

AmiEs-2009

**8th International Conference and Workshop on
Ambient Intelligence and Embedded Systems**

23 - 25 September, 2009

Funchal, Madeira, Portugal



AmiEs-09

Bluebox

Implementation of a platform independent client software for the GO Bluebox System

Nils T. Kannengießer

and

Thomas Ladehoff

Faculty of Computer Science and Electrical Engineering

Kiel University of Applied Sciences

Kiel, Germany

Thorsten Knutz

GO Systemelektronik

Kiel, Germany

Helmut Dispert

Faculty of Computer Science and Electrical Engineering

Kiel University of Applied Sciences

Kiel, Germany

Bluebox

Master Project 2nd Semester Information Technology

Nils T. Kannengießer

and

Thomas Ladehoff

Faculty of Computer Science and Electrical Engineering

Kiel University of Applied Sciences

Kiel, Germany

Program is strongly oriented towards Research and Development.

Students are from the beginning of their study program involved in R&D.

1st and 2nd Semester: lecture module with group-based R&D project

3rd Semester: 50 % research

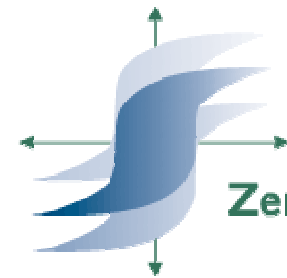
4th Semester: 100% research (Thesis)

Master Project 2nd Semester → approx. 2,5 credit points (ECTS)

