

HYDROGEN SYSTEM

Timebox 3 – 12-04-2011

This report covers Timebox 3 in the realization phase of the Hydrogen subproject, which is a part of the overall Energy Hub project.

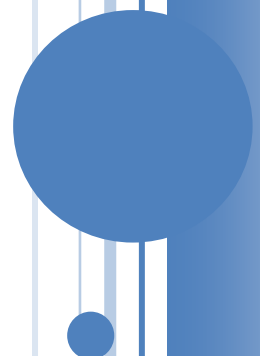
The project is a mandatory part of the 4th semester at the Electronic Design Engineer education at AU-IBT.

The Project has been supervised by Klaus Kolle and Morten Jakobsen both teachers at the Electronic Design Engineering program.

Lasse Lykkegaard

Dennis Thomsen

Knud Baastrup



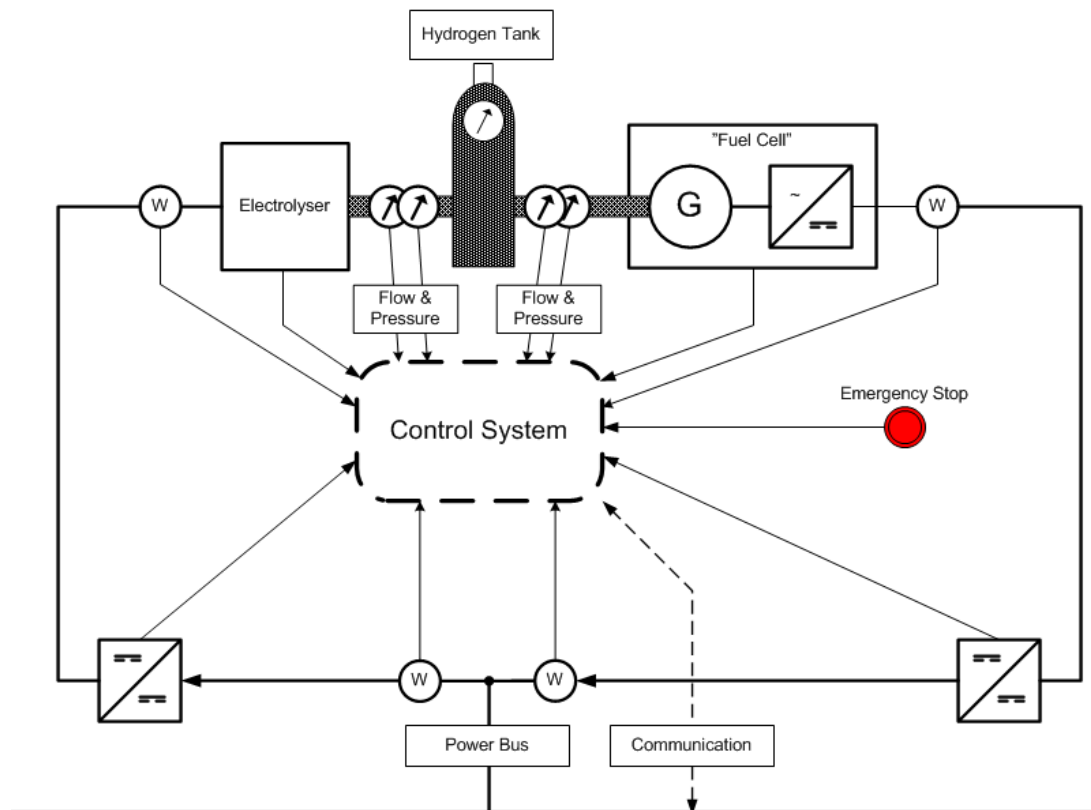
CONTENT

Content.....	2
System Overview	3
Change log	3
Deployment plan	4
Time box 3 Specification	5
Development Plan.....	5
Verification Plan.....	6
SW Design.....	7
Efficiency calculation	7
Graph	8
User Authentication	9
Login Menu.....	9
Register with client side validation of input data.....	9
Change Profile with update of email address and password	9
H2 Notification Service (Mail List).....	10
Verification	12
Functional tests.....	12
Interactive Design	13

SYSTEM OVERVIEW

Lasse Lykkegaard

Below figure show the System Overview created during the PRO3 Launch project



CHANGE LOG

This section describes the overall project adaptations done in PRO4 compared to PRO3.

The altering specifications and the work in the realization phase have revealed some none coherent specifications which have been altered.

The functions `getSensorData()` does not need the `ProductionID` argument supplied, when working in the data model...sensors are no longer connected to a Production sytem

The interface between the user-space application and the Data model was previously separated with two interfaces, one for the sensor class and one for the production system class. These two have been merged together into a `h2SystemInterface` class.

