GPRS Based System for Atmospheric Pollution Monitoring and Warning

Suciu Constantin, Florin Moldoveanu, Radu Campeanu
Ioana Baciu, Sorin Mihai Grigorescu, Bogdan Carstea, Vlad Voinea

Transilvania University of Brasov, Faculty of Electrical Engineering and Computer Science,
29, Eroilor Blvd., RO-500036 Brasov, Romania
Tel., Fax: +40-0268-474718, e-mail: moldof@marconi.unitbv.ro

***Abstract* – This paper presents the synthesis of a SCADA (*Supervisory Control And Data Acquisition*) system, named *Pollution Guard*, designed to collect and process atmospheric pollution data measured in several strategic points of a region. Pollution Guard makes use of the GPRS (*General Packet Radio Service*) data communication infrastructure from a mobile communication provider that covers a very large area, practically the air pollution data being collected from every place in the country. In comparison to other similar systems, the new functionalities provided by *Pollution Guard* are the SMS (*Short Messaging System*) and e-mail alerts generated when the level of toxic substances exceeds some given values, chosen with regard to respiratory illness.**

***Index Terms* – Data acquisition, analog-to-digital conversion, microcontroller, wireless, GPRS, SMS, Java.**

I. Introduction

It is very important to be informed about the pollution levels anytime, especially when they exceed certain limits that are considered unsafe or that are stipulated by local laws [1]. The *Pollution Guard* system is designed to provide up-to-date information about pollution levels from specific locations and send warnings whenever the composition of the air becomes dangerous [2]. The gaseous composition of unpolluted air is considered to be: nitrogen – 756,500 ppm (parts per million) (vol.), oxygen – 202,900 ppm, water – 31,200 ppm, argon – 9,000 ppm, carbon dioxide – 305 ppm, neon – 17.4 ppm, helium – 5.0 ppm, methane – 0.97 ppm, krypton – 0.97 ppm, nitrous oxide – 0.49 ppm, hydrogen – 0.49 ppm, xenon – 0.08 ppm and organic vapors ca. 0.02 ppm. The major air pollutants are carbon monoxide (CO), carbon dioxide (CO_2), ozone, nitrogen oxide (NO), lead and sulfur dioxide (SO_2) [3].

A. Benefits

When health is a major concern of our daily life, the Pollution Guard project becomes a very useful tool that offers:

- up-to-date information about the composition of air, based on user location;
- an easy to use interface, available to users everywhere;
- pollution warnings via SMS and email for registered users;
- an organized way to view information about the evolution of pollution in certain locations;
- a cheap and robust system for air pollution monitoring.

The system's functionality can be extended furthermore by creating personalized alerts for users that have an increased sensibility to a certain substance.

B. Innovation

Compared to other similar projects, *Pollution Guard* comes with several innovations. Perhaps the most important innovation is the possibility of alerting users by means of mobile communication (SMS – *Short Messaging System* and E-mails), offering this way a real-time protection to pollution hazards, anywhere, anytime. The users can also specify their preferences, in case they are sensitive to a certain substance, and thus receive customized alerts.

Another innovation point is the acquirement of data from the carbon monoxide, carbon dioxide, ozone and temperature sensors once an hour, providing therefore realistic information to the system. The data about the air composition will be thus updated and automatically displayed on the website for each location where a sensor panel is placed.

The data communication is made through the GSM network. The system is easy to configure and knows if certain measurements devices don't work. This is done by implementing a new type of communication protocol that takes advantage of the HTTP (*Hypertext Protocol*) protocol.

II. Principle of Operation

The model used for the synthesis of the *Pollution Guard* system is the client-server model, the clients being materialized by the PDU (*Pollution Detection Unit*) devices and the server by the central server which collects and stores the atmospheric data transmitted by the PDU devices [11]. The system's functionality is represented in Fig. 1.

