Mobile Telemetry System with Geo-location using a PDA

Bruno Cavaco Antunes\textsuperscript{1}, Mário Serafim Nunes\textsuperscript{2}, Rui Santos Cruz\textsuperscript{3}

\textsuperscript{1}Instituto Superior Técnico, \textsuperscript{2}Instituto Superior Técnico - INOV, \textsuperscript{3}Instituto Superior Técnico – Zapp

b.m.antunes@gmail.com, mario.nunes@inov.pt, rui.cruz@ieee.org

September 2007

Abstract—The objective of this project is to build a prototype of a telemetry system, using a device with weak processing power and a wireless connection to the Internet.

The system integrates a PDA, a RFID reader, a GPS receiver and a PC. Using some of the communication interfaces of the PDA, connected to the Internet using the CDMA2000 technology, it is possible to collect location data, read barcodes and RFID codes and send it to the Control Centre. The control centre is implemented on a PC connected to the Internet.

The main goal of the project is the development of the software for the PDA and for the Control Centre. All the software was developed so that it is easy to take advantage of the large diversity of communication interfaces of the PDA to increase the features of the system.

It was possible to build a robust system, resilient to many of the problems that happen when a wireless system is used and having a good number of interesting features.

Index Terms—Telemetry, CDMA2000, Barcode, RFID, GPS

INTRODUCTION

A telemetry system captures data from one or more sources and allows their remote collection.

In modern society, having a high productivity rate is a critical factor for every company. That is why, more and more companies in every branch of the industry, commerce and services use telemetry systems to monitor many aspects of their organization, in order to optimize the way they work.

Taking as an example a trucking company, by using a simple mobile telemetry system to collect location data from a GPS receiver and send it, via a wireless connection to the Internet, to the company’s Control Centre, it is then possible to know in almost real-time all of the fleet’s location, speed, track, etc. This allows any company of this segment to coordinate and manage their fleet more easily and react earlier to organizational problems.

Taking as another example a security company, by using a mobile telemetry system to collect barcodes strategically located in well known checkpoints and send them via a wireless connection to the Internet, it is possible to monitor the patrols in almost real-time. Additionally it may allow dynamic changes on patrol tracks ensuring that no agent is skipping patrol checkpoints.

There are many more examples that could be given, some of them more complex, where data is collected from more than one source. As an example, a delivery company can put barcodes or RFID tags in every package to deliver and put a GPS receiver in all their vehicles. It is then possible to develop a system that
controls the location of all the fleet, allowing the company to know where and when the deliveries were made, and for each vehicle, the packages that were not delivered yet, etc.

**OBJECTIVES**

The main goal of this project is to develop a prototype of a mobile telemetry system that collects data from several sources and sends it to a Control Centre by a wireless connection to the Internet. The data is collected from an industrial PDA (Personal Digital Assistant) that possesses several communication interfaces, allowing simultaneous collection of data from diverse sources of information. The data from several PDAs is received concurrently by a Control Centre, (a PC connected to the Internet).

The main assignment is the development of the software, both for the PDA and for the Control Centre, according with the requirements of the chosen data sources, always having in mind the possible increase in the number of data sources and consequent increase in the diversity of data collected.

The PDA is connected to the Internet using the 3G CDMA2000 wireless network, which coverage in Portugal, is assured by Zapp/Radiomóvel. The data sources chosen for this prototype were a GPS Receiver, a barcode reader and a RFID Reader.

Using these three types of data sources, it is possible to use the system in many areas, like security (identifying checkpoints to assure patrols), transportation (to locate vehicles), warehousing (to identify and locate merchandise), medicine (to identify patients and control medication), or retail sales (identifying merchandise to aid in stock management, billing and theft prevention), among others.

**INNOVATION**

This project has several aspects that can be considered innovative, being the most relevant the fact of the use of a 3G CDMA2000 wireless network from Zapp/Radiomóvel for the WWAN communications with the Control Centre. This technology, implemented in the 450 MHz Band, provides 3G services with national coverage (>80%), far superior to the coverage currently reached by other deployed 3G UMTS systems in the country.

