HOST - hybrid optoelectronic versatile telemetric system for local community

Ryszard Romaniuk, Krzysztof Poźniak, Rafał Salański
Institute of Electronic Systems
Warsaw University of Technology
R.Romaniuk@spie.pl

ABSTRACT

The work describes hybrid telemetric system built from cheap and standardized measurement stations equipped in cheap, off-shelf optoelectronic and classical sensors, provided with Internet connections and localized in local community administration and service centers. Cheap metrological server, for such a telemetric network, was designed. A number of such servers were manufactured and installed as a fully functional network. The design included: functional structure of the server, measurements algorithms, software and hardware layers, in particular effective connections for large number of diversified sensors. The description presented below is devoted to a particular model of the network realized practically in a few local community sites around the city of Warsaw, including the Town of Zielonka and city center. The realized functional solutions for the hybrid optoelectronic network were debated in details. The debated system functionalities embrace three layers: functions, hardware and software, and include: choice of measured parameters, measurement methods, data acquisition, access to processed data via the Internet, kind of applied sensors and methods of sensor signal transmission, kinds of applied microprocessor chips, types of PC chips, functional structure, realized data, system programming environment, telecommunication and telemetric procedures, system tests, etc. Usage of optoelectronics in the HOST system was emphasized.

Keywords: optoelectronic telemetry, sensors, environmental measurements, optoelectronic telemetric networks

1. INTRODUCTION

The aim of the work was to research a systematic approach to the distributed, environmental, universal, optoelectronics based, telemetric network relaying on a cheap and versatile solution of telemetric, Internet compatible, server. A number of intermediate tasks were solved. Several trial measurement stations were designed, realized and practically implemented. The system consisted of: environmental measurement station using Internet connection, standardized measurement interface (hardware and software), packet of control software, and packet of visualization software for measuring results on the Internet. The following assumptions were taken: standard market, off-shelf, cheap sensors of medium accuracies will be used; the role of the sensors will be to monitor roughly some parameters and discover potential risks; software for measuring station and whole network system will be done for MSW as this SO is popular in Local Community Administration and management Centers (CAC); computer fulfilling the role of a measuring station will be connected to the Internet; a separate machine will be devoted to the role of metrological data base WWW server; all involved computers will be used during the working hours of CAC by clerks to perform their daily duties; all the software and hardware of the debated system should neither slow down nor prevent, in any way, these standard CAC duties.

One of the authors of this work (R.S.) cooperated with the authorities of Zielonka Community (ZC) for the last several months. This co-operation concerned building of an advanced WWW portal of the ZC. Needs analysis showed the necessity to include local extent environment protection and metrological layers to this portal. Local Community is a fundamental and the smallest administration unity. The area of average country LC embraces up to teens closely located towns and villages. One of the tasks of LC is to monitor all environmental risks (floods, fires, industrial, etc) and to inform local inhabitants and relevant authorities of higher level in administration hierarchy. The LC is an administrative organ located the closest to the inhabitants. LC manages the most basic local problems. Localization of the central node of such distributed metrological network, spanning the whole area of LC, in the CAC is the most optimal solution. The
inhabitants may have immediate access to all important and detailed data from environmental measurements performed in the vicinity of their settlements. LC possesses constant possibility to observe the local situation on-line and record the data for off-line purposes. Later data processing and analysis is possible including: displaying average, maximal and minimal values of measurements, averaging over time and area, checking against hot spots, calculation of multi-parameter correlations, performing regressions, predictions, statistics etc. Network and particular stations may measure a number of different parameters. The choice of local measurements may depend on local requirements. The villages located over rivers may request measurements of local water level, intensity of stream flow, and water quality through opacity or chemical parameters. These parameters, through public access, may be exchanged between adjacent LC to map the flood risk and alarm situation over larger regions. Distribution of temperature, moisture (ground, brush-wood), precipitation, and other sensors in localities placed along transportation tracks and major routes allows for monitoring of local fire risks and traffic conditions. Popularization of the idea to build public network of LC measurement stations may lead to dense measurement network fit in the best way to local needs and simultaneously serving effectively the whole community.

Previous experiences of authors with single, Internet accessible, measurement stations were inspiration to build an extended network of LC metrological stations. These were localized in the city center and measured some key environmental parameters like meteorological, chemical and physical data, including: temperature, sun radiation, atmospheric pressure, noise level, wind meter, wind chilling factor, laser ceilometer (cloud base height), laser meter of air opacity and sight range determination, laser dustiness meter, level of local ionizing radiation, CO₂ and NOₓ levels, etc. These stations had the possibility to build measurement database, store, process and display on the WWW page. Some stations have worked for several years since 95 and gathered a lot of valuable local meteo and environmental/pollution data. These data may be now used for medium term estimations of environment evolution. Some of these metrological servers get great interest among public Internet users like http://pergx.ise.pw.edu.pl (x=1,2,3…). Some data were used practically for numerable purposes like: data correlation investigations, Poisson and Gauss statistics on real measurement data, calibration of barometers and height meters. Some of the parameters were simply more current than given in radio broadcasts, what induced a lot of kindness among internauts.

2. IDEA, DESIGN AND MODEL OF LC HOST METROLOGICAL NETWORK

Construction of the system bases on a very simple idea of distributed environmental metrological network possessing universal metrological servers situated in the CAC. The following justification was assumed to implement such a network on a large scale:

- More than 2000 LC in nation’s scale creates potentially a large number of measuring points,
- The distribution of potential measuring points is quite homogeneous, regionally as well as globally,
- There is ready technical background for the implementation of such a network – CAC are fully computerized because of fulfilled functions. CAC possess Internet connections. Most of these connections are, however, POTS modem ones. More and more CAC install dedicated links or ISDN modems.
- There are both side advantages. The data are gathered by the open measuring station and are, from assumption, public. They are used for LC purposes, from one side. They are also used as local data in a larger system spanning the country, from the second side.
- Targeted groups of recipients; Data from particular measuring points arrive immediately to local target recipient, which are inhabitants of this LC. They also are available to any interested person in the state of local environment, like tourists, scientists, professional and public environment protection organizations, transportation, etc.
- The presentation of data may be done on the LC WWW pages. Localization of data associates them with the place of measurement. It gets easier to find such data from particular region of the country. The distribution of such measurement and data visualization services, among numerable local databases, lowers the Internet traffic asymmetry.
- Taking advantage of the administrative division of the country may facilitate initial phase of the project. The next tier in the administrative hierarchy above commune is district. Each district has on the average up to 10 communes. The districts may also build metrological centers relevant to their territorial extent. That gives additionally around 300 next measurement points/centers located usually in bigger towns. Also other configurations are possible for free regions, voivodeships, a group of neighboring districts, etc.
The design of HOST has to take into account local conditions like:

- Particular communes are nondependent unities. There is the necessity to negotiate the implementation conditions individually for each case.
- There are different technical conditions in each LC like:
  - Computerization of CAC was done in different periods of time. Not all systems are upgraded sufficiently frequently. Thus, technical parameters of available computers, which may be potentially used for metrological stations, may considerably differ from case to case.
  - It is assumed that that the computer, with installed software for metrological station, may be used, in parallel for different purposes, by the CAC clerks.
  - There is no homogeneous method to access the Internet in CAC. All available methods on the market exist in CAC, like: leased line with constant IP, radio-links with IP, connection on demand through ISDN with dynamic IP, dial-up connection via ISP with dynamic IP, DHCP, etc.