A. The measurement process

The heart of the PDU is the PIC16F877A microcontroller [7]. Using the built ADC (*Analog to Digital Converter*) the values provided by the sensitive elements (gas sensors) are stored inside the internal RAM (*Random Access Memory*) memory of the microcontroller. The actual measurement process of the substances present in the atmosphere takes place at a specific time interval m_i (2 min). The data transfer to the central server is made according to another time interval t_i which

1-4244-0361-8/06/$20.00 ©2006 IEEE

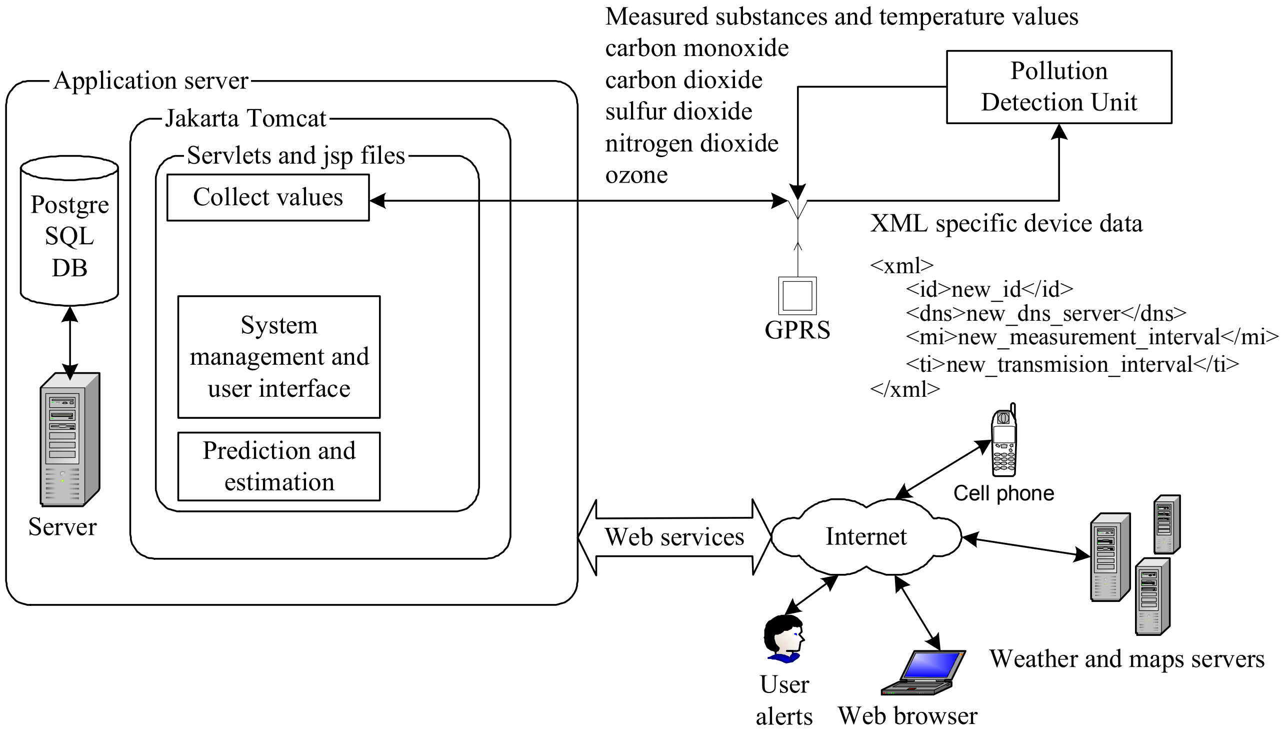

Fig. 1. *Pollution Guard* system architecture.

can be set with regard to the variation of the monitored substances (in our case 1h). The t_i interval was calculated according to the next factors:

- the power needed by the system for a GPRS data transmission [6];
- the cost of the GPRS data transmission.

