

# Measurement Data Logging via Bluetooth

*Kaj J. Grahn*

*Arcada Polytechnic, Helsinki, Finland*

[kaj.grahn@arcada.fi](mailto:kaj.grahn@arcada.fi)

*Christian Dickert and Frank Schneider*

*Fachhochschule Frankfurt am Main, Frankfurt am Main,  
Germany*

[christian.dieckert@diehl-avionik.de](mailto:christian.dieckert@diehl-avionik.de) [frank.schneider@suewaq.de](mailto:frank.schneider@suewaq.de)

*Göran Pulkkis*

*Arcada Polytechnic, Helsinki, Finland*

[goran.pulkkis@arcada.fi](mailto:goran.pulkkis@arcada.fi)

## Abstract

The implementation of measurement data logging with data communication based on Bluetooth technology is described. The design, implementation and testing of two applications are presented. The first application is an alarm system to monitor alarm sensors and warning systems on a mobile device. The second application is a temperature and voltage measurement system, which transfers temperature and voltage values. Both applications are built up around a Bluetooth developer kit. A Bluetooth enabled mobile notebook PC or a PDA acts as a server. The implementation methodology consisting of application architecture design, technological design, software design, recording of measurement data, Bluetooth data transfer, and visualization of measurement data is described in detail.

**Keywords:** Bluetooth, data logging, radio technology, sensor, temperature, voltage

## Introduction

Bluetooth is a low cost, low power and short range radio technology based on open standards. It supports both voice and data services, i.e. a combination of circuit and packet switching is used. Bluetooth is likely to be one of the most popular technologies for wireless personal area networks (PANs) (Bray & Sturman, 2001).

A number of vendors supply Bluetooth application development kits. These kits consist of a Bluetooth module connected to a computer, proper software and monitoring and debugging tools. Application Programming Interfaces (APIs) are used during the development work. APIs provide function calls that allow the application developer to access any protocol layer functionality.

---

Material published as part of this journal, either on-line or in print, is copyrighted by Informing Science. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission from the publisher at [Publisher@InformingScience.org](mailto:Publisher@InformingScience.org)

The paper describes a Bluetooth project carried out by students in Arcada (Arcada, 2005). The details of Bluetooth application development are described by the design of two Bluetooth applications for measurement data logging.

## Application Architecture

The objective is to implement measurement data logging using Bluetooth technology. A user with a notebook PC or a PDA can receive measurement data from a Bluetooth enabled sensor system without any cable connection. Application areas are:

- Industrial applications to record temporary data
- Car industry to monitor wheel pressure
- System engineering to monitor the actual state of an electrical installation
- Technical service to check and program a system

Two applications have been designed, built and tested:

1. **An alarm system**, i.e. Digital Port Request System, to monitor alarm sensors and warning systems on a mobile device. With appropriate software it is possible to immediately locate an alarming sensor.
2. **A temperature and voltage measurement system**, which transfers temperature and voltage values.

The architecture of both applications consists of 3 functional blocks (see Figure 1):

- Recording analogue measurement data on a control unit
- Receiving data from the control unit to a Bluetooth enabled system
- Transferring data via a Bluetooth connection from the system to the user equipment

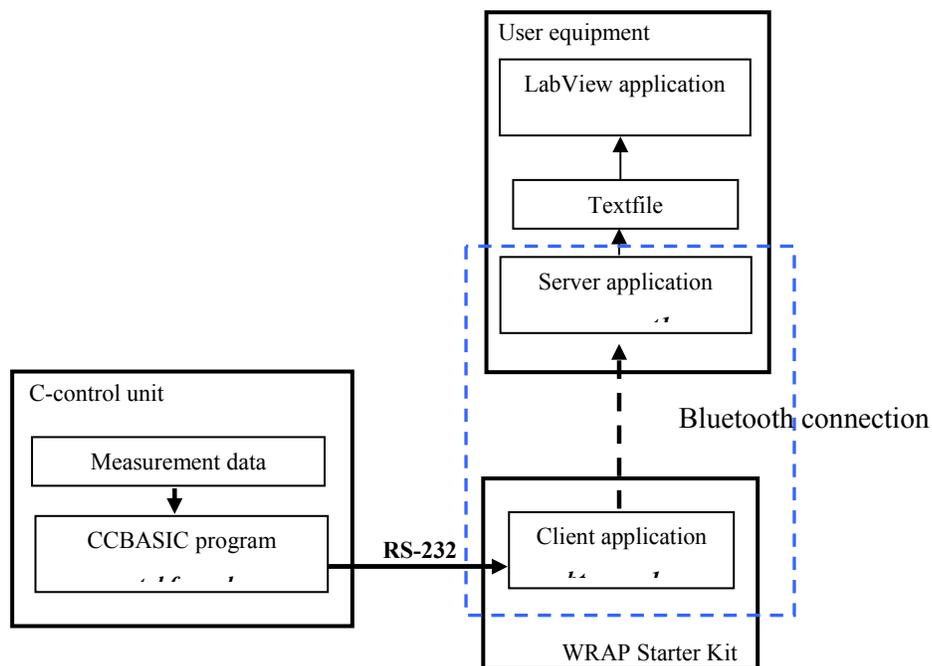


Figure 1. Functional blocks in Bluetooth data logging applications.

The functionality of the measurement data logging system is implemented by software modules written in C and C-BASIC code. The data communication of the Bluetooth connection is created by a C programmed client/server socket application.

## Technological Design

The following technology is used to realize the Bluetooth data logging architecture:

- Two BlueGiga WRAP 2151 Starter Kits (BlueGiga Technologies, 2003)
- Two C-control Plus Starter Sets (CONRAD, 2004)
- One PC for Bluetooth with Linux operating system
- One Notebook PC to receive and visualize data with Windows XP operating system
- One Bluetooth supported Palm m515 Handheld (Palm, 2004)
- One Bluetooth USB dongle (WIDCOMM, 2004)
- Sensors: AD592 transducer (temperature), C-control + batteries (voltage), C-control + switchboard (alarm system)
- LabView 7.0 software (University of Toronto, 2004)
- Laboratory facilities

A WRAP Starter Kit receives and records the measurement data through the RS-232 interface from the C-control unit. Then asynchronous transmission over the PPP/RFCOMM Bluetooth connection to the user equipment, notebook PC or handheld PDA, occurs. On the notebook PC data is displayed by means of a customized LabView application. Correspondingly, on the handheld PDA data is received and screened by using a server application.

**Development Kit.** The WRAP 2151 Starter Kit is a turnkey evaluation and development environment based on BlueGiga's WRAP™ MicroServer Technology (BlueGiga Technologies, 2004). In optimal conditions the working range is 30 m and the ordinary range is 10 m. The Starter Kit supports:

- Fully embedded Bluetooth protocol stack: Baseband, L2CAP, RFCOMM, SDP, Point-to-point and Point-to-multipoint.
- Bluetooth profiles: Serial Port, LAN Access, GAP
- TCP/IP protocol stack: IP, TCP, UDP, PPP. Utilities: DHCP, PING, Route
- Plug n' Play LAN Access Point

The Starter Kit has a  $\mu$ Clinux™ embedded operating system. Available programming languages are ANSI C, UNIX shell scripts, Waba and Java applets. WRAP I/O APIs and WRAP servers are used during program development. For further technical documentation, see (BlueGiga Technologies, 2003).

