An architecture for telemetry–based systems to control remote devices via GPRS

Jim Jones[1], Vanilson Burégio[2], Gabor Sipkoi[3].
Unibratec, Recife/PE, Brazil

[1]jimjones@ieee.org, [2]vanilson.buregio@unibratec.edu.br, [3]gabor.sipkoi@unibratec.edu.br

Abstract: *With the advent of the fast technology advance, the interaction between software and hardware has become an useful approach to solve problems in several environments, such as industrial, commercial or domestic. Its application ranges from systems that control ordinary processes and events to more complexity high-tech security solutions. In parallel with this advent, the necessity of remote management has been exponentially increasing the field of telemetry systems. In practice, many solutions have been designed regarding aspects of coverage availability, reliability and costs. However, these solutions have a high complexity level, are too much specific and most of times very expensive, which discourages developers and sponsors. Under such motivation, this paper presents a customizable, low cost approach for designing telemetry-based systems to control remote devices via GPRS (General Radio Packet Service). Additionally, we describe our implementation experience (using the proposed architecture) that was successfully applied in two real contexts: (i) the management of critical events from a 69KV electrical substation and (ii) the control of an industrial x-ray machine.*

Keywords: automation, telemetry, GPRS, remote monitoring

1. Introduction

Building telemetry systems always require much effort because of their complexity and hard design. Such high complexity almost always discourages developers who need to implement this kind of systems. On the other hand, the high cost of developing telemetry systems is also another point that makes companies chose other "more conventional" ways to solve their issues.

The main intention of this paper is demystify the supposed difficulties of telemetry systems development and show an easy way to create solutions to monitor devices with a low cost, avoiding the necessity of using any proprietary technology, for example [8] Borland C++ Builder, Microsoft Windows 2000 or [9] LabVIEW, Microsoft Visual C++. In order to achieve these goals, this paper presents an architecture that does not specify the use of any high technology or even computer language, due its design, the implementation can be made using several kinds of technologies regarding the main aspects of a system conception.

The remainder of this paper is organized as follows. Section 2 shows some implementations of telemetry systems in many kinds of environments, such as emergency medicine, electrical controls and robotics. Section 3 we present the proposed architecture, detailing and explaining its modules and workflows. Section 4 presents two successful implementation experiences using the proposed architecture, which resulted in reliable and useful solutions. With the aim of achieving a low cost development, each system module was conceived using free technologies, such as database, high-level and low-level programming languages. Finally in the Section 5 we make a conclusion of this paper and show some further works.

2. Background

In this Section we present some telemetry based projects in distinct fields, to emphasize the importance and of this kind of solution in many different scenarios. In many kind of environments and fields the necessity of remote controlled systems has grown very fast and there are already many scenarios where this kind of systems are essential for a good management, such as hydro-electrical stations or water supply monitoring. Several kinds of solutions [1-6, 8-10] had been already designed to solve problems where the information needs to be reported instantly.

In [2] a wireless controlled robot, called MICROBOT was developed to carry out emulations and activities in different environments, where they can get data and materials in dangerous places or just for alarm controls in security zones. It consists in a mobile wireless controlled device that can be accessed by a mobile phone via GPRS (General Package Radio Service).