### 2.1. Metrological station

The following requirements may be defined for a designed metrological station, taking into account the above-mentioned conditions:

- The lowest possible costs,
- Zero effort installation; The installation process of hardware and software should be confined to minimum,
- Independence from computer configuration; The software should work in the same way on all kinds of computer, old and new alike and should not depend on the kind of processor, system clock, and quantity of operational memory,
- Connection to the Internet; The station software should be able to send data independently on the kind of existing locally Internet connection. It should possess the ability to connect and disconnect automatically with Internet via analog and digital modems. Connection costs should be taken into account.
- Measurement data transmission; Due to a variety of existing links, the software should be able to measure time of data transmission to WWW server and central database, offer and manage several ways of data transmission like: sending after individual measurement, grouping data and sending in blocks after preset period of time, at certain time during off-hours for clerks, off-line work, etc.
- Local data presentation – software modules should allow to update efficiently the web pages presenting the latest measurement data.
- The measurement accuracy should be in the range of 1-10%, depending on the cost and performance of used sensors.

The most straightforward solution of measurement station construction is to equip such a station with a simple, cheap and multi-port standard A/C, ISA/PCI/USB, and internal measurement board. This is, however, against the assumption of extreme ease of installation. Board installation requires computer opening. Sometimes this may be difficult, as some of the PC manufacturers do not allow case opening by unauthorized staff without losing guarantee privileges. The idea of the work presented here is to omit any difficulties in simple proliferation of such stations.

Fulfilling of the task requires splitting of the station into three separate modules. First one is responsible only for a single task of data acquisition from measuring sensors. This module works all the time independently of the PC. It is a separate µP board with own power supply and small package. The connection to computer is done through extended distance RS (or optical RS). Measurements are done even if the host computer is switched off.

### 2.2. Software

The second module of the station is software. Software functionality embraces communication with µP driver, data readout and processing. Software should present data in a readable way on a computer on which it resides. A fundamental assumption, allowing for solution mobility is separation of WWW server from the measuring station, i.e. a computer to which the µP board is connected. The CAC, which possess their own WWW site usually do not have their own server but use hosting services offered by specialized ISP. Unfortunately there is a number of different ways of Internet connections, differing in speed and quality.
2.3. Dat presentation

Third module of the system located outside CAC is responsible for data presentation on the web. The most popular systems for web servers are now NT2000, Linux and Unix. The presentation software should be nondependent on the server platform. The chosen application environments were: C or Perl applications performed in CGI-BIN, and PHP. C application must be compiled for each operating system, thus preventing the ease of installing. PHP is a new and modern technology, more effective than Perl. PHP is a scripting language processed by server. PHP modules may be installed on all OS. The data base access interface for PHP is simple and possesses functions of building graphical files.

System design assumes usage of popular sensors, including optoelectronic ones. Very sensitive components are accessible for such values as temperature, humidity, pressure, etc. More difficult are direct measurements of harmful gaseous substances in the air and fluid in water. The costs of precise sensors of such values are quite considerable. Destination of the station for monitoring purposes allows applying cheap sensors of much smaller sensitivity. Detection of an abrupt change of measured harmful factor may result in alarms for proper monitoring services, equipped in homologated equipment. The sensors may be grouped in special stations. Simple package ay help in this.

2.4. Model of Metrological CAC Station

Build model of metrological station fulfilled all above design assumptions. The measurement sensors were arranged in three groups: basic – measurements of basic weather parameters like: temperature, humidity, pressure, sun radiation, noise level; extended – precipitation, wind direction and strength, wind chilling factor, pollution of rain water like pH and alkaline metal ions; internal – temperature, humidity, gases like CO₂, methane, etc.

The µP board with own power supply is nondependent from computer. Sensor groups are attached directly to the µP board. The board works all the time, making readout, processing and acquisition of the data. The board may gather data even not connected to the computer.

The software of the station may work in two modes – monitoring with constant display of measured parameters and in invisible mode, minimally burdening the OS. The second mode allows for normal usage of the PC and simultaneous efficient operation of the measuring station. The software during the work makes data readout periodically from µP board, processes data and transforms to visual form. Transmission of data to web server is done according to preset mode in the user interface. It is possible to send data after each measurement, during preset time of day, and in preset volume. The installation and administration module is a separate application which is not accessible to PC user.

Measurement data are presented on web server in the form of Dhtml pages and diagrams. The pages and diagrams are generated using PHP technology.

3. INVESTIGATION OF MEASURING STATION MODEL IN ZIELONKA CAC

Formal cooperation was established with the authorities of Zielonka Commune and town, concerning the realization of this work. The result of this cooperation was installation of muster measuring station visible in the Internet. Town self government (SG) gave access to CAC equipment with ISDN link to the Internet. The Zielonka Commune SG financed part of the program.

The station was constructed according to model described in the previous chapter. The sensors consisted of three main groups. The sensors groups were placed in the room of CAC, behind the window of this room and on the roof of CAC building. Measurement data transmission is planned from a local river. All sensors are connected with the µP hardware driver. Data from the driver are read by software of measurement station. Internet connection is on demand ISDN. Zielonka web site is placed on external web server. The Zielonka CAC does not possess leased line. Data update is done at a top of every new hour. The station has been working now for nearly two years. Direct monitoring has proved that the measuring software in invisible mode uses nearly nothing of the host PC resources. This software does not do any harm to CAC PC regular duties. Testing character of the measuring station, however, required its switching off from
time to time. Experienced gathered during exploitation of this station are invaluable in construction of the whole measuring network of this kind. Below we describe some of its particular solutions.

3.1. µP hardware driver

The µP task is to read parameter data from the sensors. These parameters may be analog or digital. Analog parameters are converted to digital. Data from last measurements are stored in RAM. Data transmission to a PC is done on demand by the readout panel in the metrological station software. The µP driver uses 80552 Intel chip. The µP board is equipped with 14bit DAC, EPROM memory 62256 and keyed controller of step motors.

The driver possesses individual address and RS re-transmitter. This solution allows for serial connection of many drivers using a single COM port of the PC. The applied communication algorithm provides fast and errorless data exchange. PC, which wants to make readout from a particular µP driver, sends out its address through RS. The first driver compares the addresses. If they are identical, it sends back earlier prepared measurement data. If they are not identical it re-transmits PC request to the next µP driver. The algorithm is repeated until it reaches a proper driver. The address space is suitable for as many as 256 drivers. Driver parameters are as follows: readout of 8 analog input signals (0-5V) with 10bit resolution up to 20kHz, readout of 8 digital TTL signals, recording of 8 digital TTL signals, counter 16bit, time measurement with resolution of 1µs.