**C-control Plus Starter Set.** The basic task of the C-control system is to perform analogue to digital (A/D) conversion, to edit data and to send data over the serial port. The main parts included in the Starter Set are: starter board, C-control unit, temperature sensor, switchboard and two battery power packs. For further technical documentation, see (CONRAD, n.d).

**Bluetooth USB dongle.** The dongle establishes wireless links between the notebook and the Starter Kit. The delivered software WIDCOMM provides access to the Bluetooth operations and is used for configuration. For further information, see ("WIDCOMM", 2004).

**LabView.** LabView is a graphical development environment for signal acquisition, measurement analysis and data presentation. The user creates graphical surfaces called Virtual Instruments (VIs), see (University of Toronto, 2004). LabView was used in both applications.

## Software Design

The following software components have been designed or selected and used:

- **btclient** (modified/commercial). Application on the Starter Kit to read the serial port and to send data to the PDA.
- **btsensor1** (developed). UNIX network socket client application to read data from the sensor and to send data to the user equipment. Used by Starter Kit for the alarm system application.
- **btsensor2** (developed). As btsensor1 but for the temperature and voltage application.
- **btserver** (commercial). Server application on PDA to receive data from the sensor system.
- **receive** (developed). Test program on Starter Kit to read the serial port.
- **sensornet1** (developed). Winsock server application on a notebook PC to receive data from the sensor system and to store data in a text file.
- **portrequest.vi** (developed). LabView application on a notebook PC to display received alarm system measurement data.
- **temp\_volt.vi** (developed). LabView application on a notebook PC to display received temperature and voltage measurement data.
- **WIDCOMM** (commercial). Application software for Bluetooth USB dongle on a notebook PC.

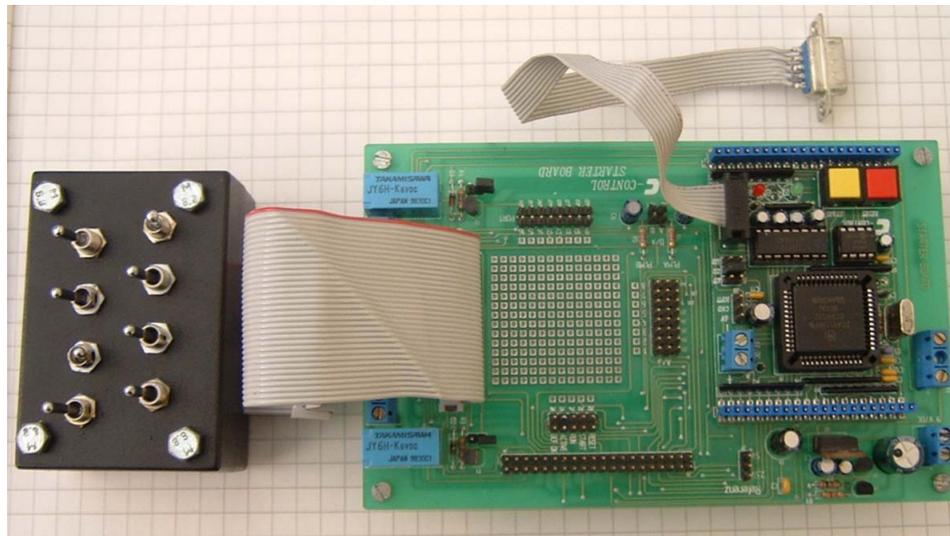
Programming of data communication of the Bluetooth connection was done for the LAN Access profile in order to connect to the sensors.

## Recording of Measurement Data

A C-control unit is used to get the analogue measurement data into digital signal form. The C-control unit is programmed to record required measurement data.

### ***Alarm (Digital Port Request) Application***

For this application the digital I/O ports of the C-control are connected with a switchboard, which supplies eight toggles to simulate the digital port status (see Figure 2).



**Figure 2. C-control unit and switchboard**

A simple C-BASIC program controls and puts out the status of the eight digital ports. The state of an active port is “-1” and an inactive port is “0”. The port numbers are also given by the program (See Figure 3).

```

1  define d1 port[1]
2  define d2 port[2]
3  define d3 port[3]
4  define d4 port[4]
5  define d5 port[5]
6  define d6 port[6]
7  define d7 port[7]
8  define d8 port[8]
9
10
11 #test
12
13 pause 25
14 print "1P1: ";d1
15 print "P2: ";d2
16 print "P3: ";d3
17 print "P4: ";d4
18 print "P5: ";d5
19 print "P6: ";d6
20 print "P7: ";d7
21 print "P8: ";d8
22
23 goto test
24
25 end

```

**Figure 3. Program code of *portrequest.bas* .**

The first eight lines define the digital ports one to eight. Then, after the program stops for 10 ms, the port number and the status of the port is prompted. The line “goto test” completes the endless loop condition. A screenshot of this C-BASIC program is shown in Figure 4. The status of the ports can be changed during execution, as done approximately in the middle of the screen.

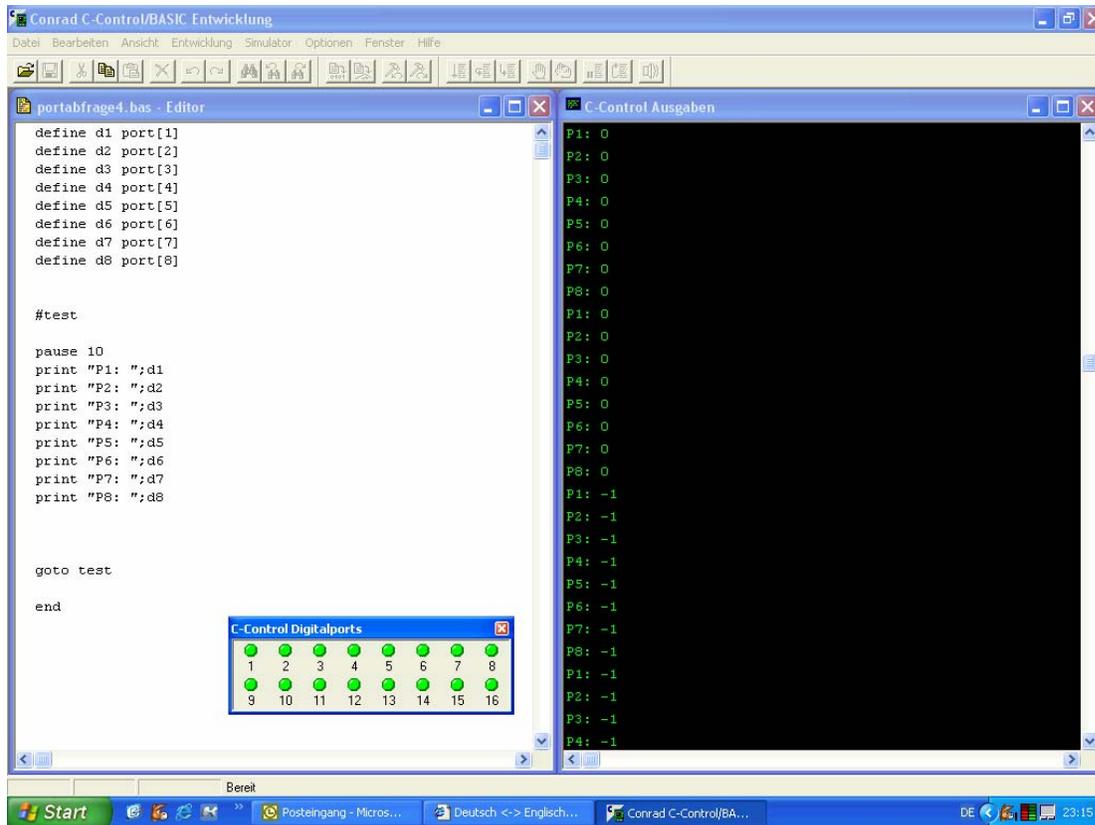


Figure 4. Screenshot of *portrequest.bas*.