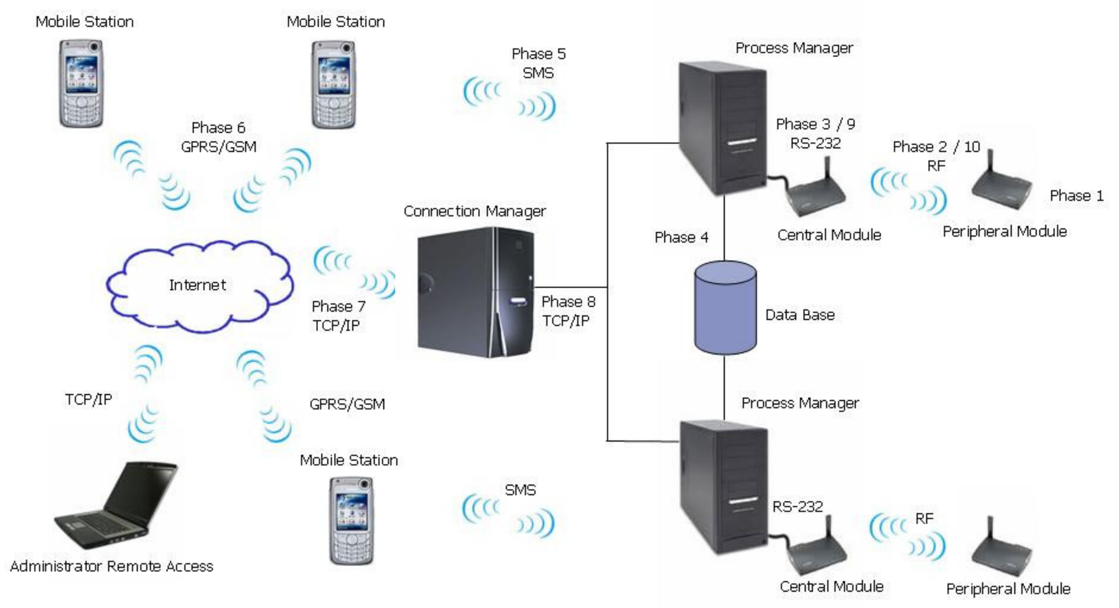
An example of the importance of telemetry-based systems can also be checked in [1], where a M2M (Machine to Machine) project has been developed in the emergency medicine field. This solution consists in a remote module that collects data from medical machines (for instance, defibrillators), in ambulances, helicopters or just home cared patients in the city of Karlsruhe and sends them to the German Red Cross in Bruchsal. All information is transmitted via GSM to a server where they will be stored and evaluated. Another examples of the telemetry-based systems approach in the medicine field is a solution [3, 4] for cardiac patients monitoring, where a device is placed in the patient and, as soon as this device detects the heart beats, it invokes the mobile phone through a RS-232 serial cable, and then establish a GSM connection through public mobile communication network to a Modem Server (TGMS) at the hospital automatically. Still on the medicine filed, was developed a [8] wearable health care system, where an intelligent device was created aimed at providing remote monitoring of the patient vital signs, such as breathing pattern, pressure, temperature, activity sensors and electrocardiogram. All these information are delivered to mobile phones, PCs, PDAs or by a specific designed device called WEATHY BOX. A similar system was developed in [9] to control monitor heart rate, three-lead electrocardiography and SpO_2. The data is collected and transferred to a PDA that forwards them to a server in the hospital via WLAN, where the data will be evaluated and stored by a system called Management Unit.

In [10] was created a multi-sensor system for water quality monitoring, where the device was organized in two blocks: The "Web-sensor measurement block" that consists in a module to collect the data from the sensors and the "Web-sensor communication block" that is responsible to take the collected data and make accessible for local wired (Ethernet cables) or wireless connections (GSM).

3. Proposed Architecture

A general definition of the proposed architecture is presented in this Section. The overall goal is to satisfy the set of drawbacks in existing solutions presented in Section 2, providing a unified vision of what the system looks like and how its internal elements are combined in order to provide users with a customizable telemetry-based system that supports the control of any kind of remote devices using the GPRS technology.

Figure 1 summarizes a general definition of the proposed architecture decomposed in six modules: (i) *Peripheral Module, (ii) Central Module, (iii)Connection Manager, (iv) Process Manager, (v) Mobile Station (Mobile phone) and (vi) Data Base.* An overview of the main phases is shown on Table 1, detailing them for modules, communication technologies and actions.



