrial applications. For instance, the integration of M2M communication modules extended with sensors in machines could allow their manufacturers to undertake preventive maintenance. They could remotely check the "health" of a machine sold and send the required technical staff to repair the problems even before the customers reported them. COMMBOTS may be used as basis for future research projects aiming to remotely control and monitor, in real-time, MicroBots or coordinated-machinery from a Control Station via public communication networks (GSM/GPRS).

A useful extension of COMMBOTS would be to enable the utilisation of local communication technology (PAN networks) when the control stations and the Microbots are close to each other (within Wi-Fi or Bluetooth range). Moreover, the cooperation among communicating microbots could be considered.

Our development of different wireless clients for COMMBOTS has allowed us to study the feasibility of achieving real-time control of remote devices by means of GPRS. The results obtained have shown that the control through web/WAP from a mobile station is not feasible when real-time responses are required. Hopefully, when UMTS data connections become the norm this issue will be solved. However, the design of a protocol specially catered for wireless communication together with the optimizations and caches of the central component in our 3-tier architecture, namely the MicroBots Proxy, have shown that the real-time control of remote devices by

means of GPRS is possible from wireless Control Stations. The results and source code of the COMMBOTS system are publicly available at [18].

## Acknowledgements

This work has been financed by a SAIOTEK 2003-04 grant from the Basque Government. In its implementation we have used resources facilitated by the Cátedra de Telefónica Móviles at the University of Deusto (<http://www.ctme.deusto.es>).

## References

1. Telemetry definition, Wikipedia, <http://en.wikipedia.org/wiki/Telemetry>, 2004
2. OWASYS owa2x M2M modules Family, [http://www.owasys.com/en/BOK\\_100\\_1000\\_R3D.pdf](http://www.owasys.com/en/BOK_100_1000_R3D.pdf), 2004
3. When Robots Rule the World, <http://www.wired.com/news/technology/0,1282,65408,00.html>, Wired News, October 2004
4. Mars Exploration Rover Mission, <http://marsrovers.jpl.nasa.gov/home/>, NASA, 2004
5. Microchip PIC 16F87X Data Sheet, <http://ww1.microchip.com/downloads/en/DeviceDoc/30292c.pdf>, Microchip Technology, 2001
6. The CMUcam2, <http://www-2.cs.cmu.edu/~cmucam2/>, Robotics Institute, Carnegie Mellon University, 2004
7. Microsoft ASP.NET Home Page, <http://asp.net>, Microsoft Corporation, 2004
8. Mobile Application Development Toolkit, <http://www.microsoft.com/downloads/details.aspx?FamilyID=f4328333-0fd4-4348-88c0-39d10fb64f0a&DisplayLang=en>, Microsoft Corporation, 2004
9. Java 2 Platform, Micro Edition (J2ME), <http://java.sun.com/j2me/>, Sun Microsystems, 2004
10. WAP Binary XML Content Format, <http://www.w3.org/TR/wbxml/>, W3C, June 1999
11. Xylomenos G., Polyzos G. and Mahonen P. TCP Performance Issues over Wireless Links, IEEE Communications Magazine, April 2001
12. Hypertext Transfer Protocol, <http://www.w3.org/Protocols/>, W3C, 2004
13. Hüttenrauch H. and Norman M, PocketCERO – Mobile Interfaces for Service Robots. Proceedings of Mobile HCI 2001: Third International Workshop on Human Computer Interaction with Mobile Devices, Dunlop, September 2001
14. Gage, D.W, Network Protocols for Mobile Robot Systems, SPIE Proceedings, 3210. Presented at Mobile Robots XII, Pittsburgh, PA, 14-17 October, 1997
15. Sgouros N. and Gerogiannakis S., Integrating WAP-Based Wireless Devices in Robot Teleoperation Environments, Proceedings of IEEE ICRA 2002, pp. 1191-1196, ISBN 0-7803-7273-5}, 2002
16. Pirkus: Bluetooth Mobile Phone Controlled Robot, <http://www.blueserker.com/html/modules.php?op=modload&name=News&file=article&sid=290&mode=thread&order=0&thold=0>, 2005
17. Fujitsu, PFU Launch Initial Sales of MARON-1 Internet-Enabled Home Robot to Solutions Providers in Japan Market, <http://pr.fujitsu.com/en/news/2003/03/13.html>, 2003
18. COMMBOTS website, <http://www.ctme.deusto.es/COMMBOTS>, Universidad de Deusto, 2004