```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:py="http://genshi.edgewall.org/">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Beer Fermenation Control System</title>
    <style type="text/css">
        body            {font-family: sans-serif;
                         font-size: 16px}
        h3              {font-size: 24px;}
        table           {border: 1px solid #000;
                         border-spacing: 0px;
                         margin-left: auto;
                         margin-right: auto;}
        table a:link    {color: #000;
                         text-decoration: none;}
        table a:visited {color: #000;
                         text-decoration: none;}
        table a:hover   {color: #000;
                         text-decoration: underline;}
        tr.even         {background-color: #FFF;}
        tr.odd          {background-color: #FFC;}
        th              {border: 1px solid #000;
                         padding: 10px;
                         border-spacing: 2px;}
        td              {border-top: 1px solid #000;
                         padding: 2px 5px;}
        .center         {text-align: center;}
        .sensor1        {background-color: #FC0;}
        .sensor2        {background-color: #C3F;}
        .sensor3        {background-color: #0C0;}
        .sensor4        {background-color: #F60;}
        .avg            {background-color: #AAA;}
        .max            {background-color: #CCC;}
        .min            {background-color: #EEE;}
        .relay_on       {font-weight: bold;
                         color: #0C0;}
        #chart_div      {margin-left: auto;
                         margin-right: auto;}
    </style>
    <!--[if lte IE 8]><script language="javascript" type="text/javascript" src="http://127.0.0.1/htdocs/flot/
excanvas.min.js"></script><![endif]-->
    <script type="text/javascript" src="http://127.0.0.1/htdocs/flot/jquery.min.js"></script>
    <script type="text/javascript" src="http://127.0.0.1/htdocs/flot/jquery.flot.min.js"></script>
    <script type="text/javascript" src="http://127.0.0.1/htdocs/flot/jquery.flot.time.min.js"></script>

    <script type="text/javascript">
        $(function() {
            var ctr = 0;

            // Assemble initial sensor data to plot from the server
            var sensor1_data = ${sensor1_data};
            var sensor2_data = ${sensor2_data};
            var sensor3_data = ${sensor3_data};
            var sensor4_data = ${sensor4_data};

            // Plot the sensor data
            plot = $.plot("#chart_div",
                [
                 {label: "Sensor 1", data: sensor1_data, color: "#FC0"},
                 {label: "Sensor 2", data: sensor2_data, color: "#C3F"},
                 {label: "Sensor 3", data: sensor3_data, color: "#0C0"},
                 {label: "Sensor 4", data: sensor4_data, color: "#F60"}
                ],
                {
                    series: {
                        lines: {show: true},
                        points: {show: true}
                    },
                    grid: {
                        hoverable: true,
                        clickable: true
                    },
                    xaxis: {
                        mode: "time",
                        ticks: 12
                    },
                    legend: {
                        show: true,
                        noColumns: 4
```

```
        }

});

// Function that actually "draws" (inserts, really) the tooltip used below
function showTooltip(x, y, contents) {
    $('<div id="tooltip">' + contents + "</div>").css({
        position: "absolute",
        display: "none",
        top: y + 5,
        left: x + 5,
        border: "1px solid #ccc",
        padding: "2px",
        "background-color": "#eee",
        opacity: 0.80
    }).appendTo("body").fadeIn(200);
}

// Show a handy tooltip when a data point is clicked on the plot
$("#chart_div").bind("plotclick", function (event, pos, item) {
    $("#tooltip").remove();
    if (item) {
        var date = new Date(item.datapoint[0]);
        var str_ts = date.getHours() + ":" + date.getMinutes() + ":" + date.getSeconds();

        showTooltip(item.pageX, item.pageY,
                    item.series.label + " at " + str_ts + " = " + item.datapoint[1]);
    }
});

// When the page loads we need to know the latest timestamp so we can keep updating from there
// Store the value in a hidden dive on the page to make retrieval simple
$("#last_ts").text(sensor1_data[59][0]);

var ret = [];
function get_new_values(timestamp) {
    // Perform an AJAX get request to get the new data values to display
    $.get("/bfcs/chart_update", {last_ts: timestamp / 1000}, function(data) {