[Figure 1 - The Proposed Architecture]

[Table 1 - The main phases of the architecture]

Phase	Action
1	The monitored device triggers an event
2	The Peripheral Module sends to the Central Module the new event. (via Radio frequency)

3	The Central Module forwards to the Process Manager the new event. (via Serial RS-232)
4	The Process Manager selects the involved operators and logs the event. (via Data base query)
5	The Process Manager sends a SMS Message for each operator.
6	The Mobile Station acknowledges the incoming SMS message.
7	The Mobile Station establishes a socket connection with the Connection Manager. (via GPRS)
8	The Connection Manager forwards the incoming connection to the right Process Manager. (via Socket)
9	The Process Manager forwards the action to the Central Module. (via Serial RS-232)
10	The Central Module sends to the Peripheral Module, that will execute the action. (via Radio Frequency)

These modules form the basic structure that work in conjunction to provide the necessary functionality. The following sections describe the details of each module

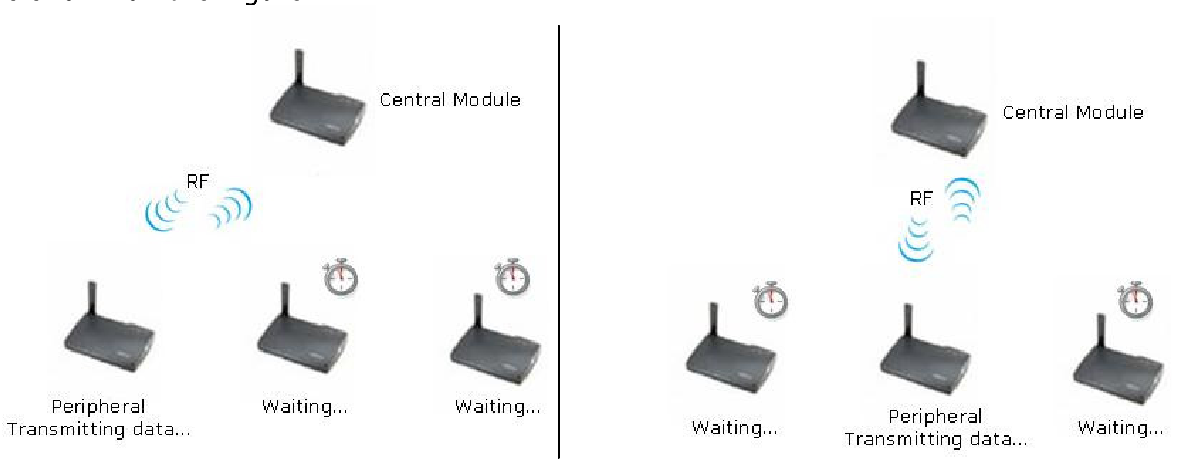
3.1 Peripheral Module

The *Peripheral Module* is connected to the device that should be controlled. This module will get the events from the monitored device and send to the *Central Module* by RF (*Radio Frequency*) [6, 7]. The module is also able to receive messages from *Central Module* and based on these messages take appropriate actions on the attached module. It makes a buffer with all events that may have occurred and waits until the next time the *Central Module* will asks if there is a new event to be reported.

This module was conceived in a way to facilitate the connection with a variety of sensors allowing it to monitor a great variety of events. This module also has the ability to control external devices including high power equipment.

