

Web-based Control Application Using WebSocket



Y. Furukawa
SPring-8, Hyogo 679-5148, Japan

furukawa@spring8.or.jp

Introduction

- Web-based application has many advantages
 - Platform independent
 - Easy to develop and many developers
 - Easy to keep running application newest
- Question is “Are control applications able to be written as web-based application?”
- Answer before HTML5 is “Partially Yes”.
 - Periodically Polling type application can be realized such as alarm display.
- What is lacks in web application?
 - Notification from server side to the client.
 - To solve this situation, long polling method, called “Comet” is introduced but there were many limitations.
 - Require open/close session for every message exchange.
 - Cannot send another message from client during waiting for returned message.

What is WebSocket?

- WebSocket a part of HTML5 and it brings asynchronous and full-duplex communication with server and client.
- WebSocket protocol is built-in most modern web-browsers and you can easy to open in a javascript, like
 - `ws = WebSocket(“ws://<host>[:port]”);`
 - `ws.send(“<message>”);`
 - `ws.onmessage() = function(event){...};`
- Web applications are written in Javascript most cases and easy to apply to message based control system like MADOCA.

WebSocket to MADOCA gateway

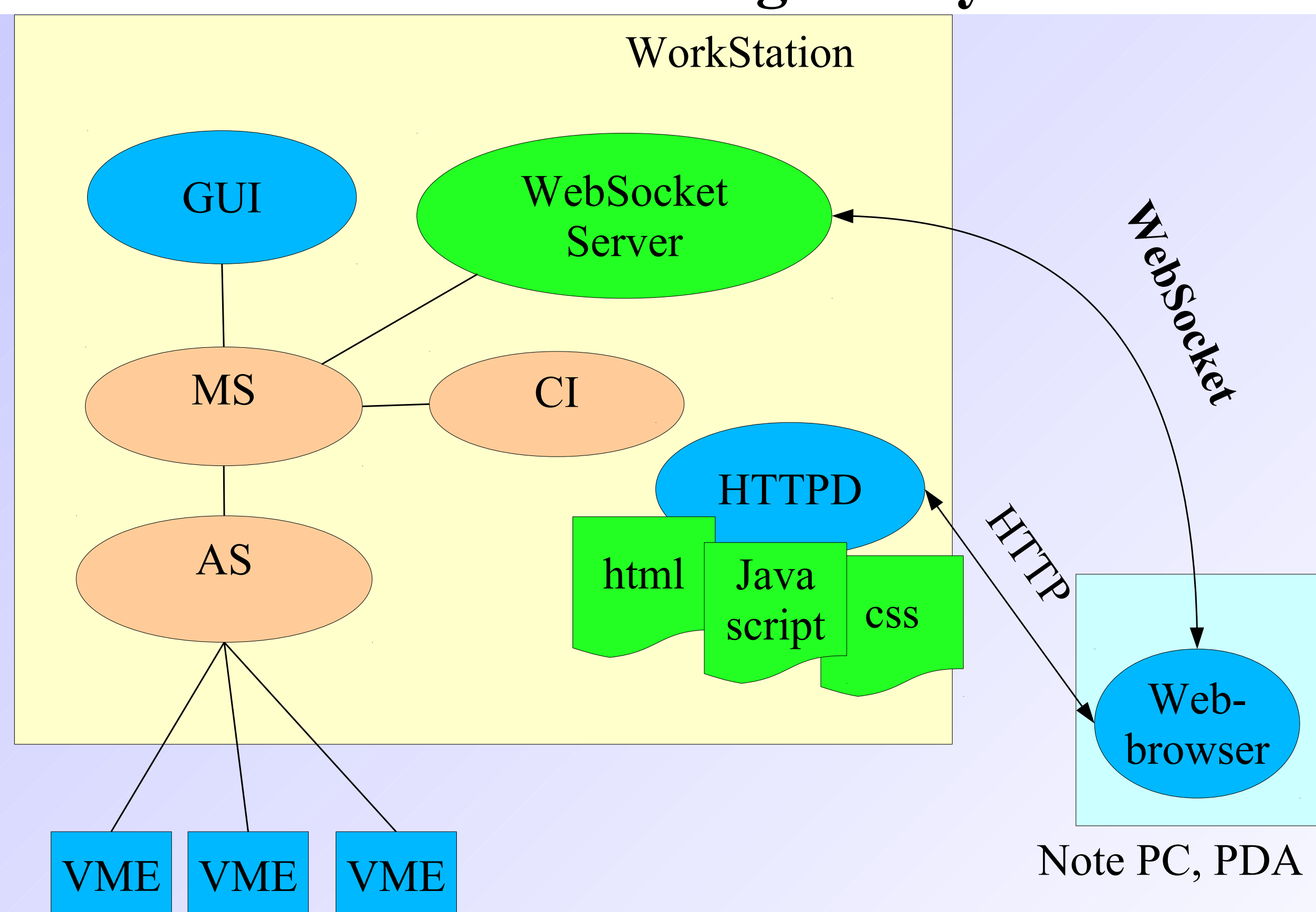
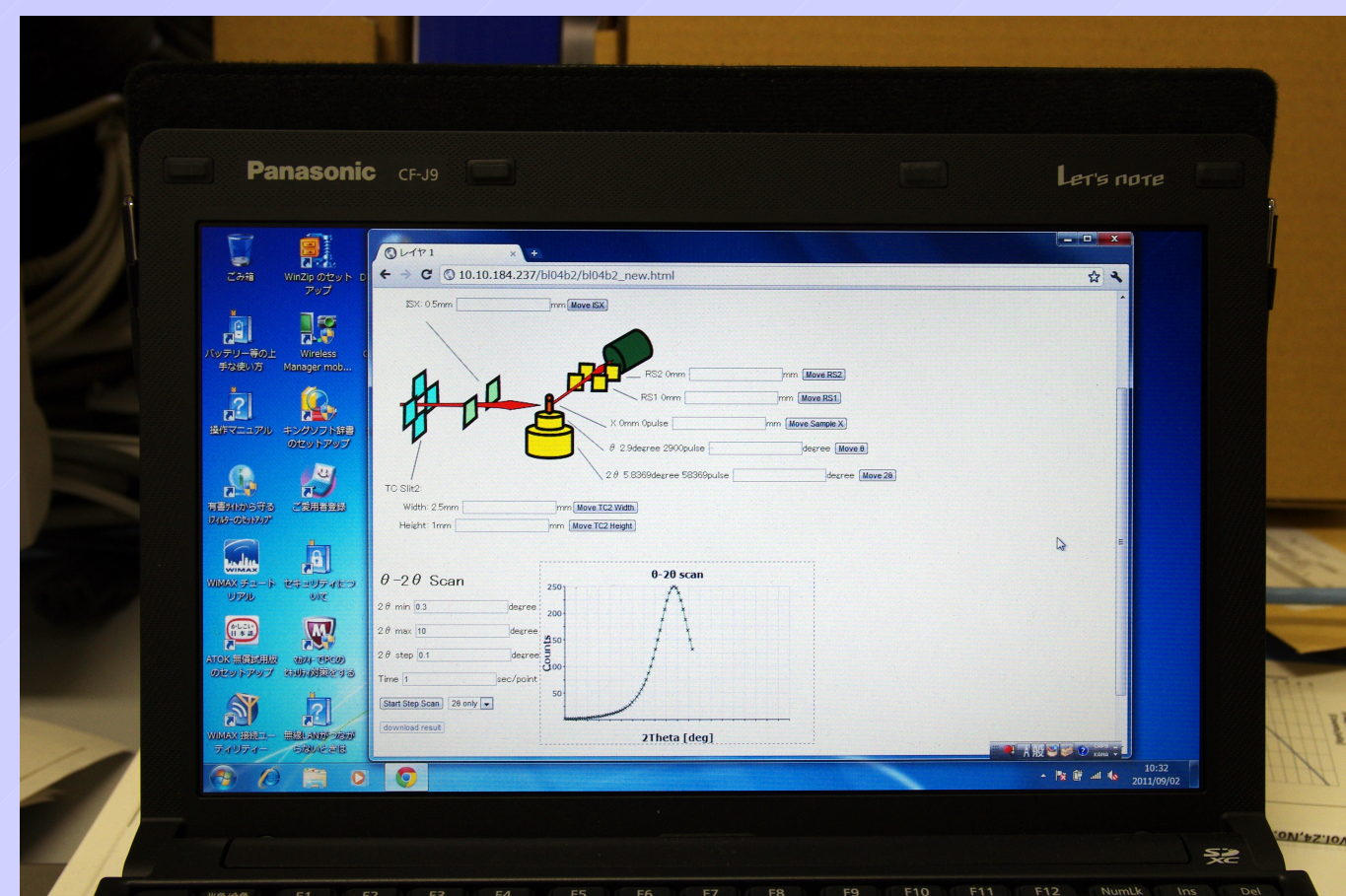
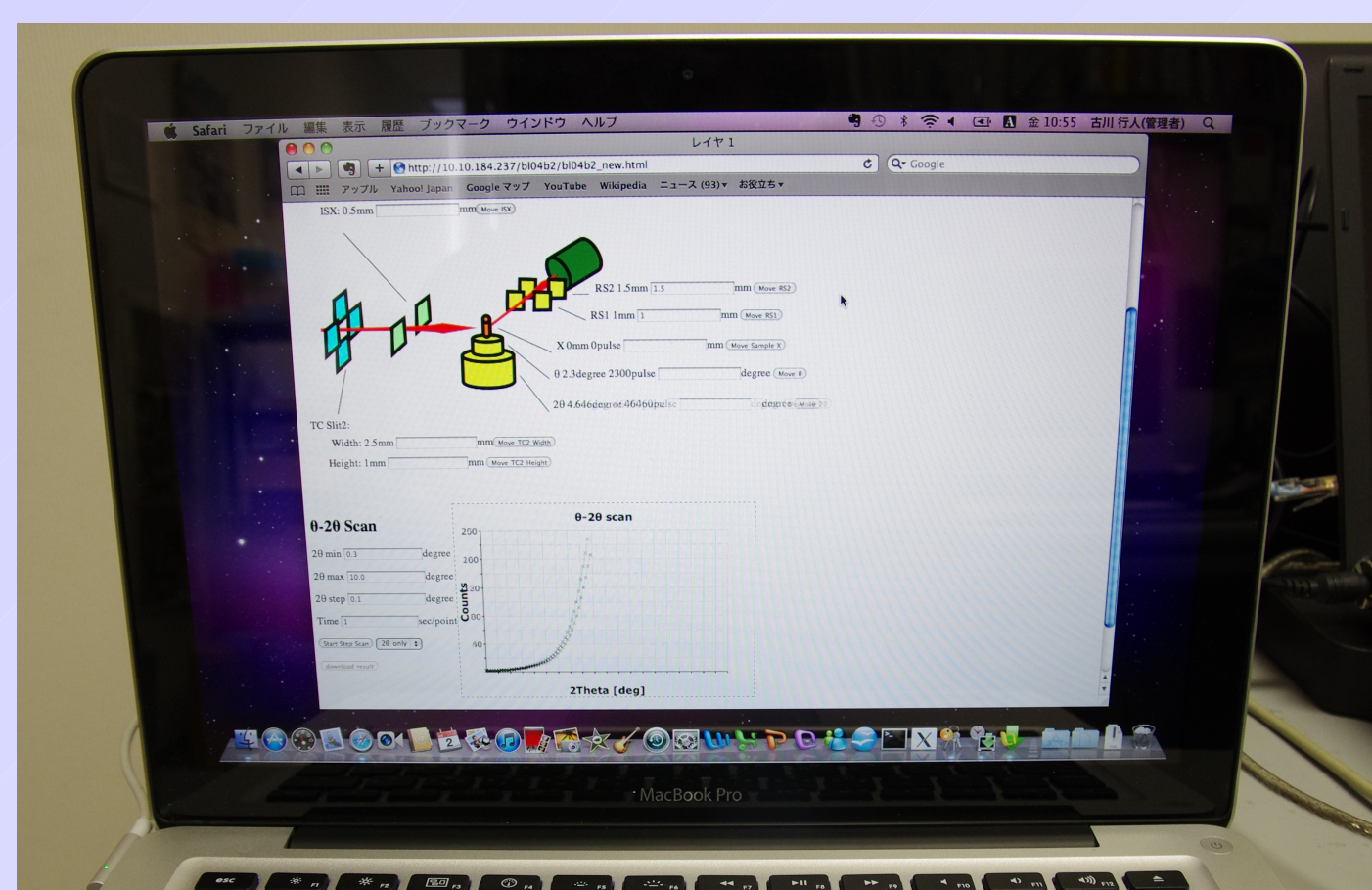