The State Machine diagram of the Fuel Cell and Electrolyzer should be altered to include a super state as emergency shutdown. At least the emergency shutdown should be made a blocking condition. Super state is preferred.

DEPLOYMENT PLAN

Knud Baastrup

Table 1 shows the deployment plans that were initially defined during the strategy planning of timebox 1. The plan has now been updated with more details on the later product versions that in the initial plan just included a heading for the expected deliverables.

Table 1 Deployment Plan

Product Version	Functionality	Deployment Date
1	<p><u>Start/Stop Electrolyzer and implement Database:</u></p> <p>The Electrolyzer can be started and stopped from a software application running in user-space on a LPC2478 Development kit. The power bus is simulated by using the XPR 60-100 power supply.</p> <p>Database design is settled and implemented on lene-lasse.dk</p> <p>Participatory session with low fidelity prototype completed.</p> <p>The product version consists of the following artifacts:</p> <ul style="list-style-type: none"> • Electrolyzer (final version) • Converter/relay-switch (prototype 1) • LPC2478 Development kit (final version) • Driver for converter/relay-switch (final version) • H2 application (version 1) • Database design (final version) • Low fidelity prototype • Report documenting the design 	15-03-2011
2	<p><u>Measure efficiency of the Electrolyzer and implement dynamic web:</u></p> <p>It is possible to measure the efficiency of the electrolyzer by measuring the energy consumed versus energy produced by the electrolyzer. This requires the ability to measure voltage, current, flow and pressure. The measured data can be read-out from the H2 application sw.</p> <p>Dynamic web site can show measured data (data are simulated using scripts).</p> <p>Video presentation to cover the participatory session with low fidelity prototype.</p> <p>The product version consists of the following artifacts:</p> <ul style="list-style-type: none"> • Current sensor (prototype 1) • Voltage sensor (prototype 1) • Flow sensor (prototype 1) • Driver for A/D converter (final version) • H2 application (version 2) • Dynamic web site (version 1) • Video presentation (final version) • Report documenting the design 	29-03-2011

3	<u>Gain CAN knowledge and finalize PHP Websites:</u> <ul style="list-style-type: none"> Establish CAN communication between 2 LPC2478 boards Login support for PHP website (with java script validation) Support for Mail notification for sensor data Improve SQL query for power calculation and improve grap presentation 	12-04-2011
4	<u>Communicate with Hub, introduce Fuel-cell and get started with FPGA support</u>	03-05-2011
5	<u>EMC verification and finalize FPGA support:</u>	17-05-2011

TIME BOX 3 SPECIFICATION

Lasse Lykkegaard

This time box covers the below requirements from PRO3 launch, which must be realized in order to meet the functionality specified for product version 3 in the deployment plan in Table 1.

ID	Requirements
N.S.	Make a working Login Script for the webpage
N.S.	Establish CAN communication between 2 LPC2478 boards
N.S.	Make a working mail system, where it is possible to mark wanted notifications
N.S.	Improve graph function, and SQL query

Development Plan

All

Timebox 3	Week 14						
Task	M	T	W	T	F	S	S
Graph function		L			L		
SQL improvements	L				L		
Mailsystem				D	D	D	D
CAN communication	L		L	L			
Login Scripts		K	K			K	K

L: Lasse Lykkegaard

K: Knud Baastrup

D: Dennis Thomsen

A: All

VERIFICATION PLAN

Lasse Lykkegaard

The work load of this time-box was not specified during the Launch-phase, therefore no pre written acceptance test available.

Requirements and specifications have changed during the project – Product acceptance test changes also and are written to fit current timebox

Table 2 Product Acceptance test relevant for timebox 3 deployment

ID	Acceptance Tests	Remarks
	Mail List <ol style="list-style-type: none"> 1. Make a new Subscription for a sensor 2. Update a exciting Subscription 3. Check for an email in mail box 4. Ensure that database was updated 	
	User Authentication <ol style="list-style-type: none"> 1. Register and login 2. Go through the different sites verifying you're still logged in and have access to pages requiring users to be authenticated. 3. Log out and ensure that you do no longer have access to pages requiring users to be authenticated. 	
	Change Profile <ol style="list-style-type: none"> 1. Login 2. Create an email notification via H2 Notification Service. 3. Change your email address via Change Profile menu option. 4. Verify that email notifications will be sent to updated email address. 	
	Can Communication <ol style="list-style-type: none"> 1. CAN communication between two board can be seen visual by connecting two boards two computers send messages to each other 	
	SQL-improvements <p>See old and new version of MySQL queries /PHP notice the difference.</p> <p>See improvements on graph-function</p>	

Test cases to verify the functionality in more detail will be developed in parallel with the implementation and the test results will be documented.

