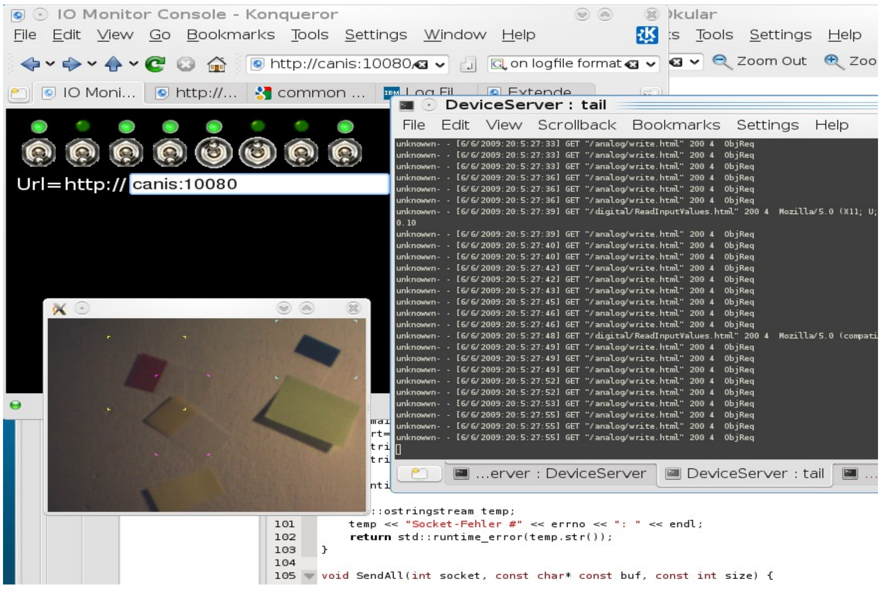
The DeviceServer API

In the following screenshot you can see the Object Recognition and the Web 2.0 AJAX API in action with the DeviceServer.



IO Monitor Console - Konqueror
File Edit View Go Bookmarks Tools Settings Window Help
http://canis:10080/
on logfile format
IO Moni… http://… common … Log Fil Extende
Url=http:// canis:10080
DeviceServer : tail
File Edit View Scrollback Bookmarks Settings Help
…erver : DeviceServer
DeviceServer : tail
Zoom Out

```
unknowwn- - [6/6/2009:20:5:27:33] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:33] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:33] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:36] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:36] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:36] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:39] GET "/digital/ReadInputValues.html" 200 4  Mozilla/5.0 (X11; U;
0.10
unknowwn- - [6/6/2009:20:5:27:39] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:40] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:40] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:42] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:42] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:43] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:45] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:46] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:46] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:48] GET "/digital/ReadInputValues.html" 200 4  Mozilla/5.0 (compati
unknowwn- - [6/6/2009:20:5:27:49] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:49] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:49] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:52] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:52] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:53] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:55] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:55] GET "/analog/write.html" 200 4  ObjReq
unknowwn- - [6/6/2009:20:5:27:55] GET "/analog/write.html" 200 4  ObjReq
```

```
::ostringstream temp;
    temp << "Socket-Fehler #" << errno << ": " << endl;
    return std::runtime_error(temp.str());
}

void SendAll(int socket, const char* const buf, const int size) {
```

Abb1.png „Browser and Object Recognition“

The Deviceserver provides the following addresses in the API Version 0.2 :

1. /analog/read.html
2. /analog/write.html
3. /digital/ReadInputValues.html
4. /digital/ReadOutputValues.html
5. /digital/WriteOutputValues.html
6. /digital/WriteInputValues.html

What you need to know and understand is the concept of the IO Group in the DeviceServer. The DeviceServer Groups every 8 Bits (a port) to a IO Group. So the inputs 1-8 of the DeviceServer corresponds to the IO Group 1, 9-16 are IO Group 2 and so forth. The same is done with the outputs. The reason for this is, that most I/O Hardware is port oriented.

Syntax and Parameter of the Adresses:

/digital/ReadInputValues.html

Parameter: IO Group (1- 16)
Beispiel: http://canis:10080/digital/ReadInputValues.html?2[,3,4]
Ergebnis: <html><body> 1 0 0 1 1 1 1 0 </body></html>
Erläuterung: Shows the IO Group 2 corresponding to the inputs I9 bis I16. The result is shown as a minimal HTML page. If you send more than one IO Group separeted by comma, you receive the coressponding values, e.g if you send 3 IO Groups you get 24 Bits returned.

/digital/ReadOutputValues.html

Parameter: IO Group (1- 16)
Beispiel: http://canis:10080/digital/ReadOutputValues.html?2[,3,4]
Ergebnis: <html><body> 1 0 0 1 1 1 1 0 </body></html>
Erläuterung: Shows the IO Group 2 corresponding to the outputs O9 bis O16. Everything is the same as with the inputs before.

/digital/WriteInputValues.html

Parameter: IO Group (1 – 16), Value
Beispiel: http://canis:10080/digital/WriteInputValues.html?2,255
Ergebnis: <html><body> 1 1 1 1 1 1 1 1 </body></html>
Erläuterung: Writes the value to the input IO Group, as result the value written to the IO Group is returned in binary.

/digital/WriteOutputValues.html

Parameter: IO Group (1 – 16), Value
Beispiel: http://canis:10080/digital/WriteOutputValues.html?2,255
Ergebnis: <html><body> 1 1 1 1 1 1 1 1 </body></html>
Erläuterung: Same as before for the Outputs.

/analog/read.html

Parameter: IO Group (1- 8)
Beispiel: http://canis:10080/analog/read.html?1
Ergebnis: <html><body> 120 89 0 0 0 0 0 0 </body></html>
Erläuterung: Reads the IO Group 1 corresponding to the analog Inputs A1 bis A8, the result is shown as minimal html page.

/analog/write.html

Parameter: IDX (1 – 64),Wert1,Wert2,...,Wertn
Beispiel: http://canis:10080/analog/write.html?1,120,89
Ergebnis: <html><body> </body></html>
Erläuterung: The parameter IDX is the number of the first analog input where we start writing the values; the parameters Wert1 – Wertn are the values written to the analog inputs starting with input number IDX.In this example the values 120 and 89 are written to the analog inputs A1 and A2.

One possible cause of trouble can be, depending on the used programming language, the setting of the User Agent. During programming the webserver I noticed that I need to send the answer in different ways depending on the browser. E.g. Firefox and Mozilla need a different way than konqueror or wget.

In case of trouble try setting the User Agent to Mozilla. The HTTP client from the Gambas Package gives the error 1018 in standard settings. If you set the User Agent to „Mozilla“ everything works fine.