## Voltage and Temperature Measurement Application

### Voltage measurement

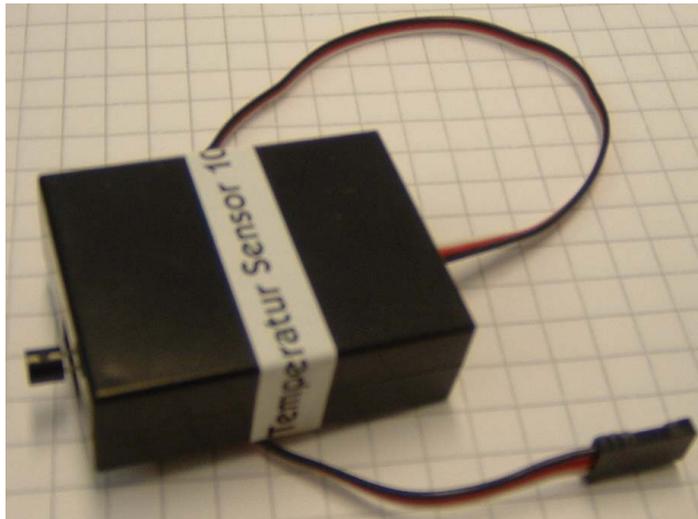
Referring to measure a voltage the C-control board allows connecting the measurement object direct to the analogue ports. Needed reference voltage is supplied by the C-control. In this case the object has been two serial connected battery power packs, each 1.2 V. This example is used to monitor the voltage level of these batteries. To depict alterations in the voltage level we used an adjustable power supply instead of batteries.

### Temperature measurement

The temperature sensor externally consists of the module housing, the temperature sensing device and the connection cables, see Figure 5. Electrical properties are influenced by the ambient temperature. The electronic inside the sensor transforms the temperature into a voltage signal between 0 and 2.5 Volt.

Main part of the electric circuit is the AD592. The AD592 is a two terminal monolithic integrated circuit temperature transducer that provides an output current proportional to the absolute temperature. For a wide range of supply voltages the transducer acts as a high impedance temperature

dependent current source of 1 mA/K. The AD592 can be employed in applications between –25°C and +105°C.



**Figure 5. Temperature sensor.**

### C-control unit program

Both the temperature and voltage sensors were connected to the analogue inputs. This means, the C-control compares the measured values with the reference voltage and after that interprets the voltage or the temperature as a value between 0 and 255. The program in Figure 6 recalculates these data into a voltage or temperature value.

```

1  define sensor1 ad[1]
2  define temp word
3
4  define sensor2 ad[2]
5  define voltage word
6
7  #program
8
9  temp = 10*(sensor1-50) / 2
10
11 voltage = (sensor2*10) / 102
12
13 print "2T : " ; temp/10 ; "," ; temp mod 10
14
15 print "V : " ; voltage/10 ; "," ; voltage mod 10
16
17 pause 25
18
19 goto program
20
21 end

```

**Figure 6. Program code of *analog\_data.bas*.**

In the first five lines, the analogue inputs, sensor1 and sensor2 and the words temp and voltage, are defined. In lines 9 and 11, conversion of the transformed values of the analogue input is made.



When doing serial communications, each word (i.e. byte or character) of sent and received data is sent one bit at a time.

RS-232 is a standard electrical interface for serial communications defined by the Electronic Industries Association ("EIA"). RS-232 actually comes in 3 different flavours (A, B, and C) with each one defining a different voltage range for the *on* and *off* levels. The most commonly used variety is RS-232C, which defines a mark (on) bit as a voltage between -3V and -12V and a space (off) bit as a voltage between +3V and +12V. The RS-232C specification says these signals can go about 25 feet (8 m) before they become unusable. You can usually send signals a bit farther than this as long as the baud rate is low enough.

## Asynchronous communication

For the computer to understand the incoming serial data, it must determine where one character ends and the next begins.

In asynchronous mode the serial data line stays in the mark (1) state until a character is transmitted. A *start* bit precedes each character and is followed immediately by each bit in the character, an optional parity bit, and one or more *stop* bits. The start bit is always a space (0) and tells the computer that new serial data is available. Data can be sent or received at any time, thus the name asynchronous. See Figure 8.

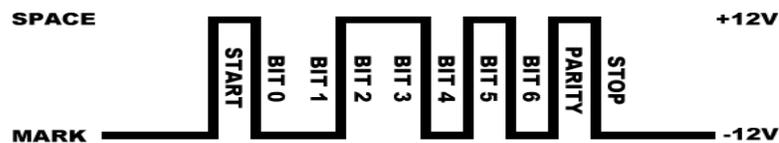


Figure 8. Asynchronous data transmission.

The optional parity bit is a simple sum of the data bits indicating whether or not the data contains an even or odd number of bits. With *even parity*, the parity bit is 0 if there is an even number of 1's in the character. With *odd parity*, the parity bit is 0 if there is an odd number of 1's in the data.

The remaining bits are called stop bits. There can be 1, 1.5, or 2 stop bits between characters and they always have a value of 1. Stop bits traditionally were used to give the computer time to process the previous character, but now only serve to synchronize the receiving computer to the incoming characters.

## Accessing the Serial Ports

The embedded operation system of the Starter Kit is Linux, an implementation of UNIX, which provides access to serial ports via *device files*. To access a serial port we open the corresponding device file. The device of the Starter Kit's user serial port is called `/dev/ttyS3` and its baud rate is usually 115200.

## Test Program

A test program `receive.c` is programmed to read and store the data from the serial port.

After connecting the C-control unit with the Bluetooth Starter Kit by a serial null-modem cable and starting the C-control application the compiled `receive.c` program can be executed on the

Starter Kit. For testing the C-BASIC program of the Port Request System is executed on the C-control unit. With a TELNET session the received data can be displayed on a local host computer (see Figure 9).