Bluebox

In co-operation with

- **Center for Adaptronics,
Kiel University of Applied Sciences**

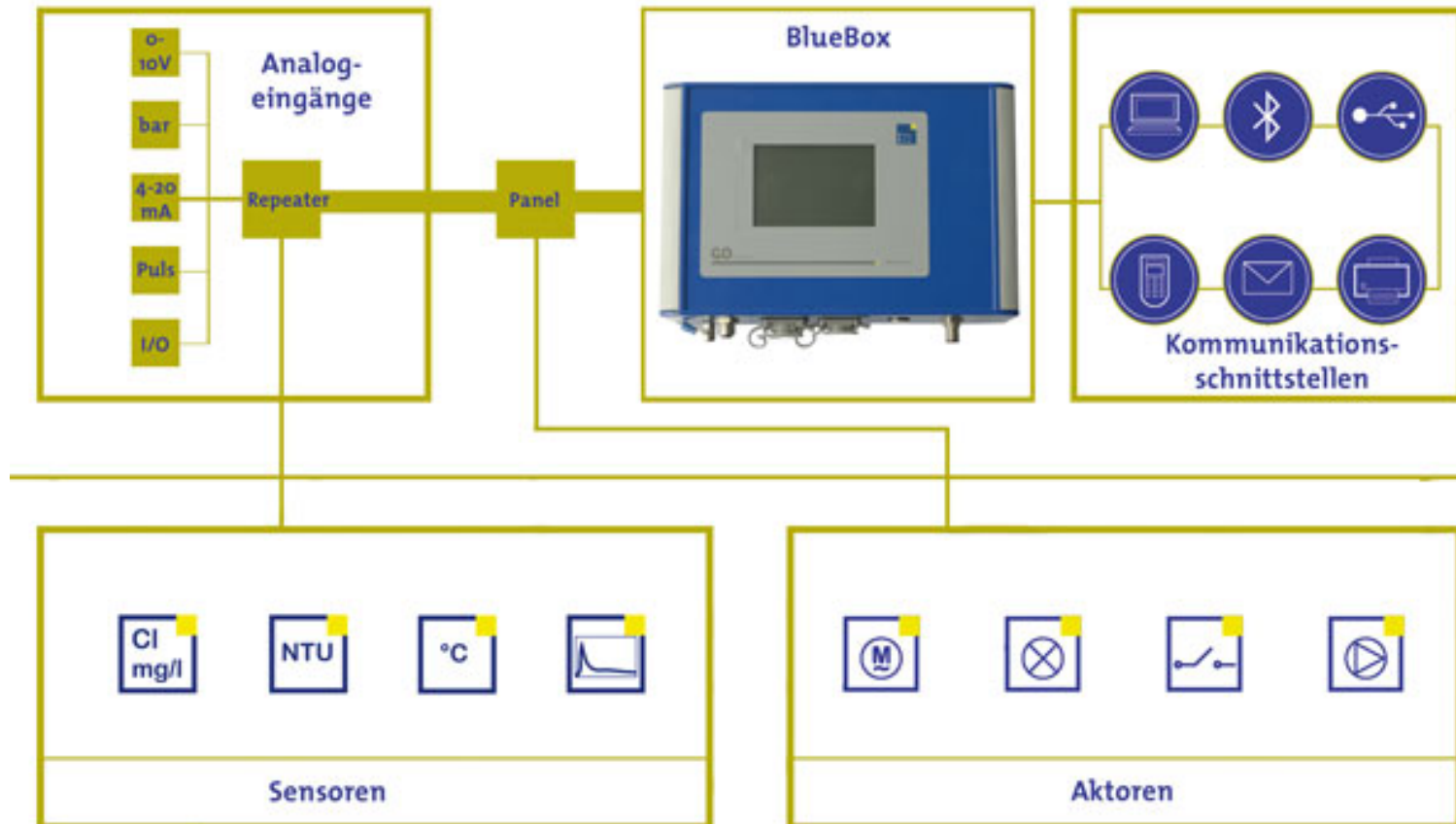


Zentrum für Adaptronik

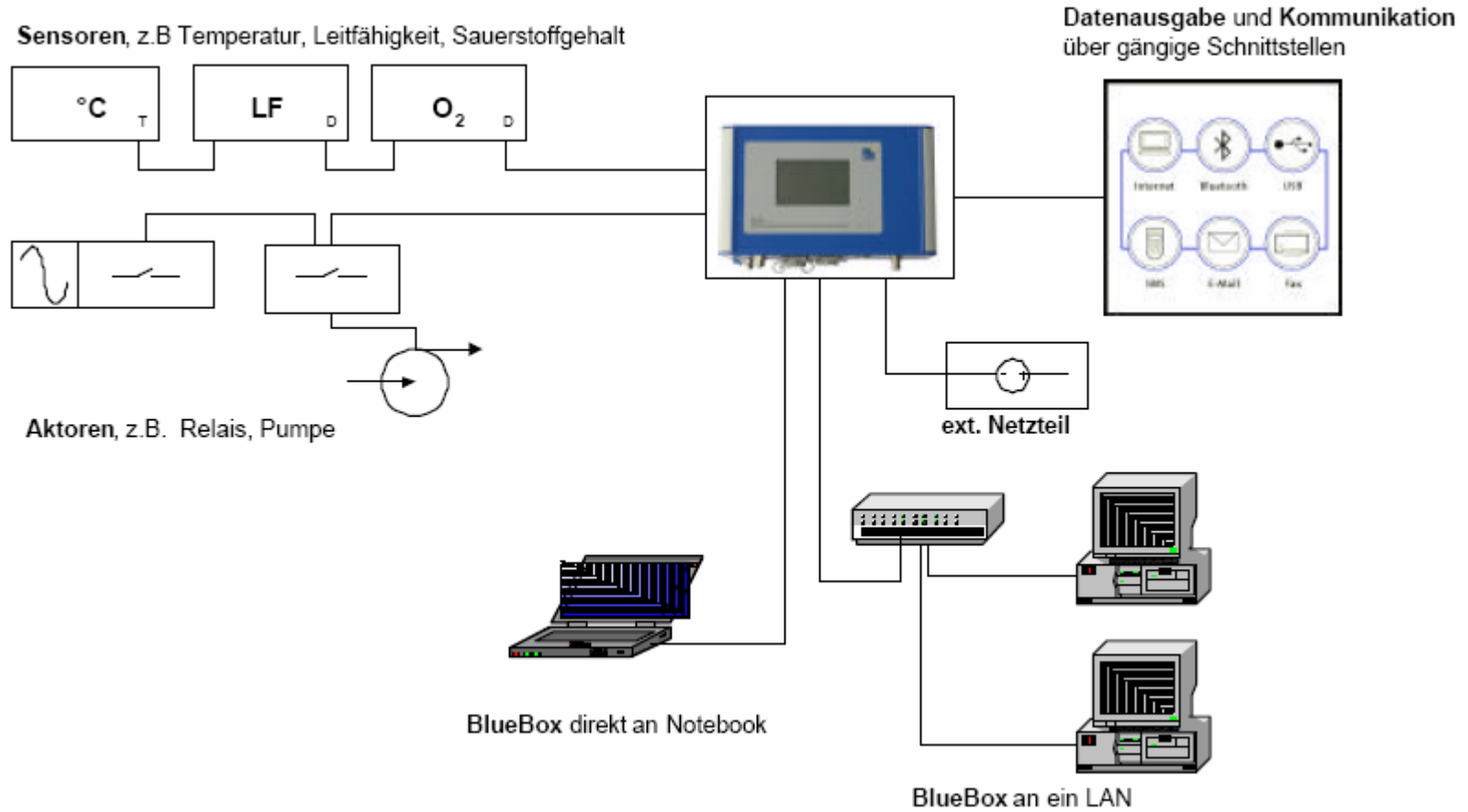
- **GO Systemelektronik**



Bluebox



Bluebox



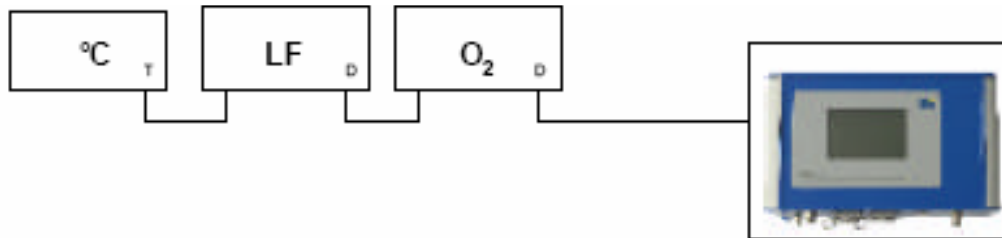
Bluebox

BlueBox sensor parameters