Figure 1: A schematic drawing of the WebSocket and MADOCA control system. Orange faced component, MS (Message Server), CI (Command Interpreter) and AS (Access Server) are standard MADOCA components. Green faced components are newly developed for the WebSocket Application.

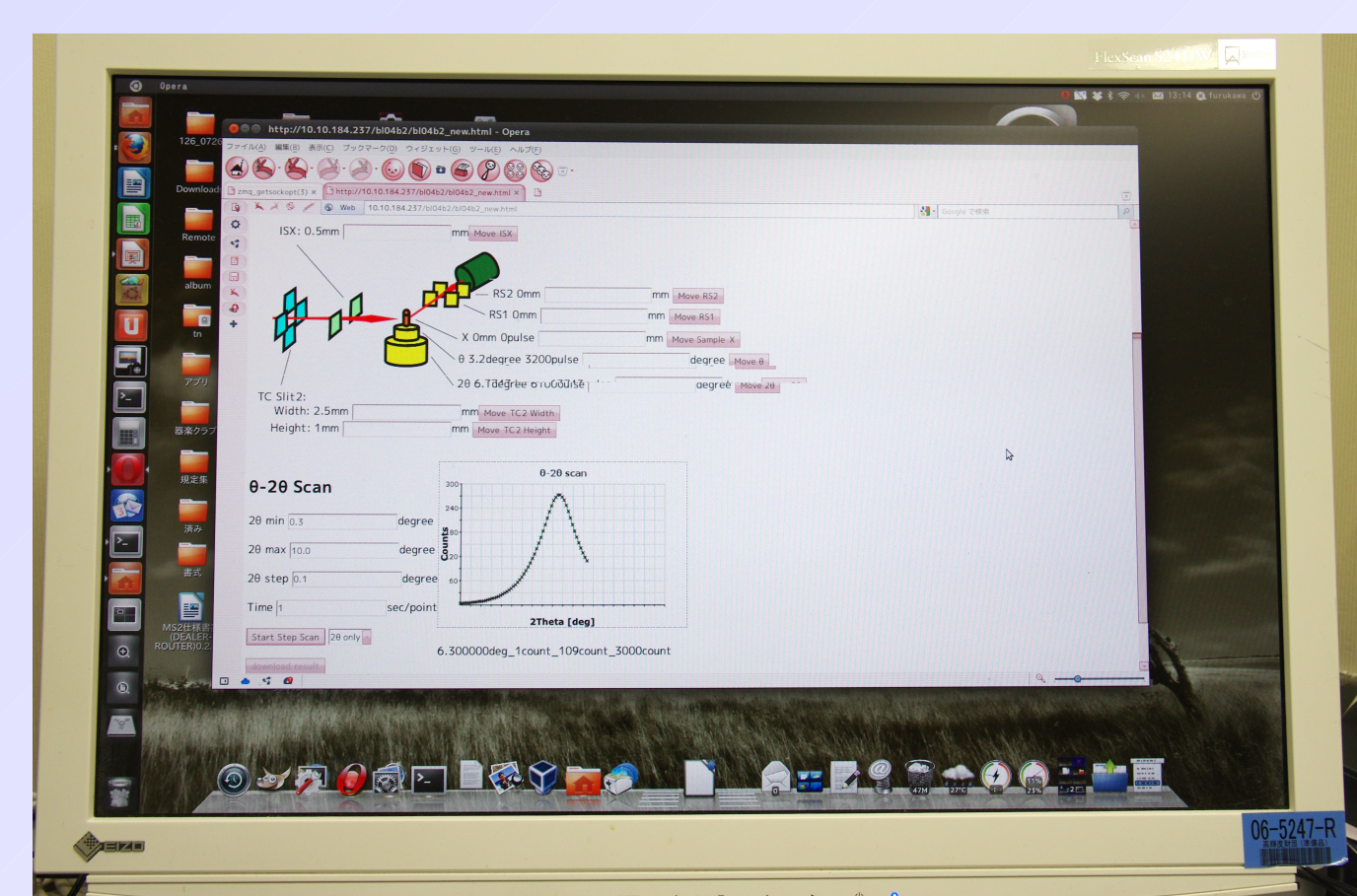
Available Platforms and Clients



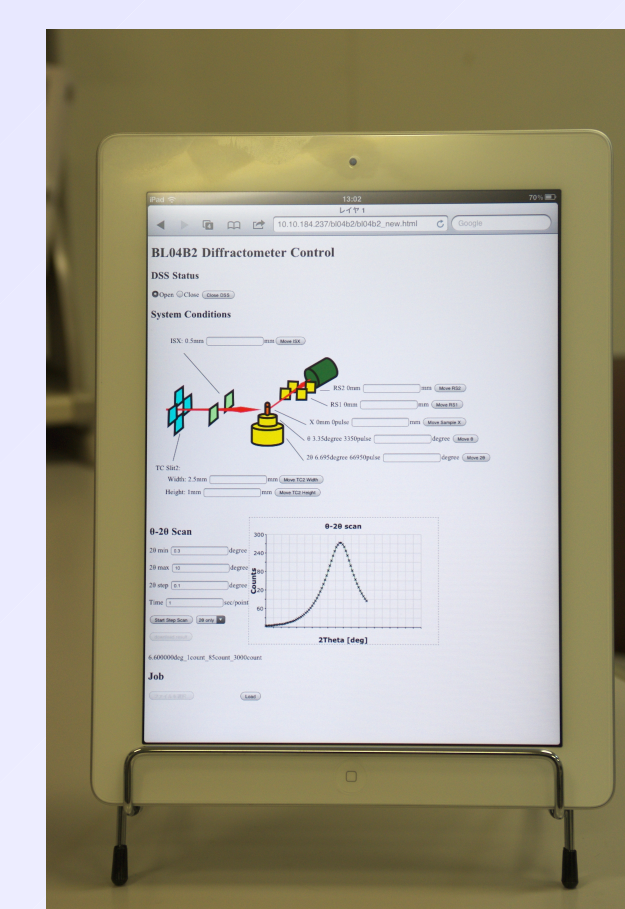
Google Chrome 13.0 on Windows7



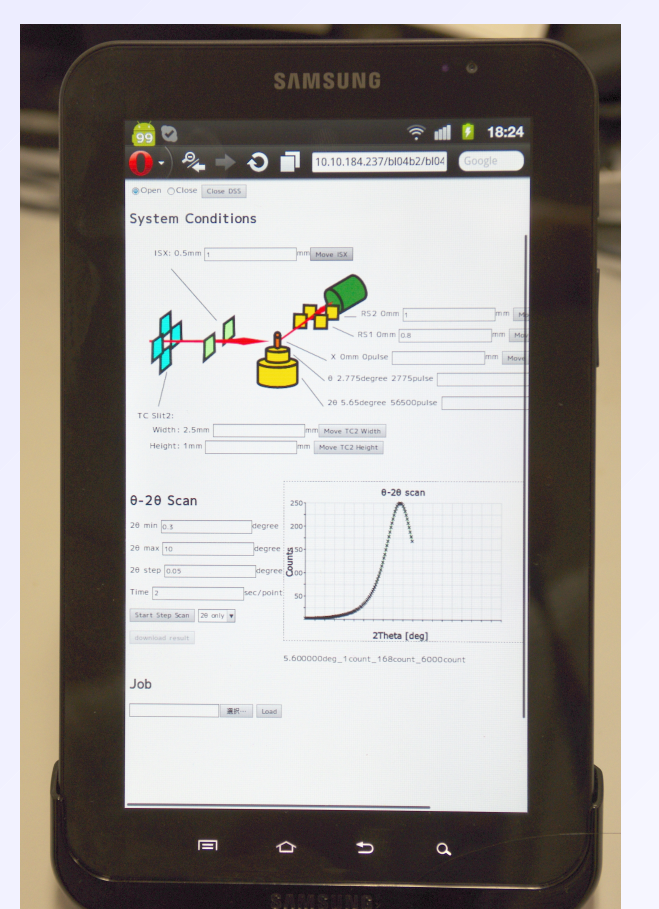
Safari 5.0.5 on MacOSX Snow Leopard (10.6)