The OP (*Operating System*) design for the PIC16F877A microcontroller will send to the server the mean value of the measurement made between two given time intervals t_i (for our initial configuration the mean value of the 6 measurements made during one hour). After this, the internal RAM memory allocated for the intermediate storage of the substances data is freed, thus making room for the next set of measurements, their mean value being sent to the database server at time t_{i+1}. If the GSM network is not reachable the data will not be erased from the memory. New measurement data will be stored until the internal and external memory will be full. This data will be sent to the server when the GSM network will be reached again.

B. The environmental warning level

If a value of one of the substances presented in the air exceeds a certain threshold value, then the GPRS data transmission takes place immediately. When this data will arrive at the central server it will be processed and if it is really an environmental problem then the server will automatically broadcast emergency SMS messages and e-mails to the subscribers of the *Pollution Guard* system.

The concentration value of toxic substances present in the air at a given moment of time is calculated using an AQI (*Air Quality Index*):

$$\mathrm{AQI} = \frac{\textit{pollutant concentration}}{\textit{pollutant nominal concentration}} \times 100 .$$

This index is calculated only on the server side and displayed in automatically generated reports and charts. The AQI is shown on the pollution map using a color code:

- BLUE – very good ($\mathrm{AQI} = [0 \ldots 33]$);
- GREEN – good ($\mathrm{AQI} = [34 \ldots 66]$);
- PURPLE – fair ($\mathrm{AQI} = [67 \ldots 99]$);
- RED – poor ($\mathrm{AQI} = [100 \ldots 149]$);
- GRAY – very poor ($\mathrm{AQI} > 150$).

C. Communication protocol

For the correct functioning of the system a PDU device contains inside his EPROM (*Erasable Programmable Read Only Memory*) memory some system variables like a unique identifier, which corresponds to a primary key from the table that holds the information about the existing PDU devices, the DNS address of the central server or the m_i and t_i intervals.

The measured data is sent to the central server using the GPRS data communication and the HTTP protocol.

Because the PDU does not have an IP (*Internet Protocol*) address, the configuration of such a unit is made after the data transmission takes place. The data is sent to the server by accessing the available POST or GET requests methods using the HTTP protocol. If the PDU needs to be reconfigured then the server will return to the PDU an ASCII (*American Standard*

Code for Information Interchange) string in XML (*eXtended Markup Language*) format which will be interpreted by the OP of the microcontroller. The string format is:

```
<xml>
    <id>new_id</id>
    <dns>new_dns_server</dns>
    <mi>new_measurement_interval</mi>
    <ti>new_transmision_interval</ti>
</xml>
```

This XML file will be returned to the PDU only in the case of a reconfiguration of the device.

The *Pollution Guard* system is able to monitor the toxic pollutants of an isolated region where the GSM network can't be reached (national parks, etc.). The PDU devices can be optionally equipped with GPS (*Global Positioning System*) receivers, the values of the measurements and the coordinates of the measurement place being stored inside the external memory accessed by the PIC16F877A microcontroller. The m_i interval and the time when the PDU can make measurements depend on the amount of external memory, a bigger memory providing to the device a longer period of functioning. If there are many PDU devices needed for monitoring an isolated region and the cost of the GPS receivers is too big, then the GPS coordinates can be stored inside the database when the system is configured. When the PDU devices are taken inside the area where the GSM network can be reached the data stored in the external memory is sent to the central server.

D. System management

The management of the *Pollution Guard* system is made using a web interface provided by the central server, thus the system may be configured and upgraded from any place in the world only by the use of an internet connection and a web browser.

Also, the user can register for air pollution warning on a certain region of interest, his way to work or his home for example. The main function of the management interface is to manage the users logged into the system and the registered PDU devices.

III. Hardware Description

A PDU was conceived in such a way that the price/quality proportion gives the best value and also the power consumption is as small as possible, regarding the fact that the mobile device will be powered by 4.5V batteries.

The PIC16F877A microcontroller is the main component of a pollution detection unit [4]. The operating system that runs inside the chip coordinates the substances measurement process, the acquisition of the GPS coordinates and the data transmission to the central server [5]. The microcontroller is mounted on a development board that provides an RS232 serial communication to the *Telit* GM862 GPRS modem and GPS receiver and a parallel connection to the gas sensors [8] - [10].