Data readout and signal formatting. Output signals of off-shelf commercial sensors use different standards. The most common formats are: analog voltage signal 0-5V, current signal 4-20mA and frequency. Often the output signals fulfill no standards at all. The described METEO station has the following signals. The following measurements use voltage readout proportional to measured value: temperature, wind chilling factor, sun radiation, atmospheric pressure, wind direction and intensity, CO, CO₂, methane and butane levels. The following values are measured by counting of pulses: the number of statics, precipitation, etc. Time measurements are used for determining of: relative moisture (duration time of output pulse from univibrator with capacitance sensor RH25), wind velocity (measurements of fan rotation time).

Data are read out, by the PC, from the µP driver, every minute. The counter measurements are set to zero after each readout and start counting again. Most of the mentioned parameters are slowly changing and readout every minute is sufficient and does not have essential influence on the measurement accuracy. There are some exceptions, however. One of this is wind velocity. µP calculates average, minimal and maximal values, prior to sending data to PC.
3.2. Sensors and their parameters

One of the advantages of the project is that it uses very cheap and popular sensors of standard accuracies. The sensors of pressure, temperature, moisture, etc, are not only cheap but quite accurate (better than 1%). Gaseous and aqueous measurements with similar accuracy require usage of expensive sensors, thus less accurate ones were used. This does not confine the functionality of the station, as its aim was to monitor the changes of some of such parameters. The procedures require generating of alarms after crossing preset threshold values. As it was mentioned earlier the sensors were grouped according to their functionalities:

- **Basic1** – temperature, pressure, humidity, sun radiation, atmospheric discharges – placed in a mushroom type package and resistant to external weather conditions;
- **Basic2** – wind intensity and direction, UV and ionizing radiation, rain meter, air transparency and visibility range, etc – placed in individual or partly integrated packages, also weather resistant;
- **Gases** – CO, CO₂, methane, butane, etc. – placed in measured areas in building.

Temperature measurements were done with LM35 sensor. It enables measurements in the range of -55°C to +150°C with the accuracy of ± 1.5%.
Rys. 2. Sensors block diagrams - temperature, pressure, relative humidity, sun radiation optical IR and UV, number and intensity of atmospheric discharges, gases, wind direction (with contactron connections), and rain meter.

The pressure was measured with linear, thermally compensated, silicone, Motorola sensor MPX4115AP. The parameters are, according to the manufacturer: maximal error 1.5% in range 0°-+85° C; thermal compensation is for the range -40° to +125° C; pressure range measurements 15 – 115 kPa; supply voltage $V_S = 4.85 – 5.35$ V DC; max current consumption $10mA$; Calibration function $V_{OUT} = V_S *(0.009 * P- 0.095) \pm 1.5\%$ (kPa).

Relative humidity was measured with linear, capacitive sensor of long term stability RH-25. The parameters, according to the manufacturer, are: measurement range 10-90% of relative humidity, the capacitance of the sensor at 55% of relative humidity is $C_S \approx 900$ pF, sensitivity of sensor is $S_S=1.8 \pm 0.4$ pF/1% of humidity change; linearity is $L_S<5\%$, work temperature range 0°-60° C.

The RH-25 sensor is a capacitance of univibrator evoking in the measurement circuit. After switching on the circuit from the μP it generates a pulse (time gate). During this time interval narrow clock pulses are counted from quartz μP generator (12MHz). Duration of the time gate is $\tau = 1.1RC$. 
Sun radiation level was measured by a photoresistor. The PR element, together with a resistor of 420Ω makes a resistive divider. The resistance of PR decreases with the intensity of incident light. As a result the analog output voltage signal increases and this increase is measured by µP.

UV radiation intensity was measured with a photodiode OSD58-7Q Centronic. It is sensitive in the spectral region 190-400nm. The sensitivities are for λ=190 nm S=0,12A/W, for λ=245nm S=0,14A/W, for λ=340 nm S=0,15A/W, thermal work range ∆T= -55°C- +70°C, according to the manufacturer data.

The atmospheric discharges are measured acoustically, optically and electro-magnetically. SEM is induced in a coil. The current pulse releases a univibrator, which in turn generates a standard 1s pulse counted by the µP.

Semiconductor oxidized sensors, with platinum micro-heater, were used to measure gases. The manufacturer was Scimarec. The parameters of the sensors were as follows: AF23 – choke-damp (CO) sensor, Rgas/Rair=0,20-0,50, Rgas=25-100kΩ, for 100ppm CO/air ratio; AF50 – methane sensor (CH₄) Rgas/Rair=0,20-0,45, Rgas= 3 kΩ - 12 kΩ, at 2000ppm CH₄/air. AF56 – butane sensor (CH₃H₁₀), Rgas/Rair= 0,08 - 0,25, Rgas= 1,5 kΩ - 6 kΩ, at 2000ppm CH₃H₁₀/air.

Rain meter was built in a classical way as a cone of 158mm in diameter (0,0196 m²). Water from the cone is carried to mechanical bistable device. It possesses two chambers A and B. One chamber gathers water. After crossing a weight/volume threshold, the two-chamber devices switches to other stable position and the second chamber starts to gather water. The water from the idle chamber spills out. The number of state changes is counted by a simple counter based on a contactron. The contactoron is switched by a magnet placed over the bi-chamber axis. A single pulse is equivalent to 0,3mm. A unity of precipitation is 1mm or 1litre/1m². The chamber switching threshold is 6ml (6000µl) with accuracy of ±5%. These values were assessed experimentally. The area of the cone is ,0196m², thus the calculations give value of 0,3mm. This is the smallest precipitation value measured by this device. Measurement accuracy may be increased by increasing area of the rain gathering cone.

Classical mechanical devices measure wind speed and direction. They bear magnets on their axes to switch contactrons. Wind intensity sensor uses only one contactron.

3.3. Sensors calibration

The µP device reads measurement data from the sensors. The data are then read out by the metrological station (PC) software. One of the functions of the software is to process/transform measurement data to real values. Application of linear sensors allows for their easy calibration by least square method. During the calibration process two values are read out in parallel, one from the investigated sensor and the second from the reference meter. The real value is a linear function of the value read by the sensor. Successive points presenting respective pairs of real and measured data makes a straight line y=ax+b. Setting the value of xᵢ from the sensor to the line function of the value read by the sensor. Application of software. One of the functions of the software is to process/transform measurement data to real values. A special software block was prepared to calibrate the used sensors with the method of the least squares. The sensors of temperature, humidity, pressure, gas concentration and sun radiation were calibrated with...
this method. The readout levels of Sun radiation sensor assume minimum 0% for cloudy night and maximum between noon and 13.00 during cloudless day.

It is necessary to take into account the ageing factor in cheap and simple sensors. The basic calibration process utilizes the least square method, which is valid for short time system operation. During medium and long time operation it is necessary to observe average monthly readouts. Taking into account the ageing factor requires addition of an ageing factor $s$ to the $b$ constant in straight-line equation $y=ax+b$ describing the best transformation between read (measured) and real values. Thus the equation is now $y=ax+b+s$, where $s$ is a difference between average readouts of successive months (weeks, etc).