3.2 Central Module

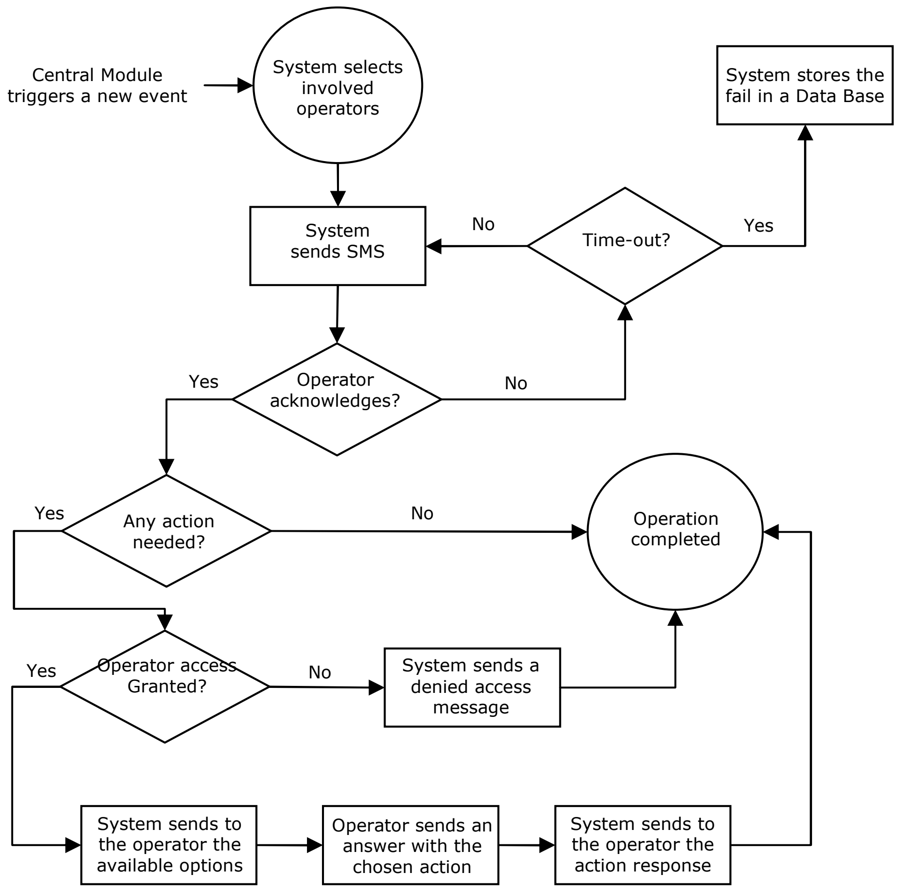
The *Central Module* represents a device connected to a computer *(Process Manager)* by RS-232 and also connected to the *Peripheral Module* by RF. This module will take all events coming from the *Peripheral Module* and forward it to the *Process Manager* by serial connection (RS-232). The *Central Module* Pools each *Peripheral Module* around asking if an event has occurred since last pooling cycle, as shown on the Figure 2.



[Figure 2 – Pooling of the Peripheral Modules by the Central Module]

3.3 Process Manager

This system is responsible to validate all system's business rules, manage all information coming from the *Central Module* and also the connections coming from the *Mobile Station.* As shown in the Figure 3, this system will also manage data of operators, devices, events, messages and actions, storing them in a Data Base.



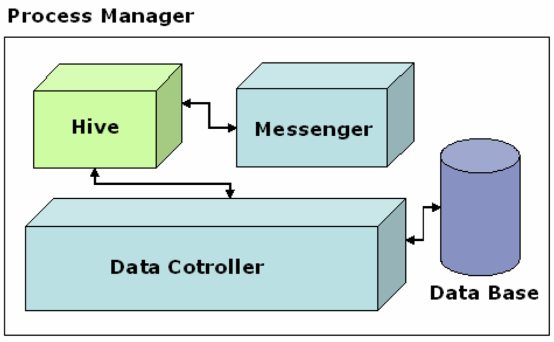
Central Module triggers a new event
System selects involved operators
System stores the fail in a Data Base
System sends SMS
No
Time-out?
Yes
Yes
Operator acknowledges?
No
Yes
Any action needed?
No
Operation completed
Yes
Operator access Granted?
No
System sends a denied access message
System sends to the operator the available options
Operator sends an answer with the chosen action
System sends to the operator the action response

[Figure 3 – Process Manager Workflow diagram (New event from the Central Module)]

Based on the message that came from the *Central Module* the *Process Manager* will select all operators involved on that certain event and send a SMS Message to each one, where the *Mobile Station* will start up and acknowledge the message. Each message will have its own id and they will be marked as pendent until the operator acknowledgement or when it gives a time-out.