Because the voltages required by the PIC16F877A microcontroller (5V) and the *Telit* GM862 GPRS modem (3.8V) are different, the RS232 connection was made using two chips dedicated for converting RS232 voltages:

- MAX232 – for converting the 5V signal from the microcontroller into 12V standard RS232 signal;
- MAX3232 – for converting the 2.8V RS232 signal from the *Telit* modem into 12V standard RS232 signal (in spite that the *Telit* GM862 modem requires a 3.8V power source, the input and output signals are propagated with a maximum voltage of 2.8V).

For the measurement process the next gas sensors are used:

- carbon monoxide: TGS2442 (Figaro);
- carbon dioxide: TGS4160 (Figaro);
- nitrogen dioxide: NO23050 (Sensoric);
- sulfur dioxide: SO2BF (Sensoric);
- ozone: O33E1 (Sensoric).

The connection between the gas sensors and the PIC16F877A microcontroller can't be made directly because of the very small output voltages provided by the sensors (CO_2 sensor: mA) or because the measurement unit is not voltage/ppm (*Particles per Million*) (CO sensor: resistance/ppm). This problem is solved by using auxiliary electronic circuits for signal conversion like OA (*Operational Amplifiers*) and transistors.

A. CO and CO_2 sensors

The TGS4160 [11] is a hybrid sensor unit composed of a CO_2 sensitive element and a thermistor. A wide range of 350 ÷ 30,000 ppm of carbon dioxide can be detected by TGS4160. The CO_2 sensitive element consists of a solid electrolyte formed between two electrodes, together with a printed heater (P_t) substrate. By monitoring the change in electromotive force (EMF) generated between the two electrodes, it is possible to measure the CO_2 gas concentration. Adsorbent is filled between the internal cover and the outer cover for the purpose of reducing the influence of interference gases. Using an operational amplifier the CO_2 sensor can be read with an electronic circuit like the one presented in Fig. 2.

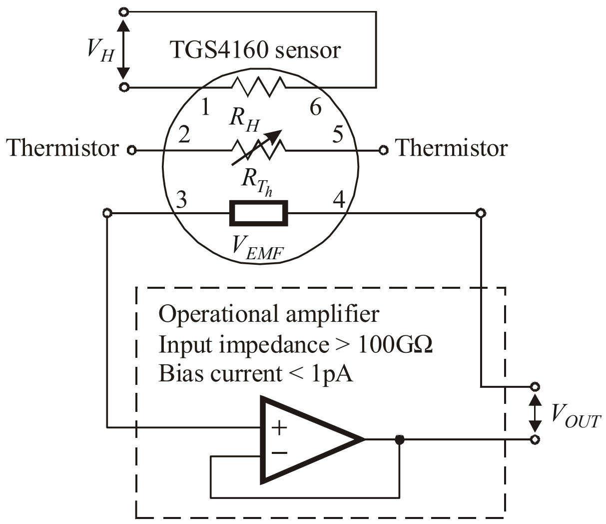

Fig. 2. CO_2 measurement using an additional operational amplifier.

The TGS4160 sensor requires heater voltage (V_H) input. The heater voltage is applied to the integrated heater in order to maintain the sensing element at a specific temperature which is optimal for sensing. Since the solid electrolyte type sensor functions as a kind of battery, the EMF value itself would drift using this basic measuring circuit. However, the change of EMF value (EMF) shows a stable relationship with the change of CO_2 concentration.