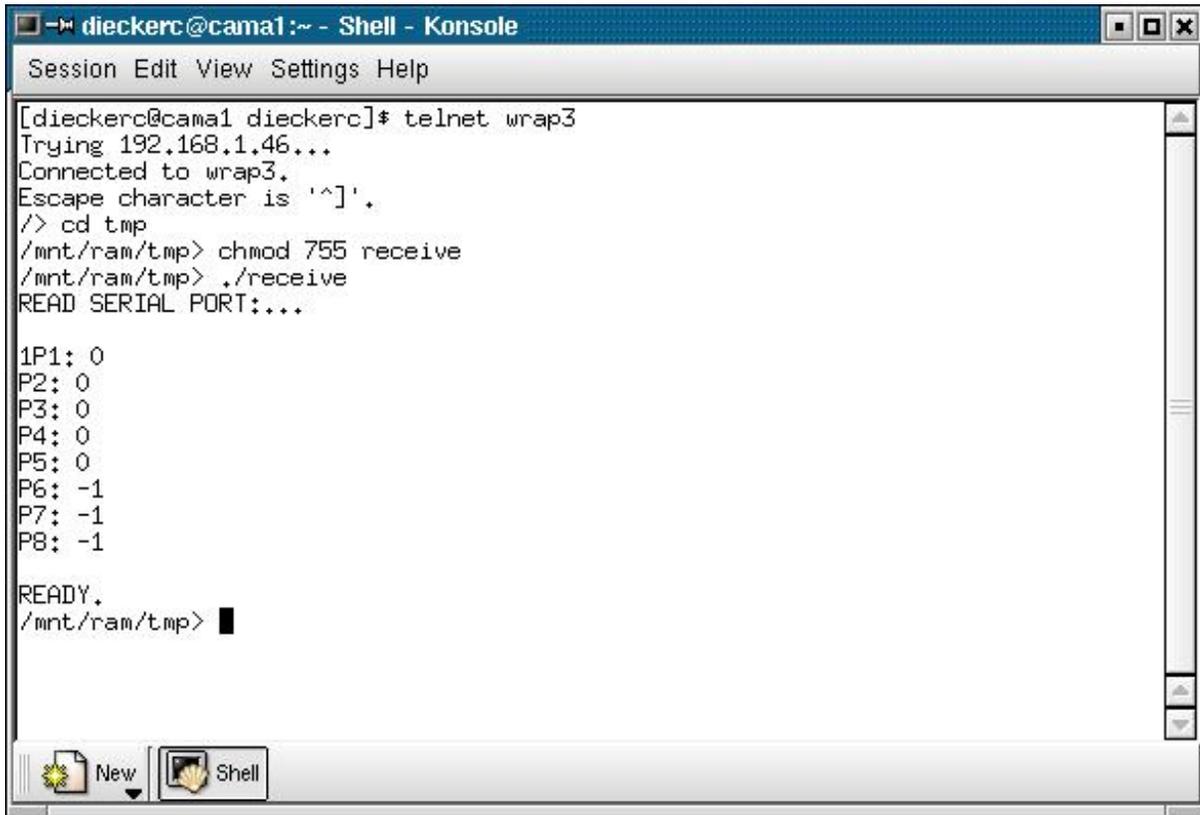


Figure 9. Output of the test program *receive.c*

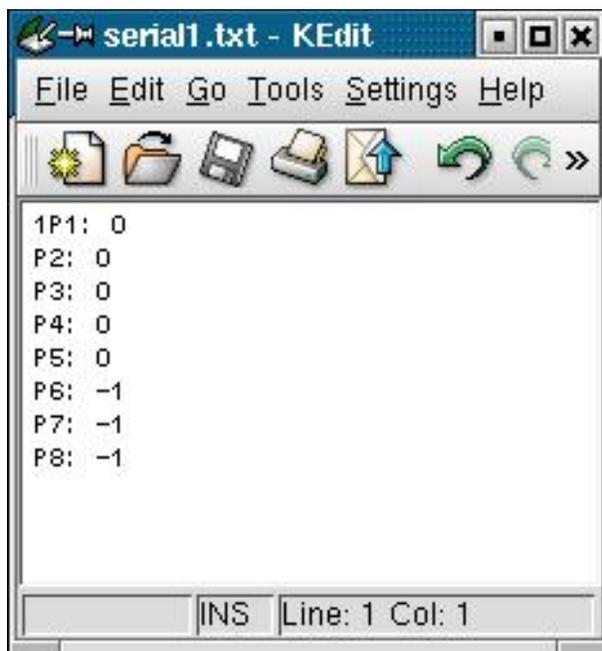


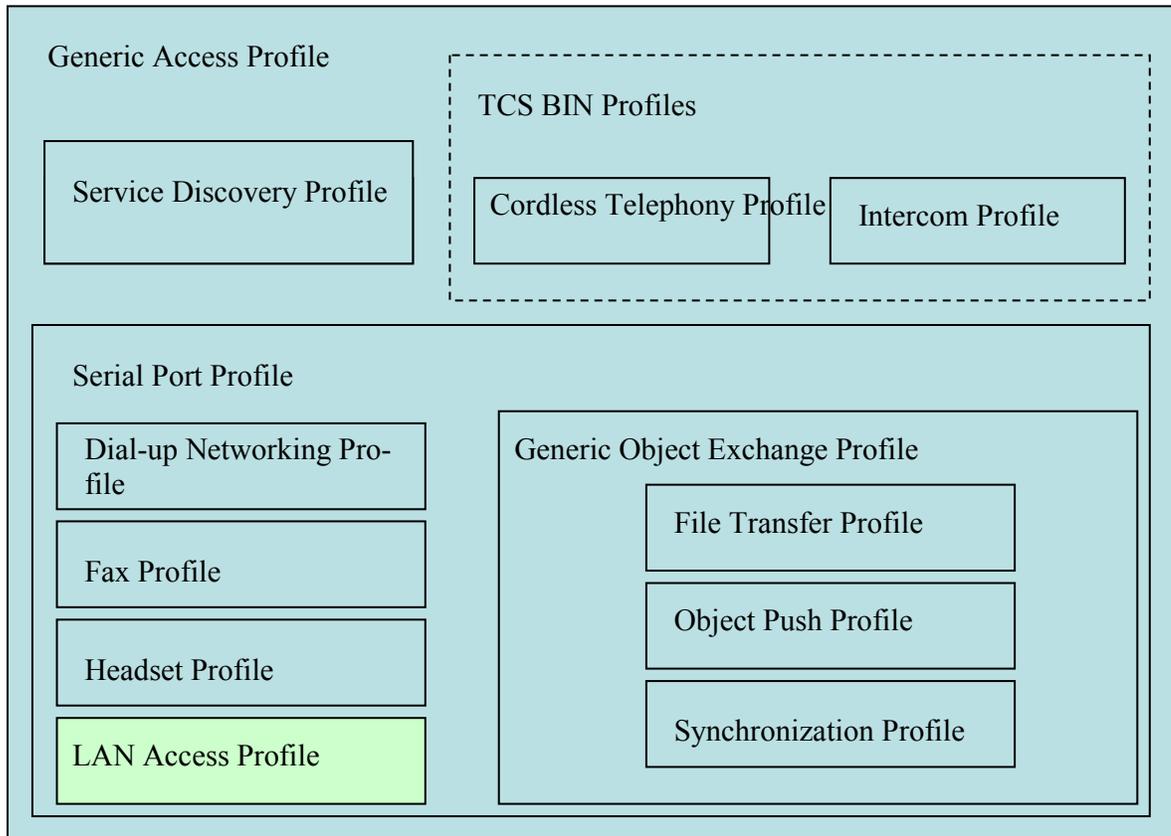
Figure 10. Screenshot of *serial1.txt*.

The received data is stored in the internal RAM of the Starter Kit as a text file "serial1.txt" (see Figure 10).

The code of the test program is embedded in socket client programs *btsensor1.c* and *btsensor2.c* of the Bluetooth transmitter (Wrap 2151 Starter Kit).

## Bluetooth Data Transfer to User Equipment

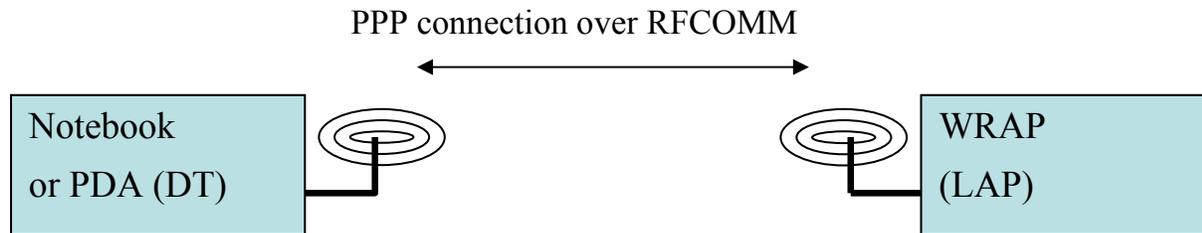
The Bluetooth SIG has identified various usage models, each of which has a supporting “profile” that defines the protocols and features the usage models draw upon for implementation (see Figure 11).