![Fig. 3. Calibration of an ageing factor](image)

Wind direction is digitized. The respective value is read out, depending on which contactron is switched on. Setting a real data is performed through comparison of data read from the sensor and range of values for particular wind direction. The eight directions are designated as: 0-N, 1-NE, 3-E, 4-SE, 5-SW, 6-W, 7-NW. The calibration of wind strength is done by the least square method. To describe real values, Beaufort scale tables were used.

Sensing (felt) temperature is a parameter, which can be obtained on the basis of two other parameters: air temperature and wind speed. The temperature of human skin is 33°C, when the wind velocity is close to zero. The body heats the air in its vicinity, insulating itself by the warm air cushion from the surrounding. This mechanism leads to feeling higher temperature than it is in reality. Faster cooling of body surface appears with increase of wind velocity. The felt temperature decreases. The South Pole researchers P.A.Siple and C.F.Passel investigated wind-chilling factor. They investigated a way water freezes in plastic bag subject to intense wind chilling. Siple and Court gave the formulas describing felt temperature. Both formulas have the same algebraic form but different constants. The differences concern the definition of heat loss $H=(A+B\sqrt{V}+C*S)dT$, where $H[\text{kg cals/m}^2/\text{h}]$, $dT$-temperature difference in (°C) between skin and surrounding, $V$-wind velocity [m/s], $\sqrt{V} = \sqrt{V^2}$. A,B,C constants are $A \approx 10,45$Siple; $9,00$Court; $B \approx -10,00$S; $10,90$C; $-1,00$S; $-1,00$C. Practical formulas are: 1) Siple: $Twc=33+(Tc-33)(0,474+0,454\sqrt{V}-0,0454\sqrt{V})$ (this formula was assumed for calculations); Court: $Twc=33+(Tc-33)(0,550+0,417\sqrt{V}-0,0454\sqrt{V})$; $Twc$– felt temperature °C; $Tc$– ambient temperature °C; $V$ – wind velocity m/sek; (skin temperature 33°C at wind velocity 1,8 m/s); 2) Siple: $Twc=91,4+(Tf-91,4)(0,474+0,304\sqrt{V}-0,0203V)$; Court: $Twc=91,4+(Tf-91,4)(0,550+0,279\sqrt{V}-0,0203V)$; $Twc$– felt temperature °F; $Tf$ – ambient temperature °F; $V$ – wind velocity mil/h; (skin temperature 91.4°F at wind velocity 4 mil/h); 3) calculation formulas according to seaborne units; Siple: $Twc=91,4+(Tf-91,4)(0,474+0,326\sqrt{V}-0,0234V)$; Court: $Twc=91,4+(Tf-91,4)(0,550+0,299\sqrt{V}-0,0234V)$; $Twc$– felt temperature °F; $Tf$– ambient temperature °F; $V$ – wind velocity in knots (1 m/s = 0,869 knot); (skin temperature 91.4°F at wind velocity 3,47 knots); 4) Formula describing felt temperature (National Weather Service): $Twc=33+(Tc-33)(0,045(5,27\sqrt{V}+10,45-0,28V)$; $Twc$– felt T °C; $Tc$ – ambient T °C ; $Twc=91,4+(Tf-91,4)(0,0817(3,71\sqrt{V}+5,81-0,25V)$; $Twc$– T °F; $V$-wind velocity mil/h; $Tf$-T °F.

4. SOFTWARE LAYER DESCRIPTION

Dedicated software packet was prepared to work with the CAC metrological server. The software was prepared within the Borland Builder C++ for MSW9x/NT/2K. Three separate but connected applications were built, due to functional separation reasons. The applications are: METEO – main application of measuring system; SETUP – auxiliary application for administration of metrological station, it enables setup of Meteo application and particular measuring channels; CALIBRATOR – auxiliary application supporting calibration of measured parameters.

While working, the Meteo program does the following operations: data readout from µP driver via RS, data transformation to real values, data averaging every 5 and 30 minutes, writing data to files, displaying of panel with
current data, graphical and text presentation of measurements, sending of data files to central database, distant updating of WWW pages. The first operation after program activation is opening and configuration of PR port with connected µP interface of the measuring station. Opening of the communication port is done by the system function:

```
HANDLE CreateFile(
    LPCTSTR lpFileName, // pointer to name of the file
    DWORD dwDesiredAccess, // access (read-write) mode
    DWORD dwShareMode, // share mode
    LPSECURITY_ATTRIBUTES lpSecurityAttributes, // pointer to security attributes
    DWORD dwCreationDistribution, // how to create
    DWORD dwFlagsAndAttributes, // file attributes
    HANDLE hTemplateFile, // handle to file with attributes to copy
)
```

where: particular parameters are: lpFileName – name of opened object, dwDesiredAccess – way of access to opened object, dwShareMode – way of object sharing, lpSecurityAttributes – indicator to protection attribute structure (only in MSW/NT), dwCreationDistribution – determination of performed action in case of object existence or nonexistence, dwFlagsAndAttributes – flags and attributes determination for object, hTemplateFile – handle for file containing additional parameters for created object. Opening of the communication port in the program has the following form:

```
hComm = CreateFile(m_sComPort.c_str(), GENERIC_READ | GENERIC_WRITE, 0, NULL, OPEN_EXISTING, 0, NULL);
```

The name of used communication port is read from system registers.

If it is not possible to open a proper serial port, the program informs about this event with a proper message and starts work in off-line mode enabling written data survey.

Next stage is setting of parameters for used serial port. This is done via the following function:

```
BOOL GetCommState(
    HANDLE hFile, // handle of communications device
    LPDCB lpDCB, // address of device-control block structure, which reads data to DCB structure. Next it is necessary to define new settings through writing them into DCB structure and use function:
    BOOL SetCommState(
        HANDLE hFile, // handle of communications device
        LPDCB lpDCB // address of device-control block structure
    )
```

Which re-initializes setting of serial port. DCB structure consists of 28 work elements. To make the program work properly it is necessary to set only some of these elements:

```
typedef struct _DCB { // dcb
    DWORD BaudRate; // current baud rate
    DWORD fAbortOnError:1; // abort reads/writes on error
    BYTE ByteSize; // number of bits/byte, 4-8
    BYTE Parity; // 0-4=no,odd,even,mark,space
    BYTE StopBits; // 0,1,2 = 1, 1.5, 2 } DCB;
```

BaudRate – data transmission rate (bits/sec), fAbortOnError – behaviour description in case of errors at reading or writing, ByteSize – bit number, Parity – parity bit, StopBits – number of stop bits. Time parameters of the port are set with a function:

```
BOOL GetCommTimeouts(
    HANDLE hFile, // handle of communications device
    LPCOMMTIMEOUTS lpCommTimeouts // address of comm. time-outs structure );

BOOL SetCommTimeouts(
    HANDLE hFile, // handle of communications device
    LPCOMMTIMEOUTS lpCommTimeouts // address of communications time-out structure
)
```

Analogically to configuring the rest of parameters, COMMTIMEOUTS structure is used:

```
typedef struct _COMMTIMEOUTS { // ctm
    DWORD ReadIntervalTimeout;
    DWORD ReadTotalTimeoutMultiplier;
    DWORD ReadTotalTimeoutConstant;
    DWORD WriteTotalTimeoutMultiplier;
    DWORD WriteTotalTimeoutConstant;
}COMMTIMEOUTS,*LPCOMMTIMEOUTS;
```
ReadIntervalTimeout – defines maximal time in [ms] which may flow between reading of two signs from the port.
ReadTotalTimeoutMultiplier – multiplication (in ms) used to calculate total readout operation time; this value is multiplied by number of bytes to be read.
WriteTotalTimeoutMultiplier – multiplication (in ms) used to calculate total writing time; value multiplied by number of bytes to be written.
WriteTotalTimeoutConstant – constant (in ms) used to calculate total time of writing operation, value added to the result calculated for the value ReadTotalTimeoutMultiplier.