Another important aspect is the communication protocol used between the PDA and the Control Centre, due to its flexibility. Joining this to the features available in the industrial PDA, that possesses several communication interfaces, it becomes facilitated the implementation of new features.

Regarding the Control Centre, it is possible to view the received data not only in regular tables, but also in Google Earth and http://maps.google.com.

It is also possible to change some of the configuration parameters of the PDA using the Control Center, like the amount of data that the PDA must keep in history.

The Control Center was developed in such a way, that it is possible to create different graphic user interfaces without having to change any code related with the communication with the PDA. The fact that the barcodes and RFID codes can be automatically associated with a location is also relevant.

**TECHNOLOGIES USED**

**CDMA2000**

CDMA2000 is a radio interface, approved by the ITU-T, under the IMT-2000 program, that supports 3G services.

The Portuguese coverage of the CDMA2000 network is assured by Zapp/Radiomóvel, providing the 1xRTT and 1xEV-DO systems, both already in Rev. A. The CDMA2000 1xRTT system, supports voice services as well as packet data services with peak data rates of 153Kbps in Rel 0 307 Kbps in Rev A, (upload or download).

The CDMA2000 1xEV-DO (Evolution-Data Optimized) system supports peak download data rates of 2.4 Mbps in Rel 0 and 3.1 Mbps in Rev A, and peak upload data rates of 153.6 Kbps in Rel 0 and 1.8 Mbps in Rev A.

The CDMA2000 terminals available support only Rel 0.
The CDMA2000 system used in Portugal operates in the 450MHz band, which is the reason why it is commonly named CDMA450. The main advantage of the CDMA450 system is the low setup cost, because a CDMA450 base station offers almost the same coverage as 25 WCDMA (UMTS) base stations, working at the 2.1 GHz band.

Table 1 shows a comparison between the coverage of the IMT-2000 systems at various frequency ranges.

Table 1 – Coverage of the IMT-2000 systems at various frequency ranges

<table>
<thead>
<tr>
<th>Frequency (MHz)</th>
<th>Cell Radius (km)</th>
<th>Cell Area (km²)</th>
<th>Relative number of Cells 450MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>450</td>
<td>48.9</td>
<td>7521</td>
<td>1</td>
</tr>
<tr>
<td>850</td>
<td>29.4</td>
<td>2712</td>
<td>2.8</td>
</tr>
<tr>
<td>950</td>
<td>26.9</td>
<td>2269</td>
<td>3.3</td>
</tr>
<tr>
<td>1800</td>
<td>14.0</td>
<td>618</td>
<td>12.2</td>
</tr>
<tr>
<td>1900</td>
<td>13.3</td>
<td>553</td>
<td>13.6</td>
</tr>
<tr>
<td>2100</td>
<td>10.0</td>
<td>312</td>
<td>24.1</td>
</tr>
</tbody>
</table>

Source: Qualcomm ITU 8/F Submission, June 11, 2001, “Coverage comparison of IMT-2000 systems at various frequency ranges, including 450 MHz”

GPS

GPS (Global Positioning System) is a navigation system based on satellite communications, made by a constellation of 24 satellites. The satellites orbit the Earth twice in less than 24 hours, in precise orbits, broadcasting information continuously. This information is received by GPS receivers, which compare the instant when the signal was emitted with the instant the signal was received, allowing the receiver to calculate the distance to the satellite. The receiver can calculate its geographical location using triangulation. To obtain a 2D location, at least 3 satellites must be in line of sight (LOS), and with another one, 3D information (altitude) is possible.

Once the location has been obtained, the receiver can calculate other information data, such as speed, direction, distance to destination, etc. The accuracy increases with the number of satellites in LOS.

Many of the current commercial receivers have the parallel multi-channel system that increases accuracy. The GPS system works in any meteorological conditions, in any place in the world, 24 hours a day.

To collect data from a GPS receiver, the most common protocol used is NMEA (National Marine Electronics Association), more precisely, NMEA-0183. In this protocol the data is sent in ASCII (American Standard Code for Information Interchange) format.