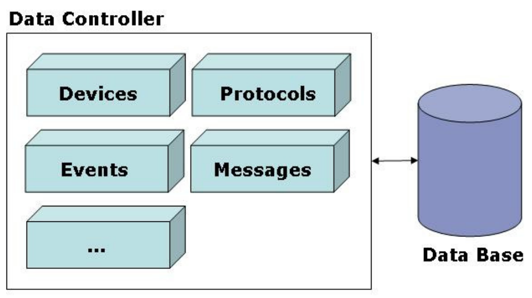
3.3.1 Inside the Process Manager

The *Process Manager* is divided in four layers, the *Hive*, the *Data Controller*, the *Messenger* and the *Data Base*, as you can check in the Figure 4.



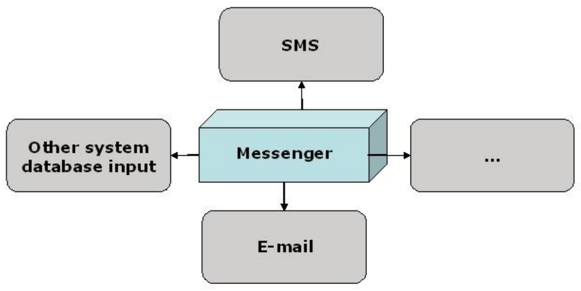
[Figure 4 – Process Manager Architecture]

- Data Controller: This Process Manager layer controls all data storage, for instance, verifying data integrity, operators authentication, control event logs, helping the Hive to validate system actions. All input and output data will come through this layer. Figure 5



[Figure 5 – Inside the Data Controller]

– Messenger: This layer is responsible to alert the operators about new events. It will receive the list of the operators which are supposed to be notified, process these data, selects the chosen deliver way (For instance: Short Message Service (SMS), E-mail, notify another system by inputting data in its database or just by triggering its specific functionality) and notify the operator. Figure 6.

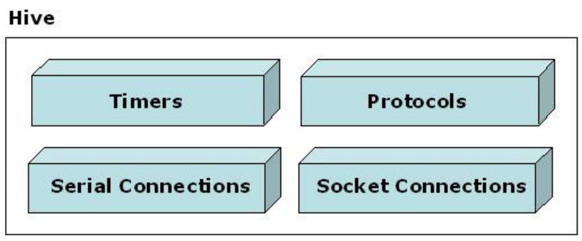


[Figure 6 – Inside the Messenger]

- The Hive: This layer is the most important one on the Process Manager. It represents a bridge between the Mobile Station and the Central Module, validating protocols, checking packages integrity, notifying the operators and managing events. Each event is controlled by a specific thread, so every new event generates in run time a new timer to controls its execution and status. A Hive process can be verified in the Figure 7 and in the following step-by-step event example:

The following example is a simple event coming from the *Peripheral Module*

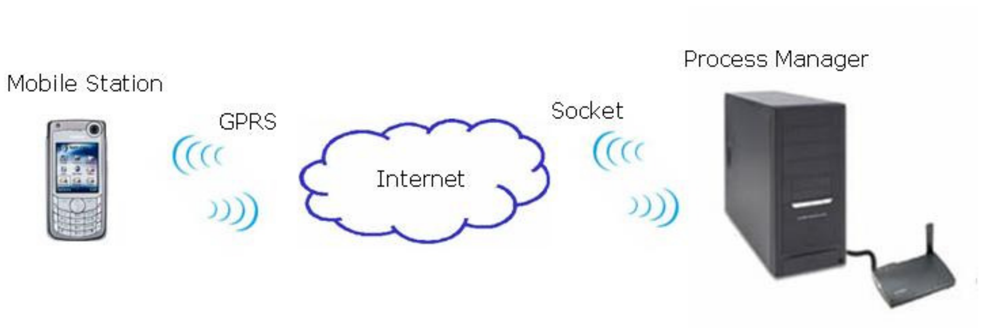
1. Serial Connection unit gets a new event;
2. Timer unit creates a new thread;
3. Protocols unit verify the message integrity;
4. Thread requests to the Data Controller the involved operators;
5. The Messenger notifies all involved operators
   (Thread keeps running as pendent, waiting for an operator response);
6. Socket Connections unit gets the Mobile Station acknowledgement;
7. Timer forward the message to the Data Controller to notify the acknowledgment;
8. Thread killed, operation completed.