Part of the code setting parameters of serial port:

```c
m_bPortReady = GetCommState(hComm, &m_dcb);
m_dcb.BaudRate = 9600;
m_dcb.ByteSize = 8;
m_dcb.Parity = EVENPARITY;
m_dcb.StopBits = ONESTOPBIT;
m_dcb.fAbortOnError = TRUE;
m_bPortReady = SetCommState(hComm, &m_dcb);
m_bPortReady = GetCommTimeouts (hComm,  &m_CommTimeouts);
m_CommTimeouts.ReadIntervalTimeout = 50;
m_CommTimeouts.ReadTotalTimeoutConstant = 50;
m_CommTimeouts.ReadTotalTimeoutMultiplier = 10;
m_CommTimeouts.WriteTotalTimeoutConstant = 50;
m_CommTimeouts.WriteTotalTimeoutMultiplier = 10;
m_bPortReady = SetCommTimeouts (hComm,  &m_CommTimeouts);
```

The program reads data from the µP driver according to the following procedure: sending 1 byte containing address of device, from which data is to be read out, reading of 1 byte of data. The cycle is repeated until all accessible data at that device are read out. Function sending 1 byte with address: WriteFile(hComm, "\x01",1,&iBytesWritten,NULL); Function reading 1 byte of data and writing to buffer „InBuff“: ReadFile(hComm, InBuff, 1, &iBytesRead, NULL); The serial port RS-232 is closed with closing of the program. Closing operation is done with the function: BOOL CloseHandle( HANDLE hObject  // handle to object to close );

During time of work, the program stores data for measured parameter from each measurement performed every 1 minute from the last 24 hours and averaged values from 5 min for 24 hours and every 30 min from the last 7 days. An object model was introduced to facilitate data storing, calculations and management. A basic object is a single measured parameter, or measuring channel of the device. All channel parameters and functions performed on behalf of the channel are included in the class „KanalMeteo“ class KanalMeteo{

```c
  public: //tabele float *tab; float *tab24; float *tab7d; //wspolczynniki przeliczania wartosci float a;
   float b; //ograniczenia wykresu int min; int max;
  float hrozdz; float pnp; float diffpix; //stany alarmowe float alarm_min; float alarm_max; //kolory
int ColorTla; int ColorLinii; //nazwy String NazwaKanalu; String SkrotNazwy; String Jednostka; bool active;
  bool is_int; //funkcje
KanalMeteo::KanalMeteo(int PomiarDzien, int PomiarTydzien); void PrzeliczWartosc(); void PrzeliczWartosc_KierunekWiatru(); void PrzesuwTablic(); void PrzesuwTablic24();
`= void PrzesuwTablic7d(); void Srednia24(int LPomSred); void Srednia7d(int LPomSred); void Max24(int LPomSred); void Max7d(int LPomSred); void Min24(int LPomSred); void Min7d(int LPomSred); void Srednia24_KierunekWiatru(int LPomSred); void
```
Class definition includes indicators to tables with measurement results: from the last 24 hours done every minute, from the last 24 hours averaged for every 5 minutes, from the last 7 days averaged every 30 minutes, parameters to calculate real values from measured results, parameters used to visualize measurement data, full name of measured parameter and its abbreviation. The class functions are: channel parameter initialization with data from program registers, data calculations from measurement device to real values, calculations of tables for 24 hours and 7 days. Depending on measured parameter it may be averaging, summation, minimum maximum, measurement data table updating stored in memory, shift of previous data and writing current data to first positions in tables, etc.

The program reads data every 60 s. The readout function is called by system clock. The communication with driver relies on sending an address byte of a device from which we want to read data. On recognition of its address the device sends out the older (s) byte from two-byte measurement data. Then, the program again sends address and device responds with younger (m) byte. Proper result of measurement is obtained as a result of calculation: result = sBajt*256+mBajt; The µP device sends successive bytes for every readout – older and younger from each measurement channel. To read all data from a device it is necessary to go through the whole readout cycle of all 16 or 32 channels, even if some channels are unused. The whole cycle is to be performed even for single channel readout. Readout synchronization mechanism was introduced to prevent data shift and resulting errors of reading from improper channel. The program reads in the loop successive data from serial port and awaits for a sequence of three successive bytes “251”, “252” and “253”. Reception of bytes in this succession means, that he next byte will be older byte of measurement data. After synchronization, the program reads data from each channel and writes in the first position of the 24-hours table of measurement data. To read data a function is used: OdczytComm(), this function is called 16 times, single time for each channel. Synchronization and data readout algorithms are presented below as well as TimerTimer() function.

Algorithm of METEO application. The operations of Meteo application can be divided into two groups:

1. Operations performed directly after program launching: reading from registers a name of the used serial port, opening and configuration of serial port, reading and initialization of name and station number, creation for each measurement channel an object of KanalMeteo class and definition of table dimensions for measurement data: 60*24(hours) = 1440 for 24-H measurements every minute, 12*24(H) = 288 for 24-H measurements with averaging every 5 minutes, 2*24(H)*7(days) = 336 for 7-days measurements with averaging every 30 minutes; read in for each channel of re-calculation parameters, read in data visualization parameters and channel name (full and abbreviated), read in of activity flag, parameters readout concerning data sending to server, tables initialization for storing of measurement time, read in of data from previous measurements from files type *.mto, drawing of main program panel on the basis of the last measurement from the recently read in file “online.mto”.

2. Operations performed during cyclic (every 60 sec) calling by the system clock of TimerTimer() function; synchronization with µP device, read in of measurement data, assessment of measurement time, re-calculation of read measurement data to real values, display of actual parameters inside the main panel, averaging for 24 hours if the number defining actualization minutes is zero or is a multiplication of 5, averaging for 7 days if the number of actualization minutes is 0 or 30, table with measurement data writing to the “online.mto” file, removing previous content and creation the new file, shifting of table data of a single position what causes removal of the last table position; for 24 H averaging – writing 24H table to “online24.mto” file and then data shifting in this table, writing in averaged data to 24H file; for 7D averaging – writing of table 7D to “online7d.mto” file and then shifting data in table; updating of graphics - figures or data tables, depending on the chosen way of display; when the time of measurement is
in agreement with updating time – sending files with data to the server, updating web page presenting the metrological station data.