**Figure 11. Different Bluetooth profiles.**

The LAN Access profile is used to connect the sensors with the user equipment. Generally this profile defines how Bluetooth devices can access the services of a LAN using the Point-to-Point protocol (PPP) over RFCOMM. It shows how the PPP mechanisms are used to form a network consisting of two Bluetooth-enabled devices. Here PPP networking over serial cable emulation is used

There are two roles defined for Bluetooth devices in this profile, LAN Access point (LAP) and Data Terminal (DT). The Bluetooth Wrap 2151 Starter Kit (with a wired connection to the data sensor) provides access to the cable modem, so it assumes the role of the LAP. The PPP connection is carried over RFCOMM which is used to transport the PPP packets and for flow control of the data stream. The user equipment, notebook PC or handheld PDA, takes place as the DT by using the services of the LAP. The DT is a PPP client. It forms a PPP connection in order to gain access to the cable modem. See Figure 12.



**Figure 12. Bluetooth connection based on the LAN Access profile**

Summary of the interactions between a DT (Notebook or PDA computer) and a LAP (the Bluetooth Starter Kit connected to the data sensors):

1. By using an application on the DT, for example Bluetooth BTW Utility, the sensor supporting LAP has to be found and selected.
2. The DT requests a base band physical link with the selected LAP. After the physical link establishment, the devices perform mutual authentication. This procedure bases on the stored link key or by pairing. Each device insists that encryption is used on the link.
3. The DT establishes a PPP/RFCOMM/L2CAP connection.
4. The LAP may use some other PPP authentication mechanism. When we connect to the sensor (LAP), the LAP may challenge the DT user to authenticate herself or himself. The DT has to supply a username and password, otherwise the PPP link will be dropped.
5. Using the appropriated PPP mechanisms, a suitable IP address is negotiated between the sensor and the DT.
6. Now IP traffic can flow across the PPP connection.

Measurement data can be transferred with a socket server program on the DT and a socket client application on the LAP.

### ***Socket Server Program of Bluetooth Receiver***

The main task of the socket server program *sensornet1.c* is to receive and store the measurement data. A flowchart of the program is shown in Figure 13.

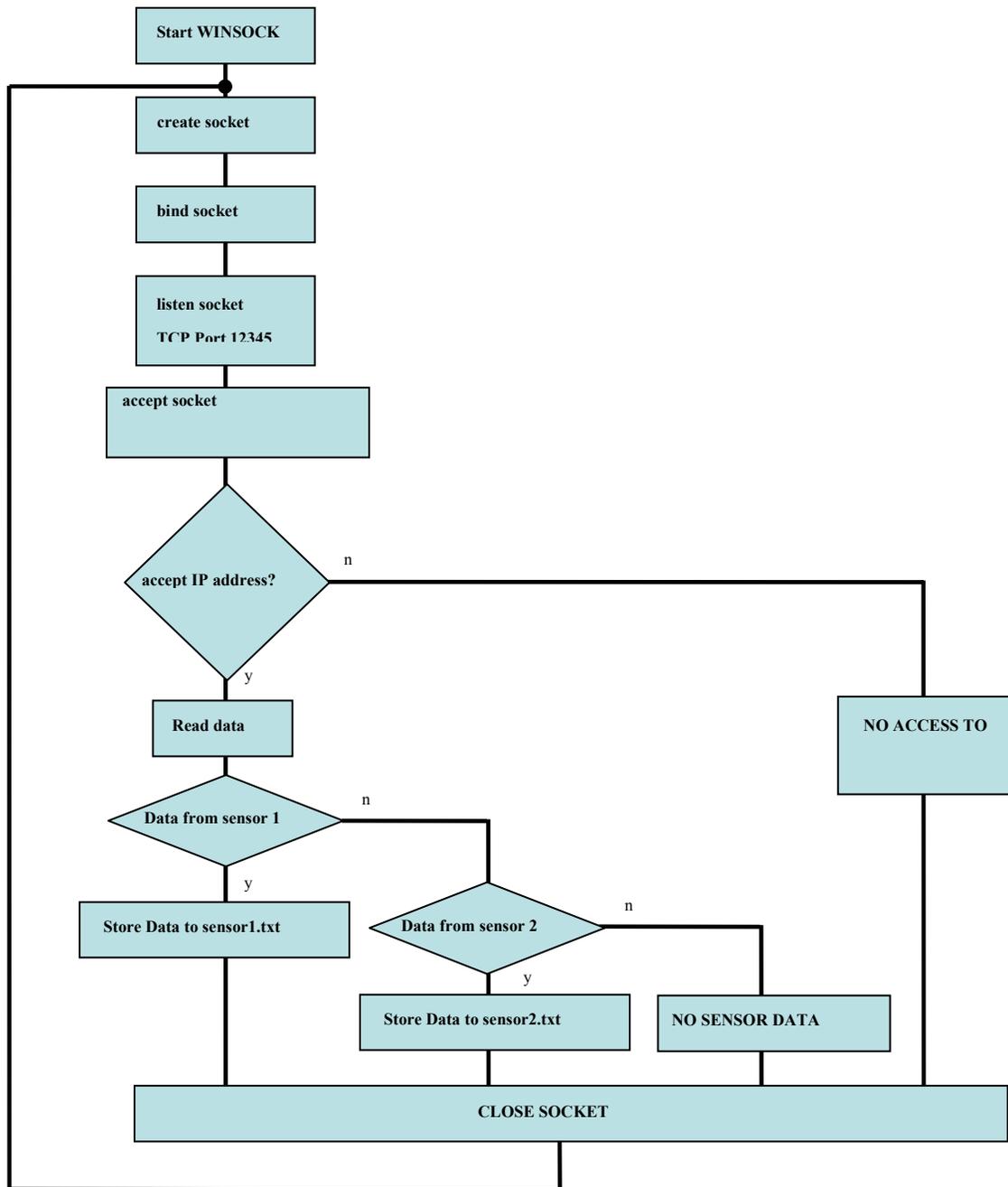


Figure 13. Flowchart of socket server program *sensornet.c*.

The socket server program *sensornet.c* is executed on a notebook PC/Windows computer, when the received data is visualized by a LabView application.

On a PALM m515 Handheld computer a commercial socket server program *BTServer* was executed (WHIZOO, 2005). Figure 14 shows the screen of the Palm computer when measurement data has been received.

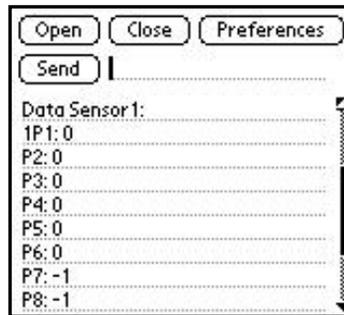


Figure 14. Measurement data displayed on the Palm m515.

### Socket Client Program on Bluetooth Transmitter (WRAP Starter Kit)

The data sensor systems are shown in Figure 15 and Figure 16.