Bluetooth

Bluetooth is a short range wireless communication technology. It was designed to replace wires and supports voice and data communication. It is a low cost technology with low power requirements and high security level.

The devices using this technology can be connected in ad-hoc networks, commonly known as piconets. Each piconet may have 8 devices and each device can be connected to more than one piconet.

Bluetooth technology works in ISM (Industrial, Scientific and Medical) band between 2.4 and 2.485 MHz. The class 2 devices (the most used) transmit with a power of 2.5 mW and have a range of 10
meters. The 1.2 version supports data transfer speeds of 1Mbps and version 2 supports 3 Mbps.

**Barcode**

First of all, barcodes are a quick, precise and easy way of obtaining data. A barcode is composed by a numerical or alphanumerical code, represented by a group of black and white vertical stripes (the most common case, where the codes are 1D).

The barcodes are read by a small laser beam which reads the barcode from the right to the left. The barcode always start and ends with a white space.

There are many ways of representing a barcode. The most popular are UPC/EAN, Code 39, Code 128 and Interleaved 2 of 5.

**RFID**

RFID (Radio-Frequency IDentification) is a radio based Identification system. The greater advantage of this technology is of not requiring contact or line of sight with the reader. The “ID tags” are radio-frequency excited circuits that can be read through a lot of different substances (ice, ink or snow, for example) where it is impossible to use optical technologies, such as barcode.

The tags are manufactured in many sizes and forms, such as animal identification labels, labels with the form of credit cards, such as Lisboa Viva cards, in form of screw (used to identify trees), or of rigid/flexible plastic labels attached/glued to cloth, books, CD/DVD cases, to prevent theft in stores.

RFID systems can work in a wide range of frequencies. There are low frequency systems, working from 30 KHz to 500 KHz and high frequency systems, working from 850 MHz to 960 MHz and 2.4 GHz to 2.5GHz. The low frequency systems are cheaper but have less reading range and are almost only used in access control systems and animal identification systems. The high frequency systems are used to monitor trains and automatic payment at tollbooths (like Via Verde in Portugal).

![Fig. 3 – RFID tag for animal identification](image)

**System Architecture**

The hardware used in this project is composed by a PDA, model M3 from Mobile Compia, a RFID reader, model PCR125 from Promag/Gigatek, a GPS receiver, model GPS-BT100 from Asus, and a PC connected to the Internet.

![Fig. 4 – System Architecture](image)

The PDA is connected to the Internet using the CDMA2000 network and communicates with the Control Centre using TCP/IP.

The GPS communicates with the PDA using the Bluetooth interface, and sends data in the NMEA format to the PDA. The GPS processes that data and sends, periodically, the relevant information to the server, discarding the rest.
The RFID reader is connected to the PDA using the CompactFlash interface. The RFID reader is always trying to collect data, and when that happens, the PDA tries to associate that data with the last location received and sends it to the Control Centre.

Some data records, of the last received, from all the data sources, are recorded in the PDA memory. The amount of data recorded is configurable.

When the Control Centre receives data from the PDA, it adds it to a database (SQL Server 2005 Express). All the data received by the server can be looked up in tables, accessible by pressing some buttons in the graphical user interface. Every data that has an associated location can be visualized in Google Earth.

**Communication Protocol**

The communication protocol is very flexible, to enable the possibility to add features to the system very easily. All the messages are text based in UTF-8 format, according to the recommendation in RFC 2277 of the IETF (Internet Engineering Task Force).

All the lines must end with a carriage return linefeed ("\r\n").

Before being possible to exchange data between the PDA and the Control Centre, the PDA needs to successfully login. To login, the PDA sends a login message to the Control Centre: "LOGIN <UserId> <PdaId>\r\n". The Control Centre answers with a "LOGIN_OK" or "LOGIN_FAILED".

The protocol was inspired in XML. All the messages are delimited by an identifier, according to the source of the data sent in the message.