Data structure and archiving. Measurements results are archived as a database. The database consists of the following parts: 24H data file, 24H averaged data file, 7D averaged data file, database for 24H files. The structure of data in each part is the same. The data are written in text files in CSV format (comma separated file). The CSV format functions as follows: each column of the table is written after the semicolon (;) sign; each row of the table is ended with system code “CRLF” meaning CR – return of the carrier and LF – go to new line. Data are written in the form: 1 column – time of measurement, 2-17 columns – values from channels 0-15. Number of signs in each column is not confined. Zero values are written to the channels that are not used, i.e. have not connected sensors. The following data are written to particular columns: 1- measurement date (or averaging date), 2- (channel 0) average wind strength, 3- (channel 1) maximum wind strength, 4- (channel 2) minimum wind strength, 5- (channel 3) relative external humidity, 6- (channel 4) internal relative humidity, optionally, 7- (channel 5) number of atmospheric discharges, statics, optionally, 8- (channel 6) rain meter, 9- (channel 7) radiation level, ionizing radiation background, optionally, 10- (channel 8) wind direction, 11- (channel 9) optical radiation of the Sun, 12- (channel 10) contents of CO particles in air in ppm, 13- (channel 11) contents of butane particles in air in ppm, 14- (channel 12) contents of methane particles in air in ppm, 15- (channel 13) atmospheric pressure. Next 16 channels are used for measuring other optional values like: UV and IR radiation levels, air transparency, cloud base height, dust sedimentation, pH and alkali ions (salt) in rain water, snow intensity, noise level, measurement station intruder alarm, web audio and video channel, etc.

24H data file has constant name “online.mto”. Data are written successively from fresh to 24H old, altogether 1440 rows. After each readout from µP device, every minute, data are written to saved file in memory of table programme “tab” for each measurement channel:

```c
for(int i=0;i<TotLPom;i++){
    fprintf(online,"%02d:%02d; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f\n", hh[i],mm[i],
    kanal00->tab[i],kanal01->tab[i],kanal02->tab[i],kanal03->tab[i],kanal04->tab[i],kanal05->tab[i],
    kanal06->tab[i],kanal07->tab[i],kanal08->tab[i],kanal09->tab[i],kanal10->tab[i],kanal11->tab[i],
    kanal12->tab[i],kanal13->tab[i],kanal14->tab[i],
    kanal15->tab[i]); }
```

Data from „online.mto” file serve to restore measurement data tables used in application in the case of its intentional or accidental termination. The data are automatically read in, each time, during program initialization. The termination parameter of the file write loop „TotLPom” has the value of 1440 during normal work of the program. During the first loop the value is zero and increases by one after each performed measurement up to the value of 1440. 24H averaged data file has constant name „online24.mto”. Averaging is done every 5min for data readout from µP device, every minute, data are written to saved file in memory of table programme “tab” for each measurement channel: for(int i=0;i<TotLPom;i++){
    fprintf(online,"%02d:%02d; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f\n", hh[i],mm[i],
    kanal00->tab[i],kanal01->tab[i],kanal02->tab[i],kanal03->tab[i],kanal04->tab[i],kanal05->tab[i],
    kanal06->tab[i],kanal07->tab[i],kanal08->tab[i],kanal09->tab[i],kanal10->tab[i],kanal11->tab[i],
    kanal12->tab[i],kanal13->tab[i],kanal14->tab[i],
    kanal15->tab[i]); }

Data from „online.mto” file serve to restore measurement data tables used in application in the case of its intentional or accidental termination. The data are automatically read in, each time, during program initialization. The termination parameter of the file write loop „TotLPom” has the value of 1440 during normal work of the program. During the first loop the value is zero and increases by one after each performed measurement up to the value of 1440. 24H averaged data file has constant name „online24.mto”. Averaging is done every 5min for data readout from µP device, every minute, data are written to saved file in memory of table programme “tab” for each measurement channel: for(int i=0;i<TotLPom;i++){
    fprintf(online,"%02d:%02d; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f; %.1f\n", hh[i],mm[i],
    kanal00->tab[i],kanal01->tab[i],kanal02->tab[i],kanal03->tab[i],kanal04->tab[i],kanal05->tab[i],
    kanal06->tab[i],kanal07->tab[i],kanal08->tab[i],kanal09->tab[i],kanal10->tab[i],kanal11->tab[i],
    kanal12->tab[i],kanal13->tab[i],kanal14->tab[i],
    kanal15->tab[i]); }

Data from „online.mto” file serve to restore measurement data tables used in application in the case of its intentional or accidental termination. The data are automatically read in, each time, during program initialization. The termination parameter of the file write loop „TotLPom” has the value of 1440 during normal work of the program. During the first loop the value is zero and increases by one after each performed measurement up to the value of 1440.
channel in table „tab24[]”. Each time after averaging the contents of the „tab24[]” tables is written to a file in time succession from the newly calculated to 24H old ones. In sum, this file possesses 288 rows. To calculate average value for maximal wind strength channel is used the following function: Max24 (int LpomSred), which chooses maximal value of the last 5 measurements. Respectively, to the channel of minimal wind strength, a function is used Min24 (int LPomSred) which chooses minimal value from the last 5 measurements. For wind direction channel, from the last 5 measurements a value is chosen which is repeated most frequently. To choose this value a following function is used: Srednia24_KierunekWiatru(int lpomSred). For precipitation channel, the value is calculated with the aid of the function: Suma24(int LPomSred) as a sum of the last 5 measurements. Data from the file “online24.mto” are used to restore measurement data tables in case of its accidental or intentional termination. Each time, during program initialization these data are automatically read in. The “online24.mto” file is send to WWW server and the page is graphically updated according to the contents of this file with 24H drawings.

Fig.4. The panel of WWW METEO Zielonka and display blocks of – wind, temperature, pressure and web video channel.

24H file database. This part of database stores measurement data in 24H files. File names are combined with measurement date and are in the form: rrrrmmdd.mto, np. 20010601.mto includes measurements data from 1 June 2001. Averaged data for 5 min are added to 24H files. The same, which are written to “online24.mto” file. Data calculations are done with the same functions. 24H file is build during the first measurement at 00:00, every 5 minutes after calculations new data are added to the file end. After writing data for all 24H, during the nearest update, this file is sent to the central database on WWW server.

7D file with averaged data has a constant name „online7d.mto”. Averaging is done every 30 minutes for data readout at each full hour and 30 minutes later. Similarly to 24H file, for the same channels, to the considered file, a value is written which is an arithmetic average of the last 30 measurements. The mean value is calculated through function: Srednia7d(int LPomSred) of class KanalMeteo. Averaged data for each parameter is kept for each channel in table „tab7d[]”. Each time after averaging, table contents „tab7d[]” is recorded to a file beginning with the most recent data to the 7D old. In sum, the file has 336 rows. For maximal wind strength channel, to calculate average value a function is used: Max7d (int LPomSred), which chooses maximal value from the last 30 measurements. Other functions are – minimal wind strength Min7d (int LPomSred), which chooses minimal value from last 30 measurements. For the channel of wind direction, out of the last 30 measurements, the value is chosen which was repeated most frequently. To choose this value the following function is used:
Srednia7d_KierunekWiatru(int LPomSred). For precipitation channel, the value is calculated with a function: Suma7d(int LPomSred) as a sum of 30 last measurements. Data from file „online7d.mto” are used to restore measurement data table used by application in case of abrupt or intentional termination. During initialization these data are read in. „online7d.mto” file is sent to WWW server to update the site contents for 7D drawings.