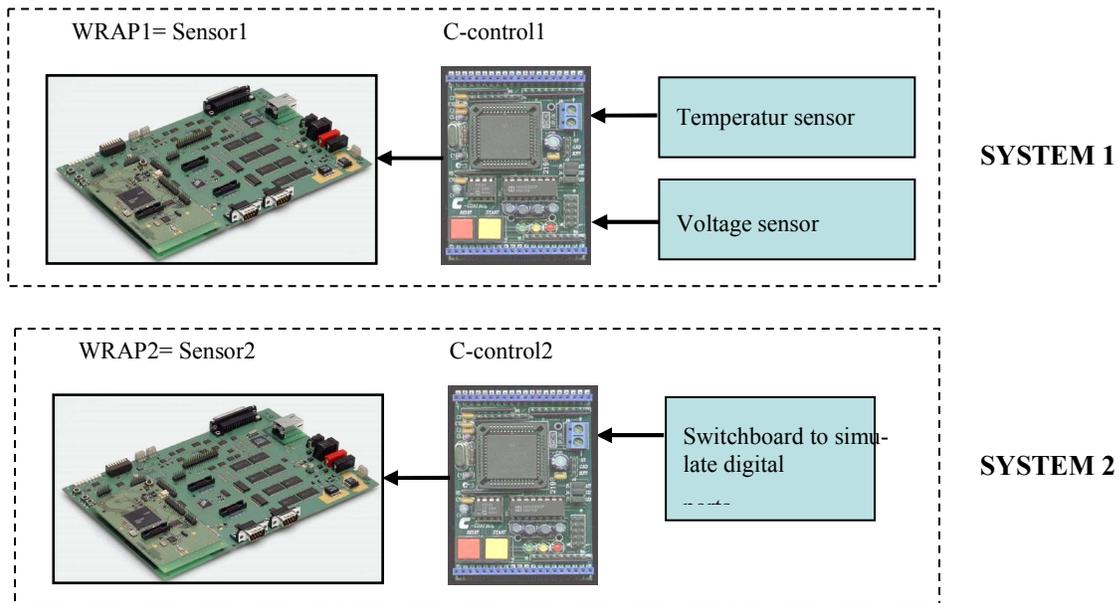


Figure 15. Both data sensor systems.

	WRAP Starter Kit	C-control
SYSTEM 1	<i>btsensor1.c</i>	<i>portabfrage.bas</i>
SYSTEM 1	<i>btsensor2.c</i>	<i>temperatur.bas</i>

Figure 16. Implemented software.

The socket client programs *btsensor1.c* and *btsensor2.c* are executed when a notebook PC/Windows computer is a Bluetooth receiver. A flowchart of both programs is shown in Figure 17.

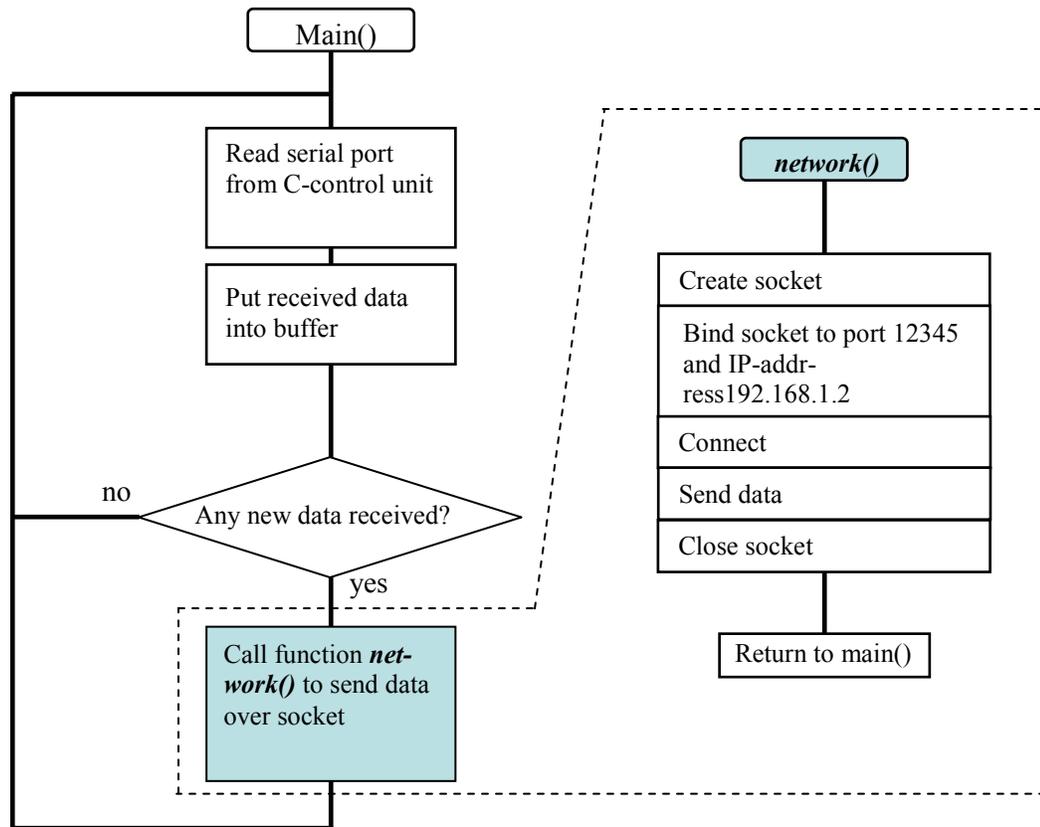


Figure 17. Flowchart of *btensor1.c* and *btensor2.c*.

A modified version of the socket client program *Btclient.c*, supplied as an example program with the WRAP 2151 Starter Kit is used, when a Palm Handheld computer is Bluetooth receiver.

## Visualization of Transmitted Measurement Data

On a notebook PC/Windows computer as a Bluetooth receiver incoming data is displayed on the screen with developed a LabView application.

For the first sensor example “Digital Port Request” the application file *portrequest.vi* is opened. After starting the application the state of the digital ports is displayed as shown in Figure 18 with the received data shown in Figure 19. This example represents a ground plan of a building. The signal LEDs in the different rooms indicate the state of each digital port. An alarm sensor could be connected to each digital port. Here sensors are simulated by a switchboard.