TGS2442 [12] utilizes a multilayer sensor structure. A glass layer for thermal insulation is printed between a ruthenium oxide heater and an alumina substrate. The gas-sensing layer, which is formed of a tin dioxide (S_nO_2), is printed on an electrical insulation layer that covers the heater. Circuit voltage (V_C) is applied across the sensing element that has a resistance (R_S) between the sensor's two electrodes (pins 2 and 3) and a load resistor (R_L) connected in series. The sensing element is heated by the heater which is connected to pins 1 and 4. Heating cycle: the sensor requires application of a 1 second heating cycle which is used in connection with a circuit voltage cycle of 1 second. Each V_H cycle is comprised by a 4.8V being applied to the heater for the first 14 ms, followed by 0V pulse for the remaining 986 ms. The V_C cycle consists of 0V applied for 995 ms, followed by 5.0V for 5 ms. For achieving optimal sensing characteristics, the sensor's signal should be measured after the midpoint of the 5ms V_C pulse of 5.0V. In Fig. 3 the time diagram for optimal reading of the TGS2442 sensor is represented. The external electronic circuit is shown in Fig. 4.

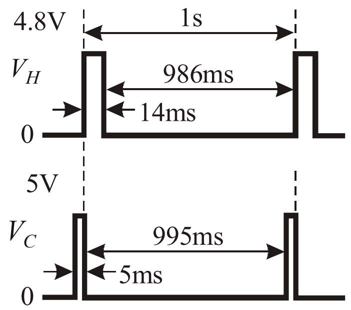

Fig 3. The TGS2442 reading diagram.

B. Measurement and conversion of air substances

The Microchip PICmicro MCU mid-range devices have relatively good analog I/O capabilities. The ADC built in the PIC16F877A can sample and process signals as fast as 25 kHz accurately. With up to 12 bits available with built-in ADCs, this type of microcontroller is very well suited for accurately measuring DC (*Direct Current*) analog voltage.

The PORTA pins can be used as either digital I/O or as analog inputs. The actual bit accuracy and operating speed is a function of the clock speed at witch the PIC micro MCU runs, in our case 4 MHz. The pins configured for reading the sensors output signal are modeled like the electronic circuit shown in the Fig. 5. R_S from the sensor output voltage circuit is the in-line resistance of the power supply. The time for the holding capacitor to load the analog voltage and to stabilize is:

$$T_{ack} = 5ms + [(temp - 25C)\cdot 0.05ms/C] + \\ +(3.19C\cdot 10^7)\cdot(8K + R_s) \tag{3.1}$$

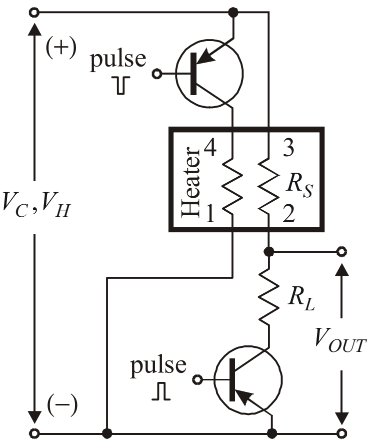

Fig. 4. External electronic circuit for TGS2442 sensor reading.

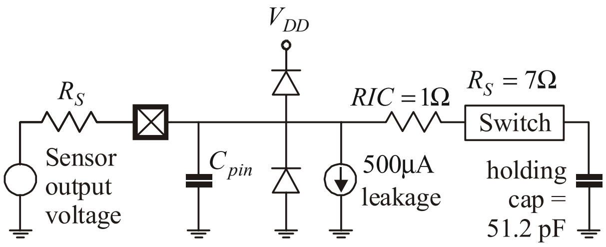

Fig. 5. The reading of a gas sensor using the PIC16F877A analog-to-digital converter.

To measure the analog voltage sensors values, the analog input pins of the PIC16F877A, which are in port A, have to be set to analog input on power up; the analog input pins are normally set to analog input and not digital I/O. To specify the modes, the ADCON1 register is written to. Table 1 shows the two least-significant bits (known as PCFG1:PCFG0) of the ADCON1 register with the types of I/O pin operation selected in the PIC16F877A microcontroller.