Opera 11.5 on Ubuntu 11.04



Safari on iPad2



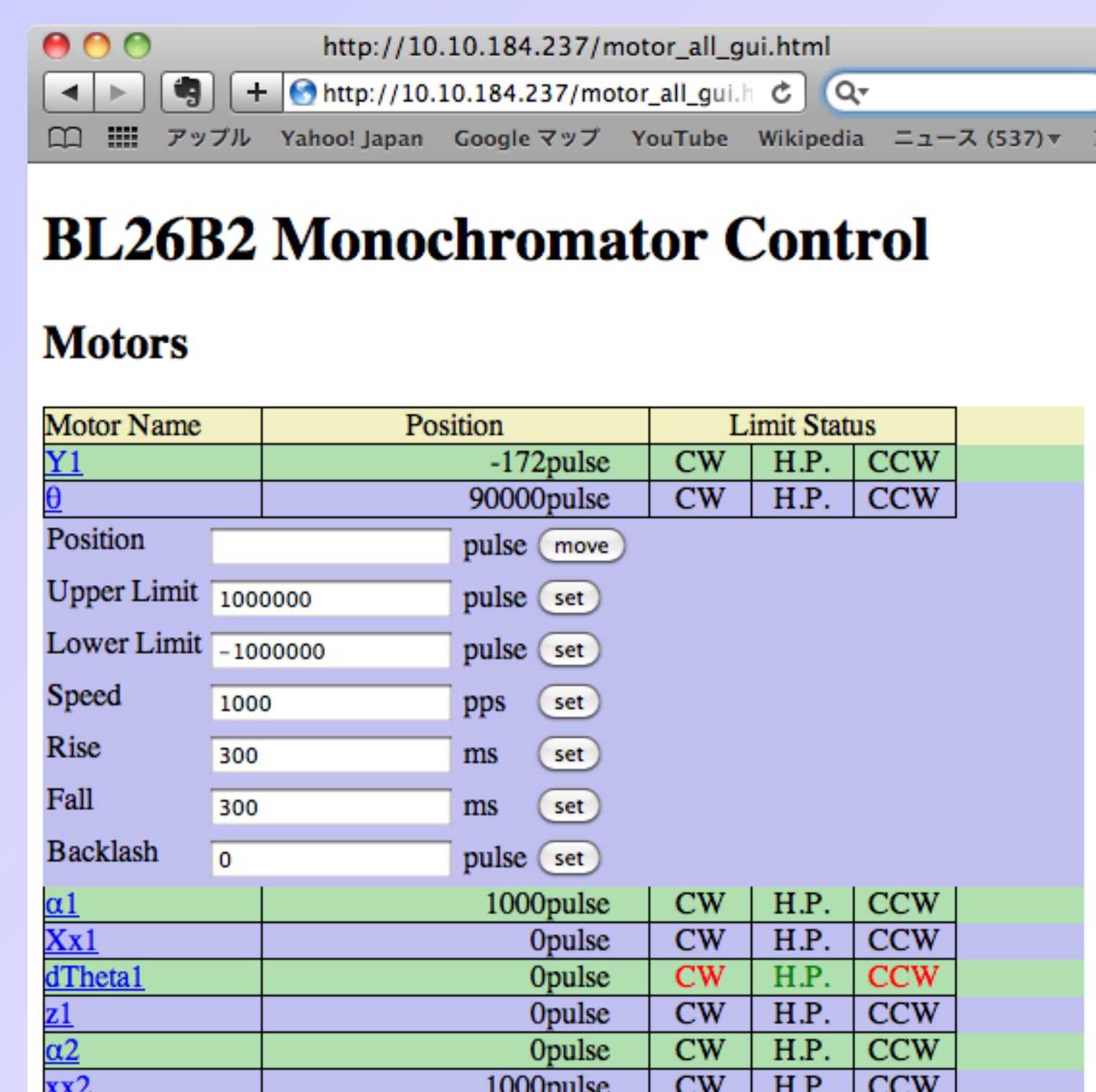
Opera Mobile on Galaxy Tab

	Windows	MacOSX (10.6)	Linux (Ubuntu)	iOS	Android (2.3.3)	Note
Google-Chrome	○	○	○			
Safari		○		○ (iPad)		
Firefox (4~)	○	○	○			WebSocket is disabled by default
Opera (10.7~)	○	○	○		○	WebSocket is disabled by default

Not only PC or Workstation OS but mobile device OSs are available for the platform of the WebSocket support web-browsers. WebSocket based control applications can be applicable for the applications running in the central control room and applications for the handy terminal which is useful for tuning, testing and trouble shooting of equipment. WebSocket is also applicable for the wide-area remote experiment system (please refer the presentation THBHAUST05 on Thursday morning) because it is easy to maintain the application on the remote user latest.