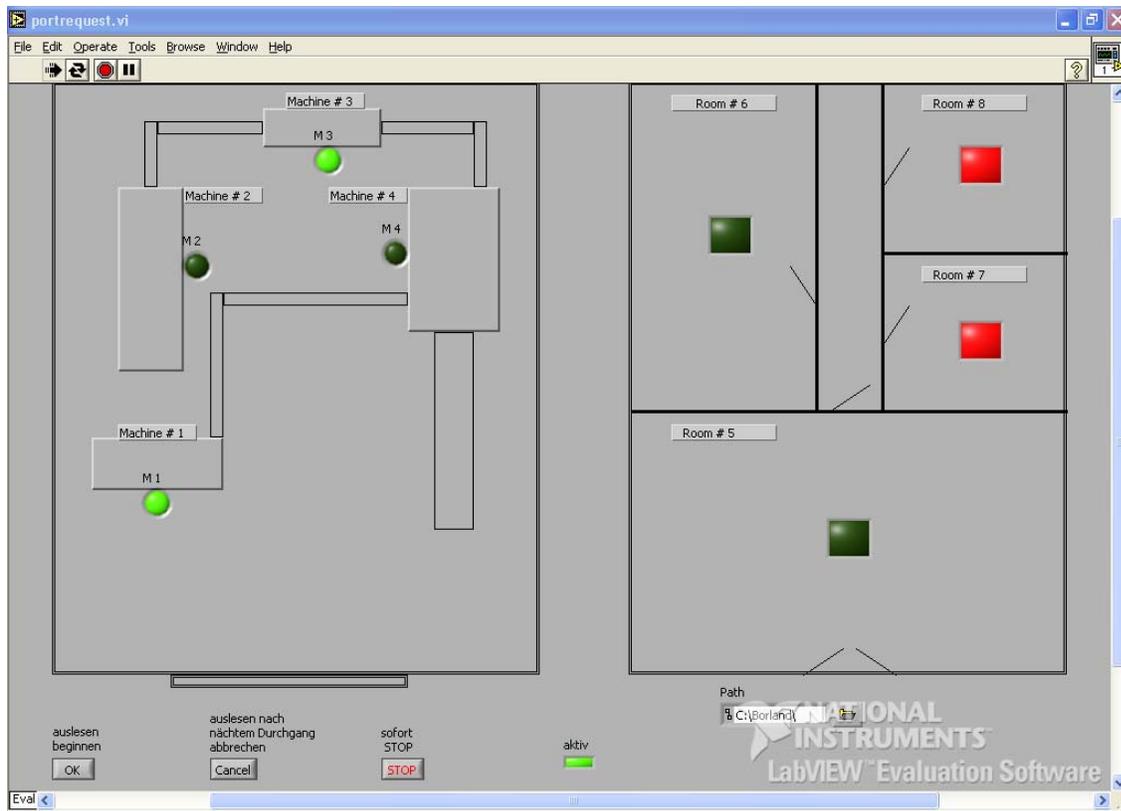


Figure 18. LabVIEW application *portrequest*.

```

sensor1.txt:

P1: -1
P2: 0
P3: -1
P4: 0
P5: 0
P6: 0
P7: -1
P8: -1
    
```

Figure 19. Received sensor data visualized in Figure 18

For the second example “Temperature and Voltage Measurement” a PPP/RFCOMM connection with sensor system 2 and the running client application on sensor system 2 is required. The LabVIEW application file *temp\_volt.vi* is opened. After starting the application the received measurement data is displayed on the screen as shown in Figure 20. The received data is shown in Figure 21.

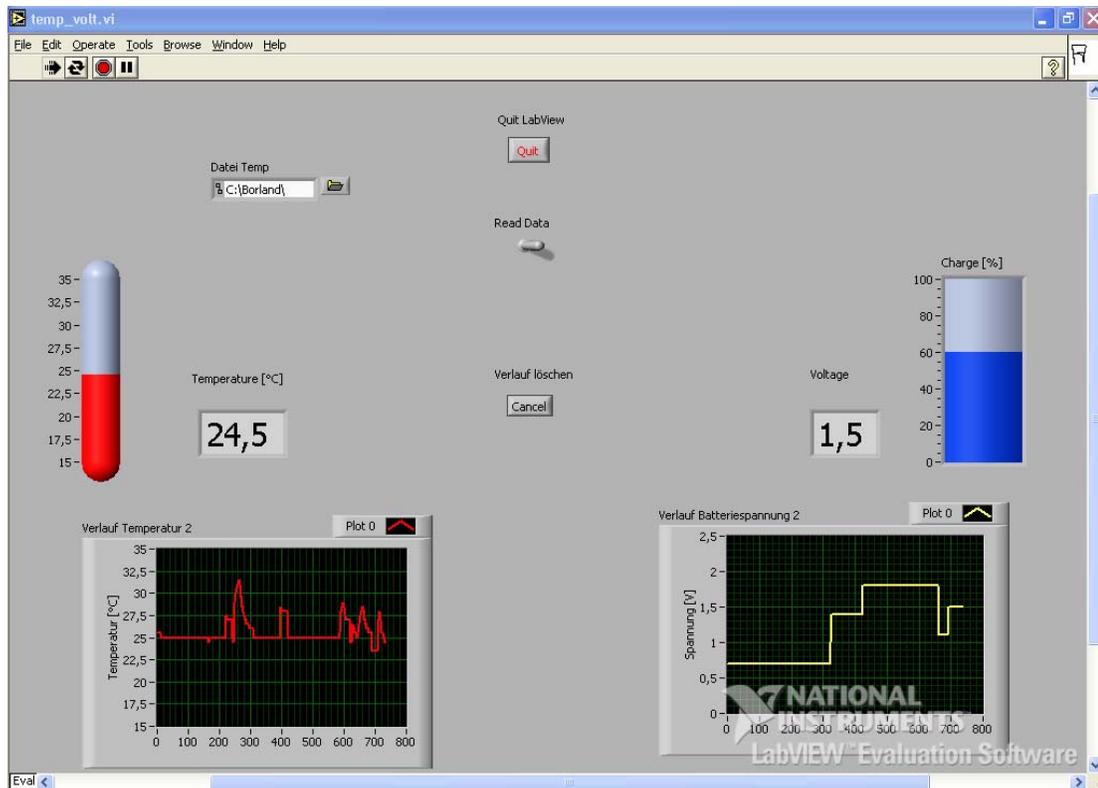


Figure 20. LabVIEW application *volt\_temp*.

```

sensor2.txt:

T: 24.5
V:1.5
    
```

Figure 21. Received sensor data visualized in Figure 20

## Testing

Both applications were tested extensively. The established system is stable and reliable. Also Bluetooth USB dongles constructed by different manufacturers were tested. The transmission range with the used hardware is about 10 meters depending on the environment between the sensor systems and the user equipment.

## Conclusions

Bluetooth sensor applications for industrial purposes were presented. The system is built up around Bluetooth developer kits. A Bluetooth enabled mobile notebook PC or a PDA acts as a server. The system can easily be applied to other Bluetooth enabled, mobile devices.

The use of developer kits enables students and researchers to quickly get an understanding of a new technology, in this case Bluetooth technology. A developer kit also speeds up the development work, because basic hardware and software functions are already pre-implemented. From a pedagogical point of view this approach allows a combination of theory and practice.

As Bluetooth has become standard in many mobile devices there is no need for special user equipment. Correspondingly, cable replacement in medical environment by using Bluetooth technology and health monitoring sensors are potential applications in the near future.

## References

- Arcada. (2005). Web Portal of Arcada Polytechnic, Helsinki, Finland. Retrieved March 10, 2005 from <http://www.arcada.fi>
- BlueGiga Technologies. (n.d). *WRAP 2151 Starter Kit*. Retrieved March 14, 2003, from <http://www.bluegiga.com/products/W2151.pdf>
- Bray, J. & Sturman, C. F. (2001). *Bluetooth 1.1. Connect without cables* (2<sup>nd</sup> ed.). USA: Prentice-Hall.
- CONRAD. (n.d). *C-Control Description I – Next Generation*. Retrieved November 10, 2004, from <http://www.c-control.de/>
- Palm. (n.d.). *Palm M515 Handheld*. Retrieved June 5, 2004, from <http://www.palm.com/products//palmm515/details.html>
- University of Toronto, Department of Physics. (2004). *LabView Tutorials*. Retrieved September 4, 2003, from [www.upscale.utoronto.ca/GeneralInterest/LabView.html](http://www.upscale.utoronto.ca/GeneralInterest/LabView.html)
- WHIZOO (2005), *BtServer for the Palm OS*, Retrieved February 12 2005 from [http://applications.palmsource.com/Software/prod\\_frame.asp?appurl=http://www.whizoo.com/apps/btserver.php](http://applications.palmsource.com/Software/prod_frame.asp?appurl=http://www.whizoo.com/apps/btserver.php)
- WIDCOMM. (n.d.). *Bluetooth USB dongle*. Retrieved November 10, 2004, from <http://www.widcomm.com>