Data presentation. Measurement data, processed in the „Meteo” application are presented in several different ways, while three of them are main: main panel, daigrammes and text data. A nondependent object class TPanel is used for each method. It is possible to switch between different windows during the work of the application. Menu “Sight” possesses 3 functions serving to change the way of data presentation. The main panel displays measurement results ant measurement time.

Measurement data are grouped by values: temperature group, wind, cloud and air, illumination, precipitation, gases, atmospheric discharges. Additionally the panel has time of measurements, program info, web video channel.

Fig. 5. Panel of measurement data tables in „Meteo” application. Presentation of average wind direction and sun illumination. Value readout from the curve.

Each displayed block in the panel is an object of Timage type. To display text or other graphical element on such an object, it is necessary to take indicator to Canvas parameter. Graphical functions included in Tcanvas class allow for drawing of contents of particular blocks. For example, for pressure block, application of graphical functions is:

- Taking of object indicator
  pwCanvas = MainForm->Img_cisn->Canvas;
- drawing of a black rectangle with green lining
  pwCanvas->Brush->Color = 0x00000000;
  pwCanvas->Pen->Color = 0x0000FF60;
  pwCanvas->Rectangle(0,0,MainForm->Img_cisn->Width,
  MainForm->Img_cisn->Height);
- Setting of font parameters used in function TextOut
The result of above function is presentation of the pressure block. Velocity and wind direction block presents real text data and clock like indicator. Three values are presented for wind velocity: max, min and average. Temperature block presents internal and external temperatures and wind chilling factor, felt temperature. The temperatures may be illustrated with mercury bar thermometer. Web camera image is situated in the main panel of the station. The value from tab[0] is displayed for each parameter.

Calculation sheet allows to display 24H measurement data for all channels. To present data is such a compact way an object TStringGrid was used. The data are presented from the youngest to the oldest. Each column has data from a single measurement channel. These are values form table tab[]. Updating takes place after each measurement.

The panel of diagrams (function charts) presents time dependent data for all measured parameters. The graphs for each channel are placed on separate folder. The folders are created with the object TPageControl. Each channel has the possibility to switch among three chart kinds: 24H generated on the basis of averaged data every 5min –data form table tab24[], 24H generated on the basis of all measurements during the last 24H – data from table tab[]. An algorithm of sending files with data is presented.

Because the characteristics of each measured parameter is different, each channel has defined parameters, which allow generation of clearly readable plot. The parameters are defined as arguments in the class KanalMeteo:min – minimal value visible n the plot, max – maximal value visible on the plot, hrozdz – separation between successive horizontal lines on the plat, pnp – number of pixels in the plot devoted to a single measurement point, by how many pixels successive measurement points are separated, diffpix – number of pixels falling to a unity of measuring parameter, this argument may be calculated from the following dependence: diffpix=(wysokość wykresu)/(max-min). The ordinate has hours of measurements. Vertical lines and relevant hours are plotted according to an algorithm, which does not allow the successive descriptions to overlap. The 0:00H line is plotted with different color. Horizontal lines are plotted for values obtained from the relation: linia=max-(i*hrozdz), where i, denotes next plotted line. This loop is repeated till the result reaches the value, smaller that min argument.

The 24H plot generated on the basis of all measurements has additional pulling strip, in order to enable full plot visualization. Most of the graphs are plotted as lines connecting successive measuring points. Rain meter result is plotted as bars from zero to the respective value. Wind strength and direction is plotted as average value and in the background at the envelope, wind direction is marked.
Each graph has an aiding function implemented, which facilitates reading of measurement values. Shifting cursor along the curve, the left lower corner of the graph window displays relevant measurement time and the exact value. This additional function turns out to be quite useful and acts as a kind of magnifying glass enabling finding of local curve minima or maxima.

Data transmission and update. The program may work in two modes. Offline – measurement data are gathered locally on a PC connected to µP measurement device. Online – apart from local data gathering, the data are periodically sent to server, where are written to database, and on the basis of updated database the WWW page is made current. The measurement data and their graphical presentation is updated. There are two cases for the online version: the PC is constantly connected to the Internet – then data updating takes place via this connection; the PC is periodically connected to the Internet via analog POTS dial up or digital ISDN modem – then before sending updating data a working and acknowledged Internet connection has to be established.

To establish and close telephone connections, the program uses function from wininet.dll libraries. Wininet library is an extension of win32 API interface added to the SO. Function call:

```c
InternetAutodial(INTERNET_AUTODIAL_FORCE_ONLINE, 0);
```

initiates modem connection with access server using default Dial-up technique, defined in the system. After ending the transmission the function is called:

```c
InternetAutodialHangup(0);
```

which causes telephone connection reset.

The files with measurement data are sent to the server using FTP protocol. TNMFTP component is used in the application. Transmission protocol is as follows: connection establishment with FTP server, sending of “online24.mto” 24H file, sending of “online7d.mto” file, sending file with whole 24H measurements – only after first update after 00:00H, sending of picture (jpg) from camera – only if this option is chosen, ending connection with FTP server. After sending data, the WWW page is updated for particular measurement station. A script, which does updating of data and graphs is done in PHP and located by the server. The application has the possibility to do the script using HTTP protocol. Script calling has to possess two parameters: date – update date and time in the form of rrrr-mm-dd-hh:mm, upd – station number. After calling the script, the station waits for answer. After receiving answer, displays information about updating on the status bar. File transfer and data updating time depends on the link throughput. This time cannot be defined or predicted exactly. The application makes use of it and following readouts may be done during file transmission to server.

A few additional functionalities were implemented after the first phase tests of the system, to increase usage. These are: Stop/Go – possibility to stop readout and make it go again; Send Data – sending data and updating “on demand”, data are send immediately after calling this function, such a function is useful in the case when the PC has no possibility of making automatic Internet connections; Re-initialization – reading in again program parameters without the necessity to stop measurements.

“Setup” block of tools serves to set parameters used in “Meteo” program. It contains 18 files. The first for main and general parameters concerning data transmission. Next 16 are used for measurement channels. The remaining one is for web camera. Next 16 are used for other, mentioned earlier, measurements. The file “Main” allows setting the following parameters: station name, station number, port to which µP device is connected, sending files to server (on-line work mode), usage of automatic Dial-up connection, setting of time HH.MM at which data sending takes place, catalogue name on server in which data are stored for this particular station. Each channel possesses a single fold. It is possible to set the following parameters for any measurement channel: channel name, abbreviated name, a and b parameters to calculate linear measurement data to its real value according to the relation $y=a*x+b$; $y$- real value, x-readout value, $[u]$ – measurement unit of particular value, parameters used to build graph of the value: min – minimal value visible on the graph, max – maximal value visible on the graph, VR- vertical resolution, number of pixels for unity of measured value, this parameter may be calculated from the relation: $\text{diffpix}=(\text{graph height})/(\text{max-min})$, HR – horizontal resolution, number of pixels for a single measurement, this means how many pixels are apart successive measurements; Separation – separation between successive horizontal lines on the graph. Camera fold chooses webvideo. Option “make a photo” means that the last photo is transmitted.
Fig. 6. File „main” in „setup” program. File with setup of measurement channel. “Calibrator” program. Main page of Metrological Server Zielonka.