TABLE I
ADCON1 BIT DEFINITIONS FOR THE PIC16F877A

ADCON1 bits	AN3	AN2	AN1	AN0
11	D	D	D	D
10	D	D	A	A
01	V_{ref}	A	A	A
00	A	A	A	A

The ADCON0 register is used to control the operation of the ADC. The data is stored in the two ADRESH and ADRESL registers in two different formats. The first format is to store the data right justified with the most-significant six bits of ADRESH loaded with zero and the least two significant bits loaded with the two most-significant bits of the result. This format is useful since the result is going to be used as a 16-bit number, with all the bits used to calculate an average. The second 10-bit ADC result format is left justified, in which the eight most-significant bits are stored in ADRESH. For implementing the analog-to-digital conversion of the sensors output voltage, the following steps are taken:

1. Write to ADCON1 indicating which are the digital I/O pins and which are the analog I/O pins;
2. Write to ADCON0, setting ADON, resetting ADIF and GO/_DONE, and specify the 4 MHz TAD clock;
3. Wait for the input signal to stabilize;
4. Set the GO/_DONE bit;
5. Pool GO/_DONE until it is reset (conversion done);
6. Read the result from ADRES and ADRESH.

IV. SOFTWARE DESCRIPTION

The *Pollution Guard* software can be broken into two separate modules:

- the central server software;
- the PDU operating system.

A. Central server software

The code of the server application is written in the Java language. The web server is accessed from the Internet using the *Apache Tomcat Software* [16]. *Apache Tomcat* has native support for the JSP-Servlet technology and to function, it needs the installation of a JDK (*Java Development Kit*) distribution [14].

The server code can be separated into three subsystems:

- the data acquisition and storage subsystems;
- the user alert subsystem;
- the user interface and administration subsystem.

The data acquisition process is implemented using the HTTP protocol. The transmission of measurement data from the PDU devices will trigger the execution of the `Insert-Values` servlet. This servlet is responsible with the verification of data integrity and if the data comes from a valid PDU unit. If the received data is valid then the servlet will execute the procedure for inserting the data in the database.

For data storage we are using the *PostgreSQL* database server [13]. The data access is made through a set of JDBC (*Java Database Connectivity*) functions, the interrogation being made using the SQL (*Structured Query Language*) language. In the JDBC context the system architecture is represented in Fig. 6. The classes for implementing the database access are defined in the package `net.pollution.base.database`. Every table is read through the use of classes derived from the package mentioned above.

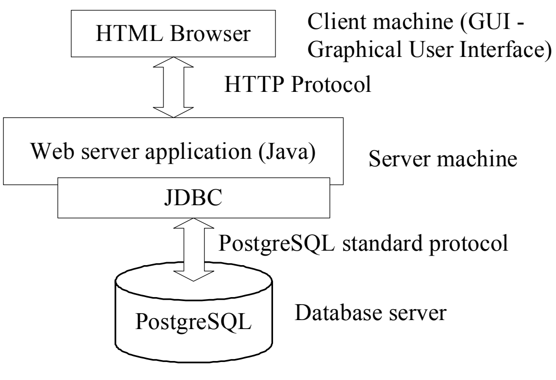

Fig. 6. Three-tier architecture.

The user alert subsystem is linked to the data acquisition subsystem. Before data storage the measured values are compared to a threshold value. If this value is bigger than the threshold value a connection to a GPRS modem is made and a series of SMS and e-mail alert messages are sent to the *Pollution Guard* subscribers and also a series of e-mail messages. For sending SMS messages a *Telit* GM862 GPRS modem was used.

The user interface and administration subsystem are accessible through a web browser. The code of the interface is written using the JSP-Servlet technology. In figure 7 a snapshot of the user interface representing the distribution of atmospheric pollution over a period of time is presented.