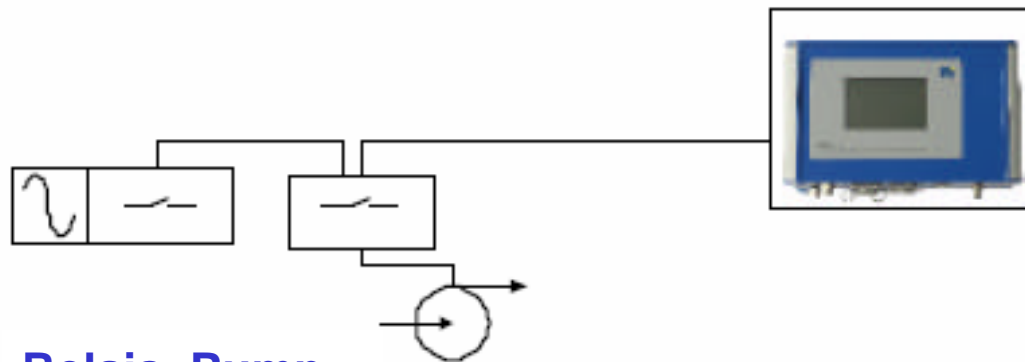
- **Conductivity**
- **Temperature**
- **Oxygen**
- **pH-Value**
- **optical Sensors**
- **Level / Pressure**
- **Redox Potential**

Bluebox

Sensors:
Temperature, Conductivity, Oxygen content



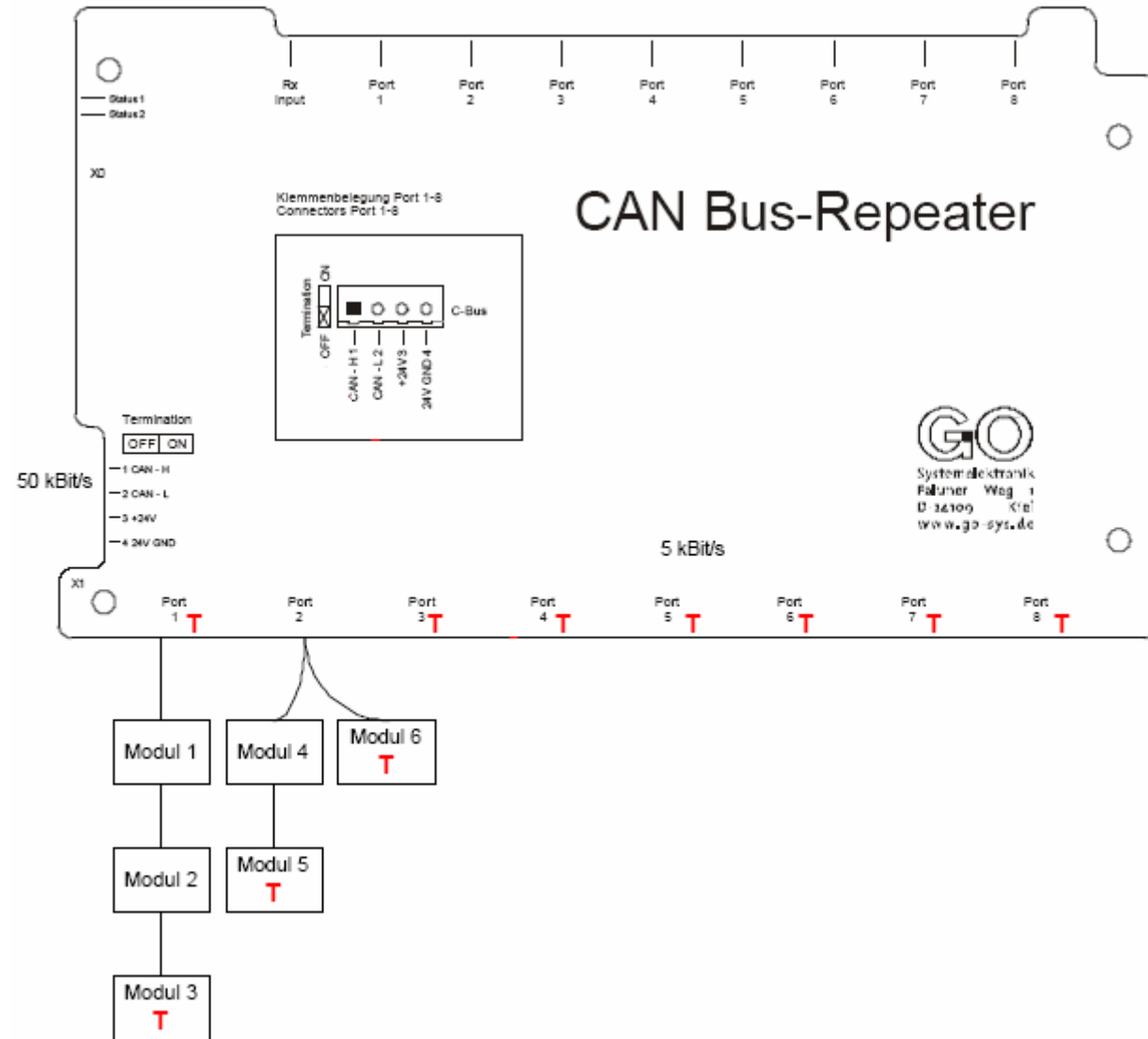
Sensors



Actuators

Relais, Pump

Bluebox



Bluebox

Problem:

**No possibility for easy Internet access
and graphical visualization.**

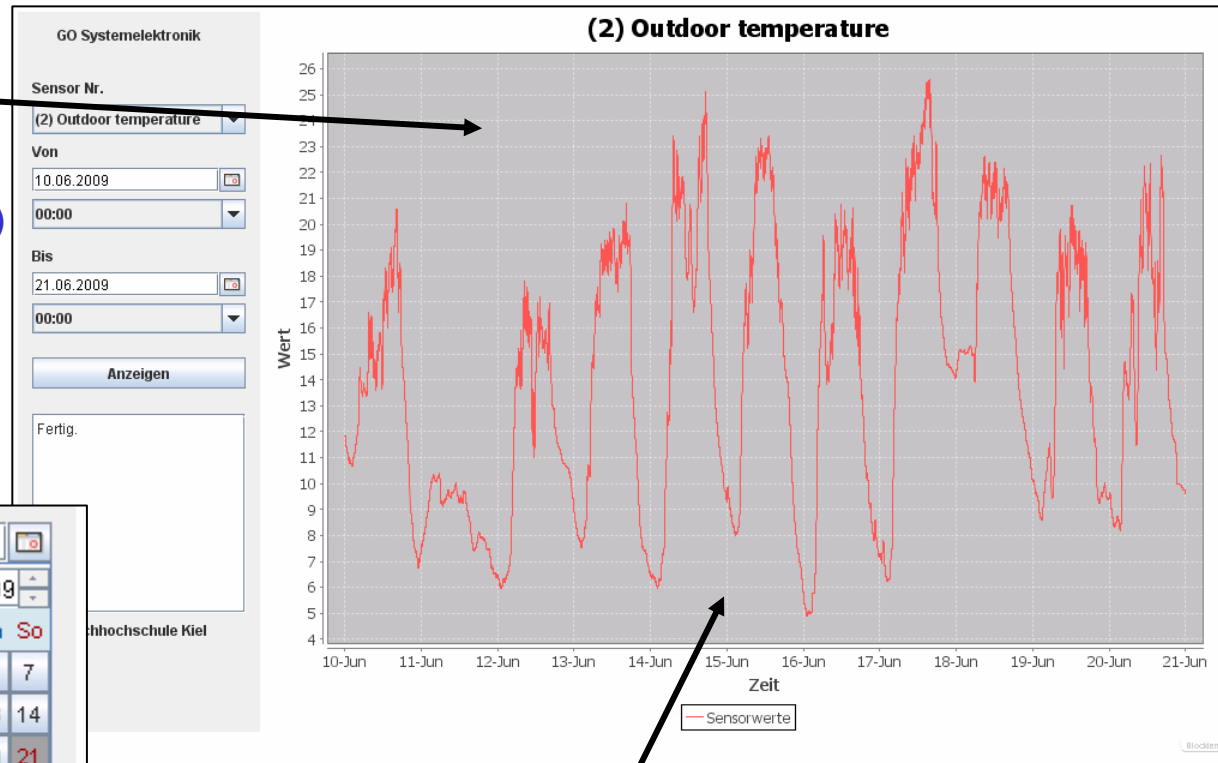
Solution:

Java Applet for the BlueBox

Java Applet – Requirements

Control

- Sensor
- From (TT-MM-YYYY HH-00)
- To (TT-MM-YYYY HH-00)
- Display Data
- Output Status

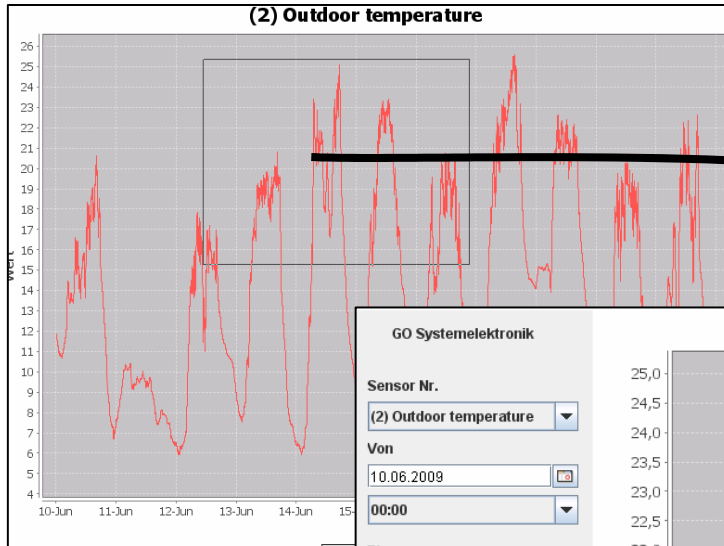


jCalendar

Output (via Jfreechart)