        ret = []

        if (data != '') {
            // Parse the returned string into an array
            rows = data.split(",")
            //<![CDATA[
            for (i = 0; i < rows.length; i++) {
            // ]]>
                ret.push(rows[i].split(" "))
            }

            // Grab the last row and set the new starting timestamp
            // alert(ret);
            $("#last_ts").text(ret[ret.length - 1][1]);
        }
    });
    return ret;
}

function update() {
    // Get the new values to plot and show in the readings table
    var ts = $("#last_ts").text();
    var rows = get_new_values(parseInt(ts));

    //<![CDATA[
    if (typeof rows !== 'undefined' && rows.length > 0) { // stuff like this is why js is stupid
    // ]]>

        sensor1_data.splice(0, rows.length);
        sensor2_data.splice(0, rows.length);
        sensor3_data.splice(0, rows.length);
        sensor4_data.splice(0, rows.length);

        //<![CDATA[
        for (i = 0; i < rows.length; i++) {
        // ]]>
            var row = rows[i];
            // Update the arrays containing the raw sensor data for plotting
            sensor1_data.push([row[1], row[2]]);
            sensor2_data.push([row[1], row[3]]);
            sensor3_data.push([row[1], row[4]]);
```

```
                        sensor4_data.push([row[1], row[5]]);

                        // Remove the last row from readings table
                        $("#readings tr:last").remove();

                        // Toggle ctr to keep track of the appropriate row CSS class
                        var tr_class;
                        if (ctr == 0) {
                            tr_class = "odd";
                        } else {
                            tr_class = "even";
                        }
                        ctr ^= 1;

                        // If the relay is on, set the appropriate CSS class
                        var relay_class = '';
                        if (row[6] == "on") {
                            relay_class = "relay_on";
                        }

                        // Insert new rows after the table header in readings table
                        $('<tr class="' + tr_class + '">' +
                            '<td>' + row[0] + '</td>' +
                            '<td class="center">' + row[2] + '</td>' +
                            '<td class="center">' + row[3] + '</td>' +
                            '<td class="center">' + row[4] + '</td>' +
                            '<td class="center">' + row[5] + '</td>' +
                            '<td class="center ' + relay_class + '">' + row[6] + '</td>' +
                          '</tr>').insertBefore($("#readings tr:eq(1)"));
                    }

                    // Assemble new plot data into format needed to plot
                    var new_data = [
                        {label: "Sensor 1", data: sensor1_data, color: "#FC0"},
                        {label: "Sensor 2", data: sensor2_data, color: "#C3F"},
                        {label: "Sensor 3", data: sensor3_data, color: "#0C0"},
                        {label: "Sensor 4", data: sensor4_data, color: "#F60"}
                        ];

                    // Plot the new data, update the grid, then redraw
                    plot.setData(new_data);
                    plot.setupGrid()
                    plot.draw();
                }

                // Re-call this function every second so that updates happen automagically
                setTimeout(update, 1000);
            }
            update();

        });
    </script>
</head>
<body>
    <h1>${title}</h1>
    <h3>Graph of Last 60 Readings</h3>
    <div id="chart_div" style="width: 900px; height: 500px;"></div>
    <div id="last_ts" style="display:none;"></div>
    <h3>Table of Last 300 Readings (5 mins)</h3>
    <table id="readings">
        <tr>
            <th>Time</th>
            <th class="sensor1">Sensor 1</th>
            <th class="sensor2">Sensor 2</th>
            <th class="sensor3">Sensor 3</th>
            <th class="sensor4">Sensor 4</th>
            <th class="max">Relay Status</th>
        </tr>
        <tr py:for="ctr, row in enumerate(results)" class="${ctr % 2 and 'odd' or 'even'}">
            <td>${row[0]}</td>
            <td class="center">${row[1]}</td>
            <td class="center">${row[2]}</td>
            <td class="center">${row[3]}</td>
            <td class="center">${row[4]}</td>
            <td class="center ${row[5] == 1 and 'relay_on' or ''}">${int(row[5]) and 'on' or 'off'}</td>
        </tr>
    </table>
    <br />
    <a href="/bfcs">Back</a> -- <a href="./${url_date}/all">All</a>
```

```
</body>
</html>
```