“Calibrator” program helps to calibrate measured parameters. Connection with µP device and data readout is done in the same form as in “Meteo” application. The application may work in two modes: measurement done periodically every 30s and manually forced one. For each channel, in the readout field, there is displayed real value read from the µP device. The data may be added to the list, after writing in real value to the field. Function “Calculate”, using measurement points added to the list, determines a and b coefficients of linear transformation of data read from the µP device to the real value. Coefficient calculation bases on the least square method. Additional function, which enables calibration of slowly changing parameters (like atmospheric pressure), realizes writing of actual list of measurement points, and reading them in again.
5. DATA PRESENTATION

Public access to the measurement data is a basic assumption of the designed system. The graphical format of data presentation is standardized. A table with data is a module, which can be inbuilt into a page non-depending on its structure and way of generation. A script `update.php` after reading in data to generate HTML page does three functions: `top(); tabela($nazwa_stacji,$tabela_z_danymi,$czas_aktualizacji);` `down();` Function `tabela()` builds new table with current measurement results, functions `top()` and `down()` contain HTML code which will be positioned respectively before (up the page) and after (down the page) the table.

Fig. 7. Sending of measurement data. Part of the transmission line may be optical fibre.

Data from the last measurement are displayed on the web page. The data from previous measurements are presented as graphs. The graphs are displayed in new windows. The windows are opened optionally through active icon, respectively 24H or 7D. The display system is simple and readable.

Fig. 8. Data presentation examples. 24H graph of sun radiation; 7D graph of atmospheric pressure.

6. TESTS OF NETWORK COMMUNICATION

A basic factor in efficient functioning of the HOST is configuration, construction and throughput of Internet links used by the system. To test data losses and delays originating from the HOST a popular Visual Route application was used. The tests for Zielonka HOST station showed that the channel of sending data to server is virtually loss less. Average packed delay is 365ms and maximal slightly over 400ms. The table shows path of the packets during the updating procedure of data on the server, and packet delay increase on the successive routers.

<table>
<thead>
<tr>
<th>Hop</th>
<th>%Loss</th>
<th>IP Address</th>
<th>Node Name</th>
<th>Location</th>
<th>Tzone</th>
<th>ms</th>
<th>Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
<td>217.99.0.115</td>
<td>pt15.warszawa.cvw.ppp.tinet.pl</td>
<td>Warszawa, Poland</td>
<td>+01:00</td>
<td>0</td>
<td>463</td>
</tr>
<tr>
<td>1</td>
<td>217.99.0.1</td>
<td>pt1.warszawa.cvw.ppp.tinet.pl</td>
<td>Warszawa, Poland</td>
<td>+01:00</td>
<td>0</td>
<td>111</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>194.204.176.17</td>
<td>z.war-ar3.204.176.zonet</td>
<td>Warszawa, Poland</td>
<td>+01:00</td>
<td>115</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>213.26.12.25</td>
<td>z.war-r1.12.25.zonet</td>
<td>Warszawa, Poland</td>
<td>+01:00</td>
<td>114</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>194.204.175.1</td>
<td>z.war-r1.175.1.zonet</td>
<td>Warszawa, Poland</td>
<td>+01:00</td>
<td>130</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>195.205.0.6</td>
<td>z.war-r2.205.0.6.zonet</td>
<td>Warszawa, Poland</td>
<td>+01:00</td>
<td>367</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>195.205.229.206</td>
<td>-</td>
<td>Warszawa, Poland</td>
<td>+01:00</td>
<td>365</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Roundtrip time to 212.244.42.5, average = 365ms, min = 286ms, max = 401ms - 2001-10-4 22:53:05
The above results are, however, not representative for most of potential measurement points, further tests were done with mobile μP measurement devices including optoelectronic sensors. Several measurement points were set up, some of them around the City of Warsaw, which are equivalent to Zielonka localization. One or two points were localized in city center. One of the measurement points was located in DESY Institute in Hamburg, Germany. The obtained results show increased packet loss, even up to 30%. The average delays on the routers are close to previous tests. Comparable are also additive times of packet transmissions.

7. HOST NETWORK DEVELOPMENT AND CONCLUSIONS

There was designed and realized a model of hybrid optoelectronic metrological station and network HOST. The model uses optoelectronic technology where it is economically and technically justified. Some of the sensors are optoelectronic. Some of the transmission lines are fiber optic. Some of the sensors co-operate directly with fibre optic signal lines. Part of the system measures surface water parameters (in local rivers and water/sewage cleaning plants), and was not described here. This part bases to a serious extent on optical technologies. The system has modular structure. It is comparatively easy to add next modules with new sensors. The system is also able to co-operate with GSM/GPRS/UMTS telephony and with such Internet/mobile technologies as SMS, MMS, e:mail, etc. The newer solutions of the μP measurement devices use USB rather than RS232. Some μ devices have micro GSM/GPS terminals. Some of the μP devices have inbuilt micromodems, extended RAM, and do not require cooperation with a mediating PC measurement station – they are capable of sending data directly to web server. Some of the μP stations gather data for prolonged period of time.

Fig. 9. Localization of measurements points in Warsaw and vicinity. Testing of packet path for experimental HOST network along the route Warsaw-Hamburg-London.

Assumed work thesis, that it is possible to build a very cheap, distributed, public, telemetric network, using optoelectronic technologies was fulfilled. The costs of such a network are approximately 10 times and more (even up to 100 times) cheaper than, what is offered by commercial firms. Separation of measurement part off the PC and making it...
advanced in functionalities (adding a single chip micro PC) makes the system very versatile. Usage inside the CAC (local commune management offices) required nearly no intervention in localized here hardware and added nearly no costs there. Standard CAC connections to the web were used which did not require the ISP change and involved no additional costs.

Thanks to its non-dependence the metrological station may work in diversified network environment, or any PC having access to the web. It facilitates creation of next measurement points. Grouping sensors for several functionalities also facilitates the system development. Some groups of sensors are easier to be implemented than others. This concern such groups as: internal and external sensors; proximal and distant sensors; water, air, weather, environmental sensors, etc. Transmission tests virtually all over Europe showed system feasibility. A number of used sensors in the system were optoelectronic: cloud height base, air transparency, IR, UV, and optical radiation of the Sun, water parameters, optical ion-meters, fiber optic pH meter, etc.

8. ACKNOWLEDGMENTS

Demonstration model of the station and network was presented during the recent National Metrological Congress at Warsaw University of Technology. The authors would like to thank dr R. Kupczak from the Department of Physics of WUT for co-operation at model design and construction and for implementation in Town of Zielonka, Town of Płaszczno, and in DESY Institute. The authors also would like to thank the authorities of Zielonka Town and Commune for co-operation at this experiment and partial financing of the project.

REFERENCES