[Figure 7 – Inside the Hive]

3.4. Mobile Station

The *Mobile Station* is an embedded system that runs in a cell phone. This system will let the operator knows what is going on with the remote device and also gives it a response making a socket connection to the *Process Manager* by GPRS, which means the operator will be able to manage devices in every place in the world that has GPRS coverage, as you can check in the Figure 8. For example: turn off / on a machine or disable a specific functionality.



[Figure 8 – data transferring between Process Manager and Mobile Station]

3.5. Connection Manager

This module was conceived to cover the necessity of a scenario that requires more than one *Process Manager.* This system works like a virtual bridge that manages connections between the *Mobile Station* and *the Process Manager.* It gets the *Mobile Station's* connections, verifies where the operator is supposed to work and forwards the connections to the right *Process Manager.* It can be plugged and unplugged at anytime, because both of them (*Process Manager* and *Connection Manager*) has the same algorithm to manage *Mobile Station* connections. In a scenario where the process manager is required, it also authenticates the operator's credentials before forwards the connection, this way the bond will be instantly established when the Process Manager receives the connection with the previously authentication.

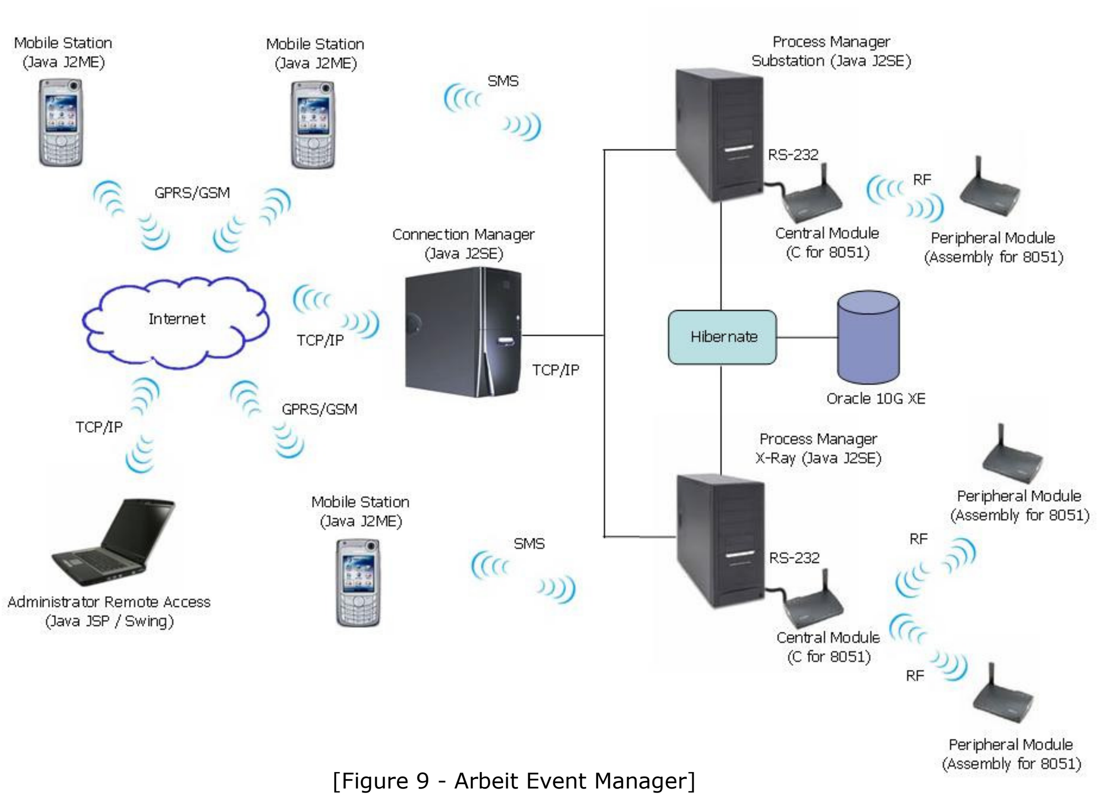
Note: This module is only necessary if your scenario REQUIRES more than one *Process Manager* as you can check above in the Figure 1, otherwise *Process Manager* itself can controls the incoming connections.