## Abbreviations and Concepts

Access Point	A hardware device or a computer's software that acts as a communication hub for users of a wireless device to connect to a wired LAN.
API	Application Programming Interface. API is a set of routines, protocols and tools for building software applications.
Applet	A program designed to be executed from within another application. Unlike an application, applets cannot be executed directly from the operating system.
ANSI C	ANSI C is a version of the C language that has been approved by the American National Standards Institute.
BASIC	Beginner's All-purpose Symbolic Instruction Code. BASIC is one of the earliest and simplest high-level programming languages.
Baseband	A transmission method in which the entire bandwidth of the transmission medium is used to transmit a single digital signal. The signal is ascribed directly onto the transmission medium without modulation of any kind. Simultaneous transmissions can be achieved through time division multiplexing. Baseband is simpler, cheaper and less sophisticated than Broadband.
DHCP	Dynamic Host Configuration Protocol. DHCP enables individual computers on an IP network to extract their configurations from a server.
GAP	Generic Access Profile. This profile defines the generic procedures related to discovery of Bluetooth devices and link management aspects of connecting to Bluetooth devices. It also defines procedures related to use of different security levels.
IP	Internet Protocol. IP specifies the format of packets, also called datagrams, and the addressing scheme. Every host needs at least one IP address.

I/O	Input/Output. The parts of a computer system that transfer data to or from peripheral devices.
Java	A high-level object-oriented programming language developed by Sun Microsystems.
LabView	Automation software designed to create test, measurement and automation solutions through data acquisition, data analysis and data visualization.
LED	Light Emitting Diode. An electronic device that lights up when electricity is passed through it.
L2CAP	Logical Link Control and Adaptation Protocol. A protocol that interfaces between the application layer and the physical layer in a Bluetooth communication system. The L2CAP allow applications to transmit and receive data packets of up to 64 Kbps in length.
PAN	Personal Area Network. A network concept in which all the devices in a person's life communicate and work together, sharing each other's information and services..
PDA	Personal Digital Assistant. Small computing devices that contain it's own software operating system that allows the user to run software processing applications. Personal digital assistants are often used to organize personal activities and may provide access to communication services.
PING	A utility to determine whether a specific IP address is accessible. It works by sending a packet to the specified address and waiting for a reply. PING is used primarily to troubleshoot Internet connections.
PPP	Point-to-Point Protocol. A method of connecting a computer to the Internet. PPP provides error checking features. Working in the data link layer, PPP sends the computer's TCP/IP packets to a server that puts them onto the Internet.
RAM	Random-Access Memory. The most common computer memory which can be used by programs to perform necessary tasks while the computer is on. An integrated circuit memory chip allows information to be stored or accessed in any order and all storage locations are equally accessible.
Route	A routing algorithm used in Bluetooth scatter nets for routing of IP datagrams.
RFCOMM	A communication protocol used in the Bluetooth system that uses packet messages to emulate operation of a serial communication port. It is capable of simulating the software and hardware operations of RS-232 (EIA/TIA-232-E) serial ports.
RS-232	Recommended Standard-232, a standard interface approved by the Electronic Industries Alliance (EIA) for connecting serial devices.
SDP	Service Discovery Protocol. SDP is the communication messaging protocol used by the Bluetooth system to allow devices to discover the services and capabilities of other devices.
Serial Port	A port, or interface, that can be used for serial communication, in which only 1 bit is transmitted at a time.

## Measurement Data Logging via Bluetooth

TCP	Transmission Control Protocol. TCP enables two hosts to establish a connection and exchange streams of data. TCP guarantees delivery of data and also guarantees that packets will be delivered in the same order in which they were sent.
UDP	User Datagram Protocol A connectionless protocol that, like TCP, runs on top of IP networks. UDP/IP provides very few error recovery services, offering instead a direct way to send and receive datagrams over an IP network.
UNIX	A multi-user, multitasking operating system developed at Bell Labs in the early 1970s. UNIX was designed to be a small, flexible system used exclusively by programmers. UNIX was one of the first operating systems to be written in a high-level programming language namely C. This meant that it could be installed on virtually any computer for which a C compiler existed.
USB	Universal Serial Bus. An external bus standard that supports data transfer rates of 12 Mbps. A single USB port can be used to connect up to 127 peripheral devices, such as mice, modems, and keyboards. USB also supports Plug-and-Play installation.
VI	Virtual Instrument. LabVIEW programs are called virtual instruments (VIs) because the appearance and operation imitate actual instruments. VIs may be used directly by the user or as a subroutine (called subVIs) of a higher program which enables a modular programming approach.
Waba	Waba is a programming platform for small devices. Waba defines a language, a virtual machine, a class file format and a set of foundation classes. The syntax of the Waba programming language is a strict subset of the syntax of the Java language. This allows developers who are familiar with Java to quickly start programming in Waba.
WIDCOMM	Application software for Bluetooth USB dongle on a Notebook PC.
Winsock	Windows Socket. An API for developing Windows programs that can communicate with other machines via the TCP/IP protocol.
WRAP	Wireless Remote Access Platform. A configurable platform optimized for embedded and industrial applications.
$\mu$ Clinux	$\mu$ Clinux is a version of Linux for CPUs without virtual memory or an MMU (Memory Management Unit) and is typically targeted at deeply embedded systems with very little memory or permanent storage.

## Biographies



**Kaj J. Grahn**, Dr. Tech., is presently senior lecturer and research scientist in Telecommunications at the Department of Business Administration, Media and Technology at Arcada Polytechnic, Helsinki, Finland. He is also Program Manager of the Electrical Engineering Programme. His current research interests include Wireless and Mobile Network Security.



Dipl.-Ing. **Christian Dieckert** was born in Frankfurt am Main, Germany on November 18, 1976. After his professional training as a communication electronics technician he entered the University of Applied Science in Frankfurt/Main to study communication engineering. In year 2003 he prepared his degree thesis at Arcada Polytechnic in Helsinki, Finland with the topic “Measurement Data Logging by Using Bluetooth Technology”. After completing his studies in spring 2004 he started working as a customer support engineer at DIEHL Avionic Systems, Germany, where he is responsible for support and maintenance planning as well as technical training for aircrafts high lift systems. Christian’s favourite free time employment is doing sports and travelling.



Dipl.-Ing. **Frank Schneider** was born near Frankfurt am Main, Germany on June 07, 1978. After his school leaving examination in 1997 he started vocational training as an electronics technician. Between September 2000 and March 2004 he studied electrical engineering at the University of Applied Sciences in Frankfurt/Main. Together with his fellow student Christian Dieckert he authored his final thesis “Measurement Data Logging by Using Bluetooth Technology” at Arcada Polytechnic, Helsinki, Finland. Now he works as a junior assistant at the central office for network control in a local power supply company, where he is responsible for networks and communication systems. Living near Frankfurt, his spare time is dominated by sports, travelling and distance learning for a degree in economic science.



**Göran Pulkkis**, Dr. Tech. in 1983 in Digital Systems Laboratory of Helsinki University of Technology, is presently senior lecturer and research scientist in Computer Science and Engineering at the Department of Business Administration, Media and Technology at Arcada Polytechnic, Helsinki, Finland. His current research interests include Network Security and Applied Cryptography