Java Applet – Requirements

JFreechart Funktionen



GO Systemelektronik

Sensor Nr.
(2) Outdoor temperature

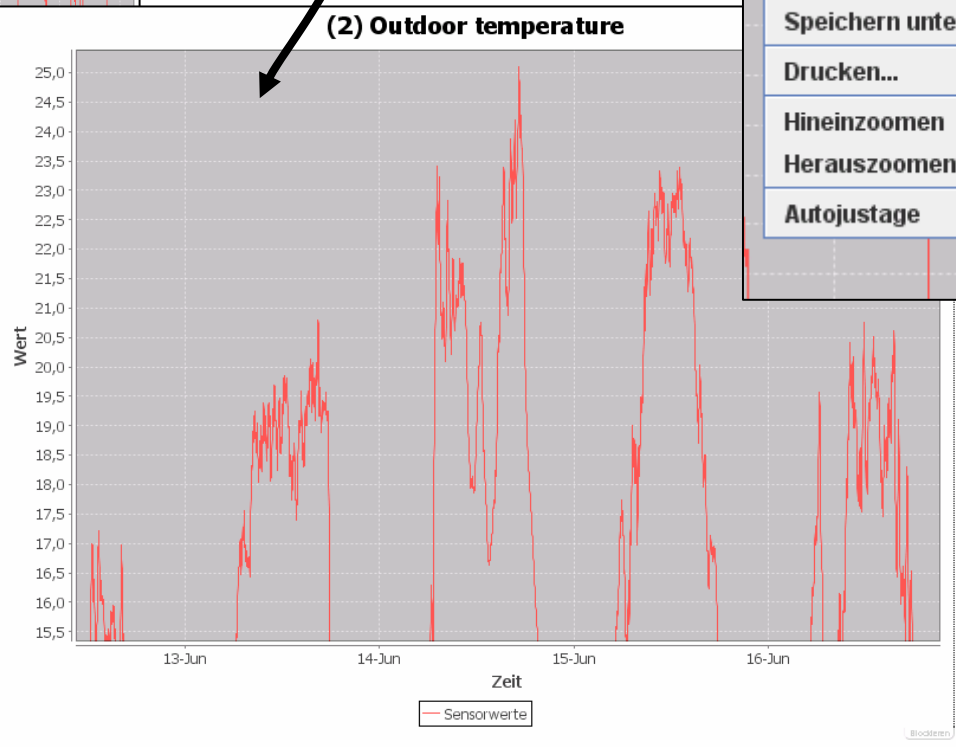
Von
10.06.2009 00:00

Bis
21.06.2009 00:00

Anzeigen

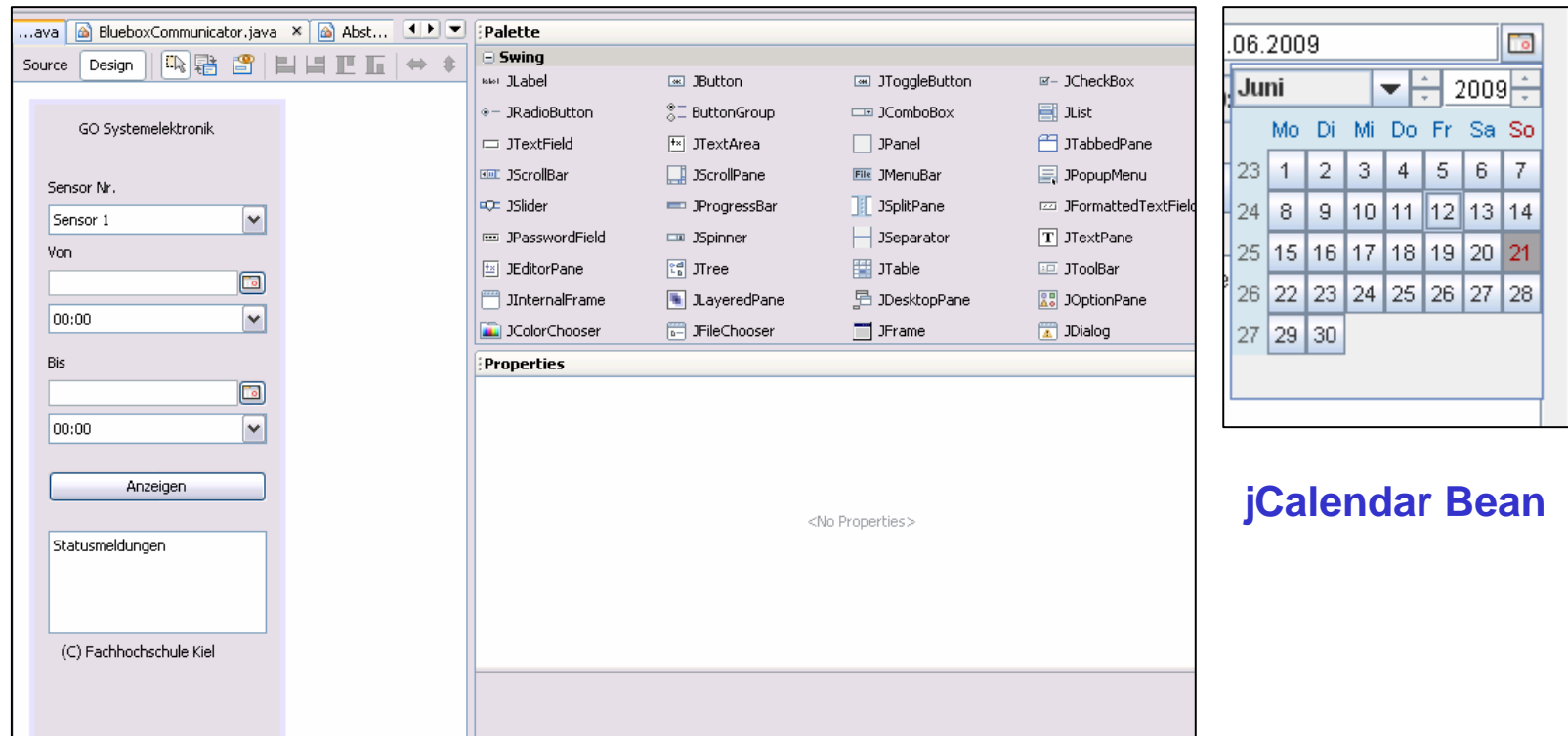
Fertig.