4. **Implementation experience**

In this section we present an implementation experience of the proposed architecture in a 69KV electrical substation and an industrial x-ray machine. This project (called *Arbeit Event Manager*) was developed using the following technologies to implement the elements defined in Section 3:

- *Peripheral Module*: implemented using an 8051 based microcontroller that was programmed in assembly;
- *Central Module*: implemented using an 8051 based microcontroller that was programmed in C
- *Mobile Station*: Java J2ME;
- *Process Manager*: Java J2SE;
- *Connection Manager*: Java J2SE;
- *Database*: Oracle 10G XE (free version)
- *Frameworks:* Hibernate and Struts.
- *User interfaces:* JSP and Swing

Due its scenario, the *Arbeit Event Manager* required a *Connection Manager*, since the controlled devices are several kilometers from each other.


Mobile Station (Java J2ME)
Mobile Station (Java J2ME)
SMS
Process Manager Substation (Java J2SE)
RS-232
RF
GPRS/GSM
Central Module (C for 8051)
Peripheral Module (Assembly for 8051)
Connection Manager (Java J2SE)
Internet
TCP/IP
Hibernate
Oracle 10G XE
TCP/IP
GPRS/GSM
TCP/IP
Process Manager X-Ray (Java J2SE)
Peripheral Module (Assembly for 8051)
Mobile Station (Java J2ME)
SMS
RF
RS-232
Administrator Remote Access (Java JSP / Swing)
Central Module (C for 8051)
RF
Peripheral Module (Assembly for 8051)

[Figure 9 - Arbeit Event Manager]

69KV Substation monitoring

The architecture proposed in this paper was implemented to develop a system to notify the operators from a glass factory about some events in a 69KV substation. The *Peripheral Module* is connected to an energy monitor that will monitor power variations in a predefined power range and detects any power loss in one or more voltage lines. Any variation detected by the energy monitor will be informed to the *Peripheral Module* and a message will be sent to the *Central Module* that will forward the message to the *Process Manager*, and then notify the operators in the *Mobile Station*. This way the operator will be warned of substation abnormal events in real time, make reports and charts using the *Administrator Remote Access*, to analyze the recent events and take preventive actions, to avoid any eventual damage.

Industrial X-ray monitoring

This solution was conceived to interact to an industrial x-ray machine remotely. A total failure of the refrigerating system of this machine can be fatal. The refrigerating system can generate several status signals that are monitored by the *Peripheral Module*. Massages are sent to the *Central Module* that will forward the message to the *Process Manager*, and then notify the operators in the *Mobile Station*, but, differently of the solution above, in this implementation the operator will be able to operate the refrigerating system remotely. For instance, if the solution detects an abnormal temperature because of a failure in the cooling system, the operator can turn-it off remotely until the technical personnel solves the problem. This will avoid possible damages to an expensive and critical system.