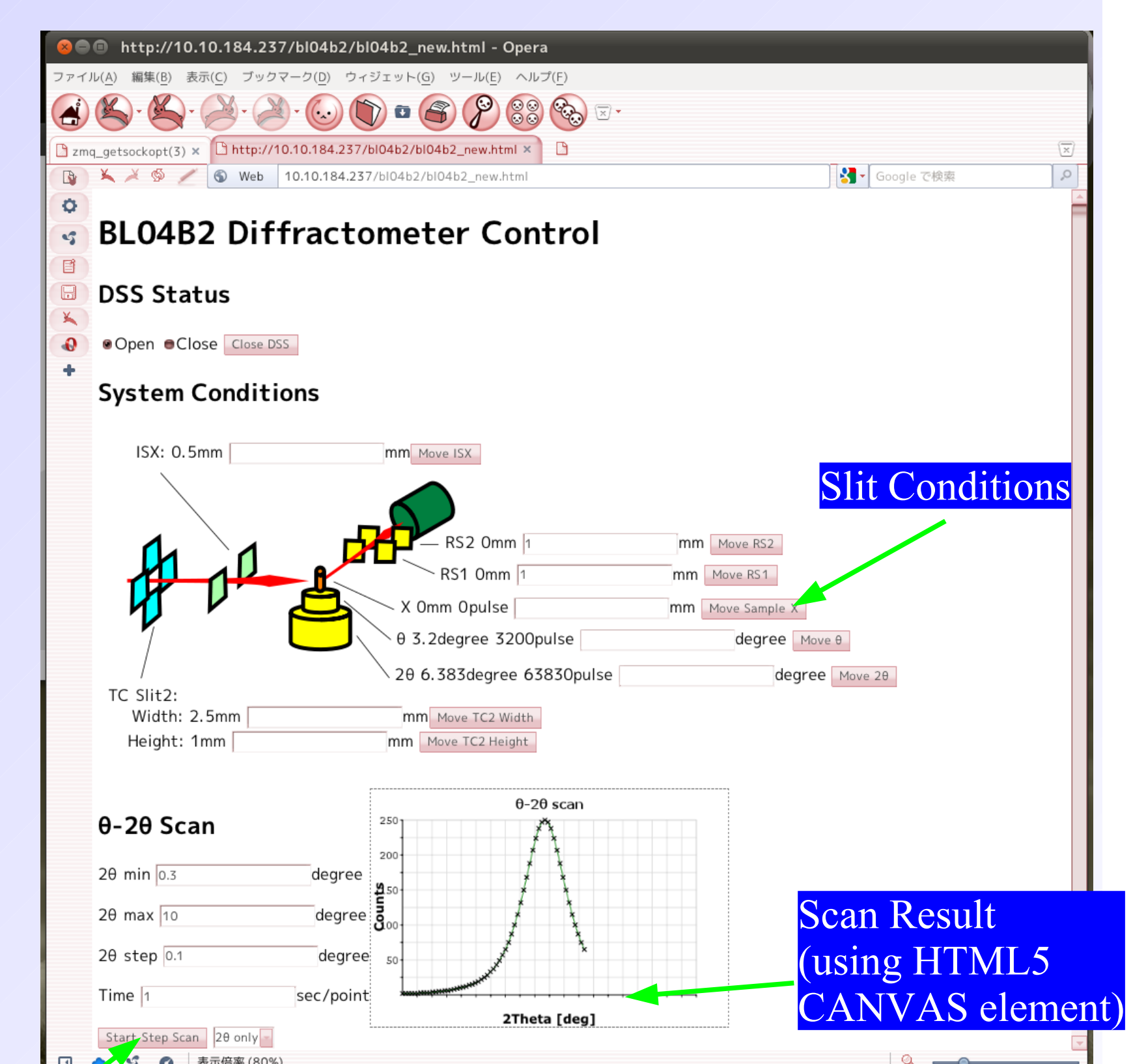
Examples

Stepper Motor Control



You can monitor stepper motor position and can display/hide parameter entries by clicking motor name. Displaying/Hiding mechanism is realized by cascading style sheets.

Diffractometer Control



Scan Parameters

You can change slit conditions and make a scan to obtain diffraction data. This web page is designed using BlueGriffon (<http://bluegriffon.org>).

Note on WebSocket Version

The WebSocket protocol is under development. Latest version is “hybi-10” which is opened to the World in this July and expected to be a final version. My WebSocket server is not based on hybi-10 but on hybi-00 because currently most browser’s implementations are based on the hybi-00 (Latest Google Chrome and Firefox’s WebSocket implementations are based on newer version). There are reported that the hybi-00 contains a security issue, so the Opera browser and Firefox’s WebSocket are turned off by default. You have to turn on the WebSocket using “about:config” dialog.

Conclusion

- Answer for the question “Are control applications able to be written as web-based application?” become now “YES” with the WebSocket.
- WebSocket based application will be used widely on control applications.