The base structure of the messages is:

```
MESSAGE_TYPE
  Message ...
  Message ...
...
/MESSAGE_TYPE
```
For example:

```
INFO
ClientInfo Bruno 949346528
/INFO
```

The message types are INFO, GPS, BARCODE and RFID. With this type of structure, if anyone wants to add a new type of data source to the system, the only thing required is a new message type. For example, to take advantage of the PDA’s embedded camera, the only thing needed is to add a new message type, like CAMER.

The tables below show the name of all the messages of each message type.

```
<table>
<thead>
<tr>
<th>Table 2 – Message type “BARCODE”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Centre → PDA</td>
</tr>
<tr>
<td>HistorySizeConfig</td>
</tr>
<tr>
<td>LastDataReq</td>
</tr>
<tr>
<td>HistoryInfoReq</td>
</tr>
</tbody>
</table>

Table 3 – Message type “RFID”

<table>
<thead>
<tr>
<th>Control Centre → PDA</th>
<th>PDA → Control Centre</th>
</tr>
</thead>
<tbody>
<tr>
<td>HistorySizeConfig</td>
<td>Data</td>
</tr>
<tr>
<td>LastDataReq</td>
<td>HistorySize</td>
</tr>
<tr>
<td>HistoryInfoReq</td>
<td></td>
</tr>
</tbody>
</table>

Table 4 – Message type “INFO”

<table>
<thead>
<tr>
<th>Control Centre → PDA</th>
<th>PDA → Control Centre</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefreshRateConfig</td>
<td>LocationInfo</td>
</tr>
<tr>
<td>LocationInfoReq</td>
<td>RefreshRate</td>
</tr>
<tr>
<td>HistoryInfoReq</td>
<td></td>
</tr>
</tbody>
</table>

Table 5 – Message type “GPS”

**M3 CLIENT**

The software developed for the PDA is called M3 Client. It was developed in C#, using the .Net Compact Framework 1.

The M3 Client was divided in several classes. The main classes are MainForm, RFIDReader, TcpConnection, GpsProcessing and BarcodeReader. There are also the classes GpsInfo, Checkpoint and Form_About.

The class TcpConnection manages the connection, reconnecting automatically when the connection is lost, and is used to transfer data, to and from the Control Centre.
The classes GpsProcessing, BarcodeReader and RFIDReader handle the connection and reception of data from their respective data sources.

The class MainForm is the application’s start up class, containing the Main(). It provides the graphical user interface as well as event processing for all the other classes, including some heavy tasks like the decoding and processing of the messages sent by the Control Centre.

The classes GpsInfo and Checkpoint are used as containers for the data received from the GPS, RFID and barcode readers.

The M3 Client shows a variety of valuable information to the user, like the status of all the data sources, status of the connection to the Control Centre, maximum amount of data of each type that the PDA saves for history purposes, current amount of information saved, last data collected by every data source, etc. It also saves the main configuration parameters in a XML file when the application exits. There are also some mechanisms of failure prevention, like automatic reconnect to the Control Centre or automatic deactivation and reactivation of the RFID reader, when it is removed or attached to the PDA. The function that decodes the data received from the Control Centre is also resilient against a large variety of syntax errors of the received messages.

The M3_serverBase.dll is the core of the Control Centre. It has all the non graphical functionality of the Control Centre, including everything related with the communication with the PDA and access to the Database.

Its main classes are M3_ServerBase, Logger, DataBaseConnectionManager and ArrayListExt. There are also the classes connectionInfo and CheckpointInfo.

The class CheckpointInfo is just a container for the barcode and RFID data received by the Control Centre. The class connectionInfo is used to save the information about the PDA that connects to the Control Centre (one object of this class for each PDA).

The class ArrayListExt is used for the list of PDAs connected to the Control Centre.

The DataBaseConnectionManager manages all the accesses to the database. It is very robust against errors, reconnecting automatically to the database when the connection is lost and repeating the queries when they fail.

The M3_ServerBase is the main class of this module. It contains the list of the PDA connected to the Control Centre (using the ArrayListExt and connectionInfo classes), handles the communication with all the PDAs, including the decoding and processing of all the messages sent by the PDAs and inserts the received data in the database (using the DataBaseConnectionManager).

It is possible to use this class to build Control Centers, with different graphical user interfaces.
**M3 Server**

The M3 Server is the Control Centre. It provides the graphical user interface to the M3_ServerBase.dll.

The main classes are ServerForm and DataShowForm. There are also the classes DataFormValidCheckpoints, DataFormUsers and Form_About.