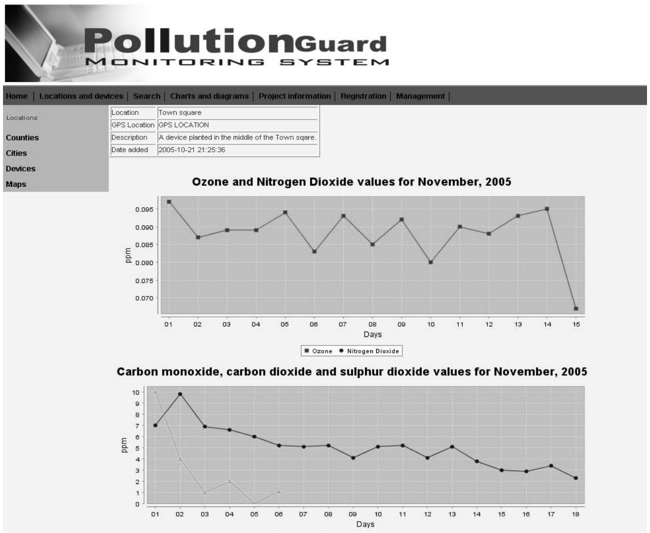

Fig. 7. User interface snapshot.

B. PDU operating system

The flow chart describing the functioning of the OS running inside the PIC16F877A microcontroller is presented in Fig. 8.

For the measured data to be transfered to the server, the *Pollution Guard* system makes use of the HTTP protocol. The configuration data needed for the connection is:

- The DNS (*Domain Name Server*) of the server to be contacted;
- The application level protocol: HTTP 1.0 (RFC1945 – *Request For Comment*) [15].

Using the standard AT command set, mapped into the internal RAM of the PIC16F877A microcontroller, the ASCII strings passed to the *Telit* GM862 GPRS modem are:

```
AT+CGDCONT=1,”IP”,“internet.gprs”,”0.0.0
.0”,0,0<cr>
AT#USERID=”EASY GPRS”<cr>
AT#PASSW=”EASY GPRS”<cr>
AT#SKTSET=0,80,82.78.144.155/servlet/Ins
ertVaues?id=2154&o=3.2&cm=2.1&cd=0.1&sd=
3.1&nd=2.4&t=29><measured_data<cr>
```

V. CONCLUSIONS

Human activities generate emissions of pollutants that are changing the chemical composition of the Earth’s atmosphere.

Observation of how the planet system is responding to this anthropogenic perturbation is the key to understanding the consequences of both natural and human induced modification.

The *Pollution Guard* system is able to alert people with health problems caused by atmospheric pollution and to collect data at a configurable time interval. The measured substances are made public on the Internet, this way the concentration levels of toxic substances being monitored over the time.

A picture of the PDU device can be seen in Fig. 9.

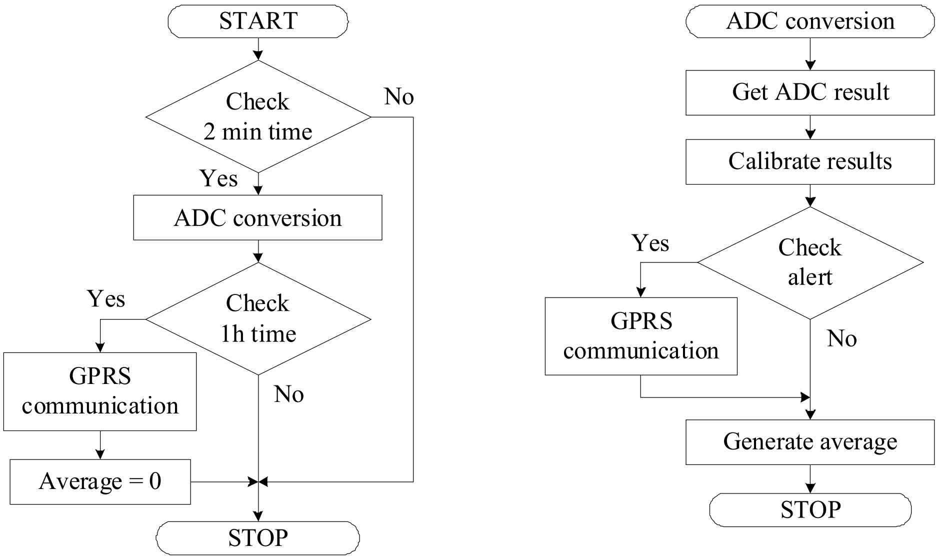

Fig. 8.Timer interrupt for sensors reading (analog-to-digital conversion) and GPRS data transfer.