(C) Fachhochschule Kiel



Java Applet – Requirements

Control Section implemented using NetBeans Form Editor



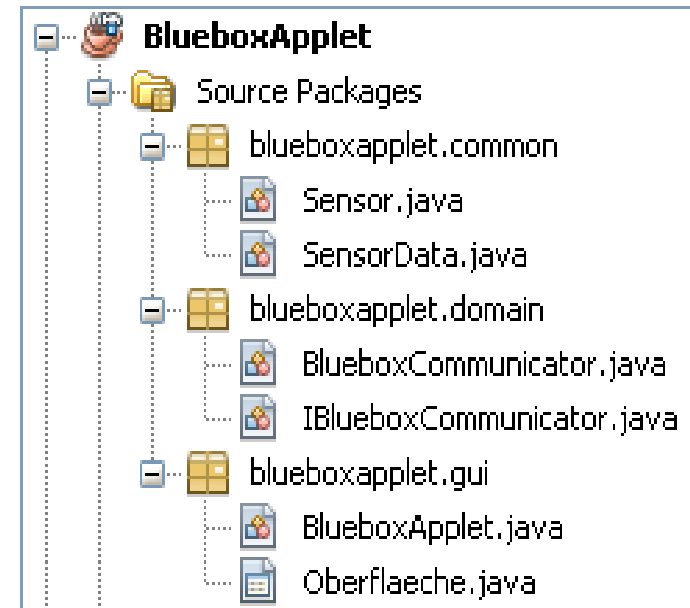
jCalendar Bean

Applet Structure

Structure of the applet

The applet is divided into three layers (Java packages) to achieve a clear logical separation between the different program parts.

- **blueboxapplet.gui**
Implementation of the Graphical User_Interface.
- **blueboxapplet.domain**
Implementation of the processing tasks (e.g. analysis of sensor data) and the communication with the BlueBox
- **blueboxapplet.common**
Classes that are used in the other layers.

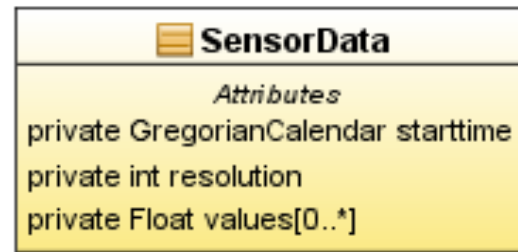
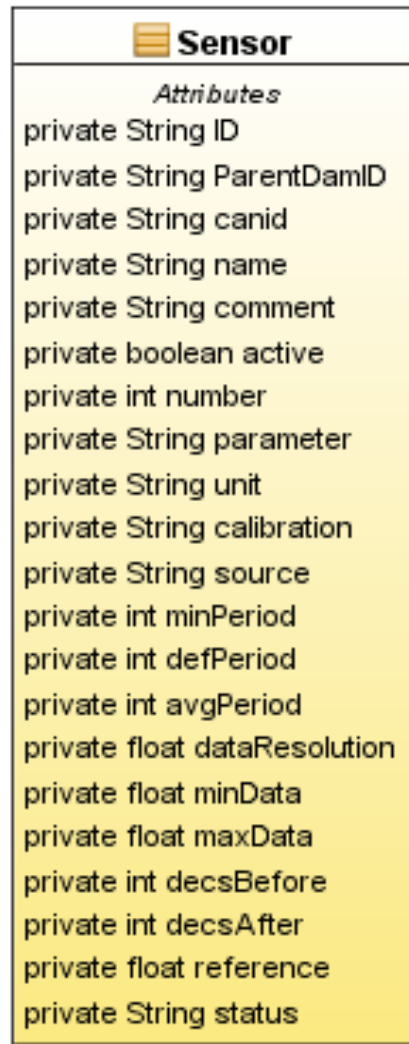