The classes DataFormValidCheckpoints and DataFormUsers allow the user to view and change the list of registered PDA users and valid barcodes and RFID codes, accessing the database directly.

The class DataShowForm creates a window that allows the user to visualize the data received by the Control Centre, in a table.

The class ServerForm is the main window of the graphical user interface of the Control Centre. It initializes the M3_serverbase and calls all the other forms when needed. This class allows the user to send commands and information requests to the PDA, and also visualize the data received. The data received can be visualized in tables, using the DataShowForm. Every data that has location information can be visualized in Google Earth, and the last location received can be visualized in http://maps.google.com.

Like in the M3 Client, the M3 Server saves some configurations in a XML file. The function that reads these values also has some error correction mechanisms, generating missing files in some occasions, and also tries to auto detect the location of the Google Earth executable.

**ConsoleServer**

The ConsoleServer was developed to demonstrate how easy it is to make a Control Centre, using the M3_ServerBase.dll. It does not allow any interactivity but it supports everything regarding user login, data reception, decoding of messages, database access and creation of log files.

Although it may seem useless, the ConsoleServer proved to be very useful during the performance and tuning tests.

**Performance and Tuning Tests**

The most difficult things to process during the normal use of the system are the history requests. When the size of the history is very large, the processing of these requests can become very heavy.

There are two ways of sending the data: all at once, or line by line. To test the best approach, a series of tests were conducted, measuring the time spent by the PDA to send all the data, using different sizes for the reception buffer of the Control Centre. The results can be seen in Table 6.

<table>
<thead>
<tr>
<th>Buffer size [bytes]</th>
<th>Time [s] (all at once)</th>
<th>Time [s] (line by line)</th>
</tr>
</thead>
<tbody>
<tr>
<td>256 ((2^8))</td>
<td>47</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>47</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>54</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>46</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>56</td>
<td>18</td>
</tr>
<tr>
<td>2048 ((2^{11}))</td>
<td>46</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>47</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>47</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>53</td>
<td>22</td>
</tr>
<tr>
<td>65536 ((2^{16}))</td>
<td>49</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>44</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>52</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>51</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>48</td>
<td>22</td>
</tr>
</tbody>
</table>

Table 6 – Send history: all at once vs line by line

The results indicate that it is better to send data line by line, and so, this was the method adopted for use in all the history requests. The results also indicate that
the size of the reception buffer does not affect the time spent by the PDA to process and send the data.

During this test, a different behaviour in CPU usage of the Control Centre was detected, depending on the size of the reception buffer. A test was made to measure the CPU time spent by the Control Centre to receive and process the same data, using different sizes for the reception buffer. In this test the Control Centre used was the ConsoleServer, instead of the M3 Server. This decision was made because it is easier to modify the Console Server to send periodic history requests.

After analyzing the results shown in graphic 1, the value of 2048 bytes was chosen for the size of the Control Centre’s reception buffer.

![CPU Time vs Buffer Size Graph](image)

**Graphic 1 – Relation between buffer size and CPU Time**

**CONCLUSIONS**

It was possible to develop a system that possesses all of the initially requested features, such as, collecting location data from a GPS receiver, remote configuration of some of the PDA parameters, PDA identification, etc. There were also implemented several features not included in the initial proposal, such as collecting data from an RFIDReader and from a barcode reader with the PDA, or visualizing location data using Google Earth, in the Control Center.

In terms of future development, it is possible to increase the features of the system very easily, due to the flexibility of the communication protocol and the variety of communication interfaces of the PDA, allowing the use of new data sources, or taking advantage of the modularity of the Control Center, as well as developing new graphical interfaces for the Control Centre, using the M3_ServerBase.dll.

**REFERENCES**

[22] http://msdn.microsoft.com (Documentação sobre C#)