5. Conclusions and further works

Due the actual technologic scenario, telemetry systems are everyday more and more useful, for instance: important actions that need to be executed in anytime from anywhere, remote machines management in dangerous or inaccessible environments, residential remote security systems, and so on. With this advent an avalanche of innovative solutions are developed everyday, but most of them, although useful, are still quite expensive, which makes the companies look for other kind of ways to solver their problems.

In this paper, we present an easy architecture to develop telemetry-based systems with a low cost, regarding the aspects of performance, coverage and reliability. Beyond that, we also present an implementation experience which resulted in a reliable system to monitor events from a 69KV electrical substation and an industrial x-ray machine.

As future work, we are using the present architecture to build a Java framework to support the development of multi-platform systems that require communication with machines and remote devices, such as mobile phones, PDAs or Pocket PCs.

Acknowledgments

The author gratefully would like to thank Kayro Féllyx H. Pereira and Reginaldo Soares Júnior for their intense efforts and contributions on the electronic and low-level programming development, without them this work would not be possible. The author also thanks Prof. Andre Fowler Annes and Prof. Dr. Adalberto Cajueiro for their invaluable and essential advices.

References

[1] Bogdan Ciubotaru-Petrescu; Dan Chiciudean; Razvan Cioarga; Daniela Stanescu “Wireless Solutions for Telemetry in Civil Equipment and Infrastructure Monitoring” - University of Timisoara, Timisoara, Romania.

[2] Diego López de Ipiña; Iñaki Vázquez; Jonathan Ruiz de Garibay; David Sainz “GPRS-based Real-Time Remote Control of MicroBots with M2M Capabilities” - University of Deusto, Bilbao, Spain.

[3] J. Schöchlin; T. Bachmor1; D. Teufel; K. Bickel; A. Bolz1; “KATER – KARLSRUHE TELEMETRY TEST IN RESCUE SERVICES” - Universität Karlsruhe, Karlsruhe, Germany

[4] Y Jasemian, E Toft, L Arendt-Nielsen; “Real-time remote monitoring cardiac patients at distance” - Aalborg University, Aalborg, Denmark

[5] Jouko E. Pakanen, Dr.; Kai Hakkarainen, Eng.; Kari Karhukorpi, Eng.; Petri Jokela, MSc.; Timo Peltola, MSc, Jyrki Sundström, Eng. “ A LOW-COST INTERNET CONNECTION FOR INTELLIGENT APPLIANCES OF BUILDINGS” PUBLISHED: July 2002 at http://www.itcon.org/2002/3/

[6] Michael John Brady; Thomas Cofino; Harley Kent Heinrich; Glen Walden Johnson: Paul Andrew Moskowitz: George Frederick Walker “Radio Frequency identification tag” Patent number: 5682143

[7] Telemetry definition, Wikipedia, http://en.wikipedia.org/wiki/Telemetry

[8] R. Paradiso; MEIOR SpA and SMARTEX s.r.l., Prato, Italy “WEARABLE HEALTH CARE SYSTEM FOR VITAL SIGNS”: 4th Annual IEEE Conf on Information Technology Applications in Biomediane, UK.

[9] Yuan-Hsiang Lin; I-Chien Jan; Patrick Chow-In Ko; Yen-Yu Chen; Jau-Min Wong; and Gwo-Jen Jan; “A Wireless PDA-Based Physiological Monitoring System for Patient Transport: IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE, VOL. 8, NO. 4, DECEMBER 2004

[10] O. Postolache; P. Girao; M. Pereira; H. Ramos; “An internet and Microcontroller-Based Remote Operation Multi-Sensor System for Water Quality Monitoring”: Sensors, 2002. Proceedings of IEEE Publication Date: 12-14 June 2002, Instrumentation and Measurement, IEEE Transactions on page(s): 322- 329, Volume: 54, Issue: 1, Feb. 2005