Applet Structure

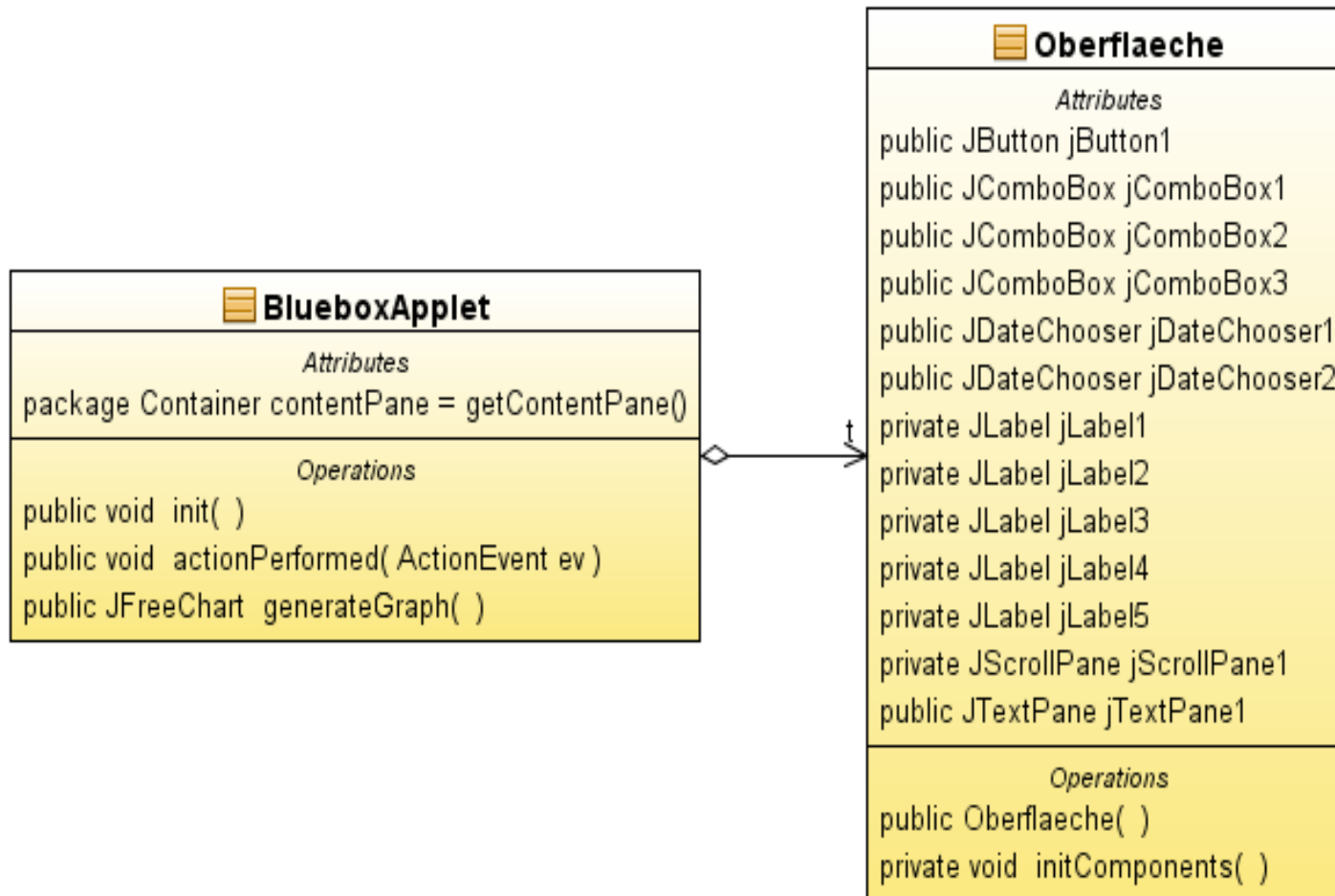
Receiving and processing the data

Sending and receiving data to and from the server is done by a generic method, named *getServerResponse*. It takes the request string as an argument and returns the received data in a list of strings (Java class *ArrayList*). Generally each element in the list represents the string of one data packet received from the server.


Applet Structure – Sensor Classes



Applet Structure – GUI Classes



Applet Structure – Processing Classes

|  BlueboxCommunicator |
|---|
| <i>Attributes</i> private String host private int port private int maxtimespan private Socket socket |
| <i>Operations</i> public BlueboxCommunicator(Applet applet) public BlueboxCommunicator(String host, int port) private String getMessage() public String[0..*] getServerResponse(String request, String endcondition) |
| <i>Operations Redefined From IBlueboxCommunicator</i> public String initConnection() public void closeConnection() public Sensor[0..*] getSensorList() public SensorData gdb(Sensor sensor, GregorianCalendar starttime, GregorianCalendar endtime) public int getMaxtimespan() public void setMaxtimespan(int maxtimespan) |

Applet Structure – Processing Classes

Receiving and processing the data

Sending and receiving data to and from the server is done by a generic method, named *getServerResponse*.

It takes the request string as an argument and returns the received data in a list of strings (Java class *ArrayList*). Generally each element in the list represents the string of one data packet received from the server.

This method *getServerResponse* is called by the methods, which analyze the data of the server:

- *getSensorList*
Receiving the list of sensors from the database
- *gdb*
Receive the data of a sensor (same name as the Bluebox command)