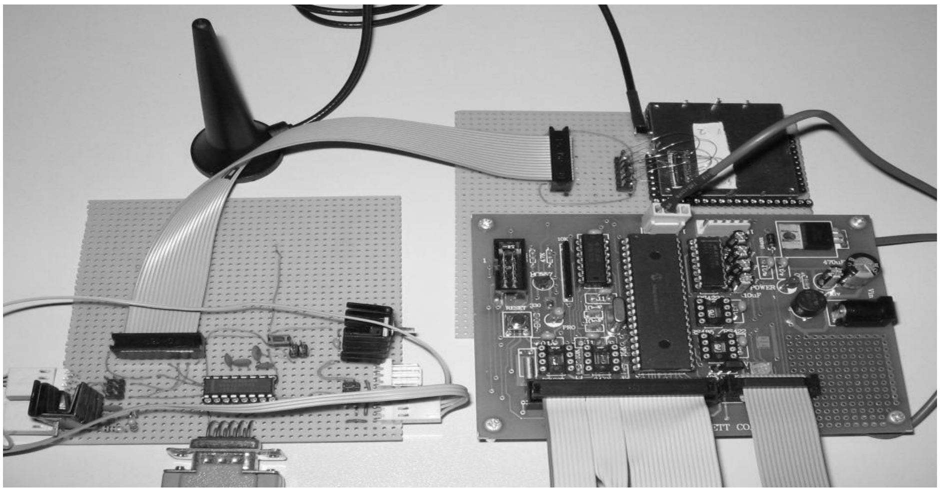

Fig. 9. PDU implementation.

REFERENCES

[1] *** "Health Aspects of Air Pollution with Particulate Matter, Ozone and Nitrogen Dioxide", Report on a WHO Working Group, Bonn, Germany, 2003.

[2] R. Gorham, "Air Pollution from Ground Transportation – an Assessment of Causes, Strategies and Tactics, and Proposed Actions for the International Community", United Nations, 2002.

[3] *** "A Survey of Monitoring Instruments for Measurement of Airborne Pollutants" New York State Energy Research And Development Authority, *Nyserda*, October 2002.

[4] M. Predko, *Programming and Customizing PICmicro Microcontrollers*, *Mc Graw Hill*, 2002.

[5] P.N. Borza, C. Gerigan, P. Ogruțan, Ghe. Toacșe, *Microcontrollers – Applications* (in Romanian), *Ed. Tehnică*, Bucuresti, Romania, 2000.

[6] F. Sandu, D.N. Robu, P.N. Borza, W. Szabo, "Twin Microcontroller – GSM Modem Development System", in *Proc. of the 8th Inter. Conf. on Optimization of Electrical and Electronic Equipments – OPTIM 2000*, Brasov, Romania, 2002, pp. 701-704.

[7] *** Microchip PIC16F877 Datasheet.

[8] *** Microchip MPLAB Development Software, version 7.20, www.microchip.com.

[9] *** Telit GM862 GPRS/GSM modem, "Software User Guide".

[10] K. B. Mayer, "Over the Air Programmable Stand-alone Controllers Using GPRS Technology", Part fulfillment thesis of the degree of Bachelor, The Australian National University, June, 2002.

[11] *** "Technical Information for TGS4160", Figaro Engineering Inc., Osaka, JAPAN.

[12] *** "Technical Information for TGS2442", Figaro Engineering Inc., Osaka, JAPAN.

[13] *** http://www.postgresql.org.

[14] *** http://www.java.sun.com.

[15] *** RFC1945 – Request For Comment

[16] *** http://jakarta.apache.org/tomcat.