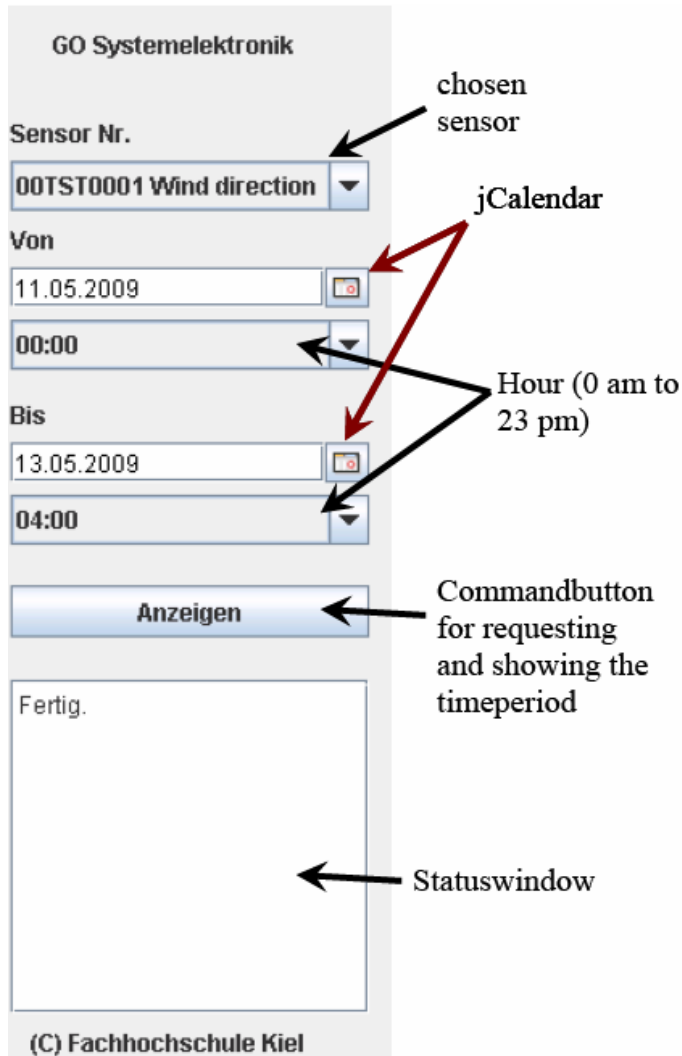
For test purposes:

A BlueBox simulator has been implemented

Bluebox – Java Commands

| | |
|-------------------------|---|
| getserialno | BlueBox serial number |
| getstarttime | Get database starttime(GMT) |
| getsensorno | Number of sensors |
| resetdam | Set DAM pointer to first DAM |
| getdam | Show next DAM info |
| resetsensor | Set sensor pointer to first sensor |
| getsensor | Show next sensor info |
| getsensordata | Show sensor data |
| getadamnr | Number of actuator DAMs |
| getadam | Show next actuator DAM info |
| getactuatornr | Number of actuators |
| getactuator | Show next actuator |
| gettime | Get BlueBox date & time |
| getposition | Get GPS or GEO position |
| getstatus | Get BlueBox status information |
| gettimeserver | Get NTP status |
| password(pw) | If password is required |
| gdb(...) | Get database entries |
| getchangelog(..) | Get changelog entries |
| quit | Close connection |

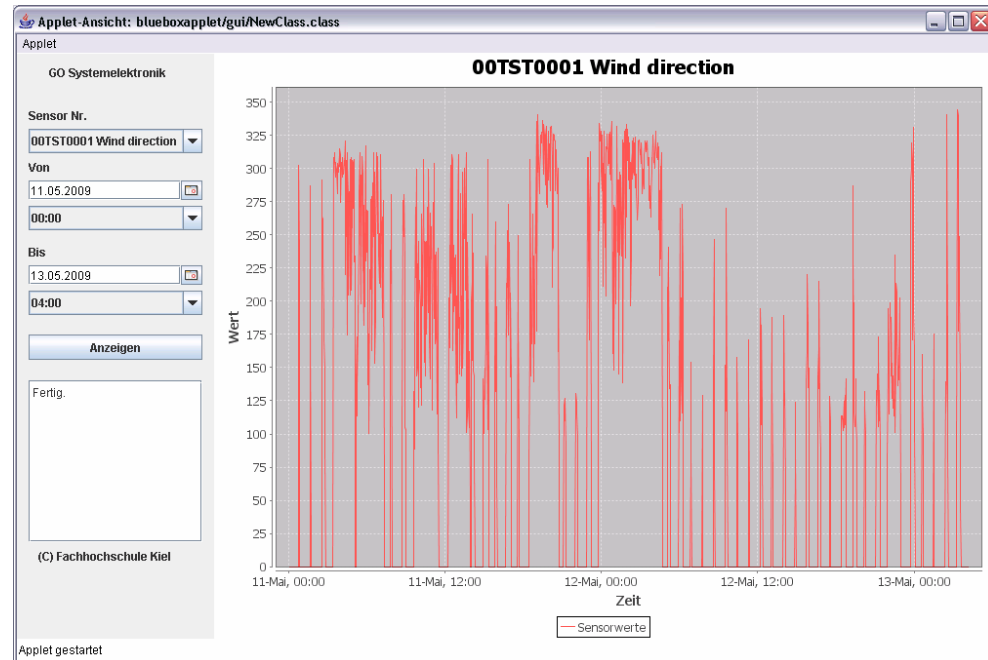
Bluebox



The GUI

The GUI is divided into two parts. On the left side there are the options and controls and on the right side the generated diagram (or nothing at start-up).

The left part is created by using the Netbeans Form Editor for easy adjustment of controls in further applications. The sensor names for the drop down-box are generated just in time, when starting the applet by fetching the sensor list from the server.



Bluebox

Thank you!

Now Texas A&M University
College Station, Tx, US.A.

Contacts:

Nils T. Kannengießer
and

Thomas Ladehoff

Faculty of Computer Science and Electrical Engineering
Kiel University of Applied Sciences
Kiel, Germany

Thorsten Knutz

GO Systemelektronik
Kiel, Germany

Helmut Dispert

Faculty of Computer Science and Electrical Engineering
Kiel University of Applied Sciences
Kiel, Germany