SW DESIGN

Efficiency calculation

Firstly our idea was to make a SELECT statement in a SELECT statement, thereby pinpointing the wanted values and let MySQL do the calculation. Due to an unsolved error, it did not bail to our advantage. The return table had some surplus data, which we were unable to get rid of. This was then sorted out with forloops in php.

This problem has been corrected.

In Figure 1 it is shown how to calculate the average of our sensors system.

6 iterator functions called a,b,c,d,e,f loops through the two tables of interest, when all required conditions are fulfilled the row is returned. The data row then contains the wanted data and is averaged.

Two tables are needed to cross reference the "sensor number" with the "sensor name" and in the same time all data must be between two dates.

```
#Efficiency over time
$result9 = mysqli_query($dbc,"SELECT AVG( (
    a.data * b.data *100
) / ( e.data * $hydrogenconstant ) ) AS
data
FROM `sensor` a, sensor b, sensor_list c, sensor_list d, sensor
e, sensor_list f
WHERE a.timestamp = b.timestamp
AND a.timestamp = e.timestamp
AND a.sensor_id = c.sensor_id
AND b.sensor_id = d.sensor_id
AND e.sensor_id = f.sensor_id
AND c.name = 'current_in_sensor'
AND d.name = 'voltage_in_sensor'
AND f.name = 'flow_out_sensor'
AND c.production_id =0
AND d.production_id =0
AND f.production_id =0
AND a.timestamp
BETWEEN SUBDATE( NOW( ) , $_POST[data_interval] )
AND NOW( )
AND b.timestamp
BETWEEN SUBDATE( NOW( ) , $_POST[data_interval] )
AND NOW( )
AND e.timestamp
BETWEEN SUBDATE( NOW( ) , $_POST[data_interval] )
AND NOW( )
");
$result9 = mysqli_fetch_array($result9,MYSQLI_ASSOC);
```

Figure 1: shows the averaging of the sensor data

Graph

The graph is generated with a php library – phplot. A php-file generates a PNG-image, this sends out headers of an image and mySQL connection will not work. Even stacking headers with ob_start didn't solve the problem. Therefore this alternative approach shown in Figure 2. All the data is transferred to a file and read again by the graph.php file.

```
$File = "Generator_Graph.txt";
$Handle = fopen($File, 'w');
while($data = mysqli_fetch_array($all_data,MYSQLI_ASSOC))
{
    if($_POST["graph_interval"] == 365)
    {
        $key = date("d/m",strtotime($data['timestamp']));
    }
    else if($_POST["graph_interval"] == 31)
    {
        $key = date("d/m",strtotime($data['timestamp']));
    }
    else if($_POST["graph_interval"] == 7)
    {
        $key = date("H:i",strtotime($data['timestamp']));
    }
    else if($_POST["graph_interval"] == 1)
    {
        $key = date("H:i",strtotime($data['timestamp']));
    }
    $set = $key . "\n" . $data['data'] . "\n";
    fwrite($Handle, $set);
}
fclose($Handle);
print ' ';
```

Figure 2: the timestamp is stripped according to the number of samples

After this function was made we've implemented sessions, it might be a possibility to use a session to hold the database connection, gaining access for all pages without opening a socket to the database. At least the Filename shall be something different for each session. For example a code generated from the session id. The best way was to make graph.php query for its own data.

Figure 3 below shows how the data is retrieved in graph.php getting the filename with the _GET method.

```
//read data from file
$File = $_GET['file'];
if($handle = fopen($File, 'r'))
{
    while (!feof($handle))
    {
        $time = fgets($handle, 512);
        $data = fgets($handle, 512);
        if ($time) //tests for the last sample before EOF
        {
            $seff_stat[] = array($time,(int)$data); //appends on an existing array
        }
    }
    fclose($handle);
}
```

Figure 3: data read from file

User Authentication

Knud Baastrup

Login functionality has been added to allow user authentication for certain web pages and as well to support an email notification service (as described in the

H2 Notification Service (Mail List section further below) that require a registered email address.

Currently it is only the pages **H2 Notification Service** (mail_list.php) and **H2 System Dokumentation** -> **Launch Report** -> **"All links"** (our_Report_document.php) that require a registered user.

In general all user inputs are managed via the HTML form tag using appropriate input element tags and data is sent to the PHP server via the form action attribute, when the submit input element has been activated (button pressed).

Login Menu

The login menu provides the option to Login or Register, or if already logged in, to Logout or Change Profile. The login menu is implemented in the /includes/login_menu.html file and on the web page located in the right side of the Link Banner.

The login menu is available on all web pages and allows the user to make use of the login options at any time. Some pages will however prompt the user to login, if not already done.

Login will be handled by the login.php page. The user_id is after login saved in a session, which then only require the user to login once when exploring the H2 Web site. Prior to login, the current URL is saved in the session (implemented in login_menu.html), which allow relocation back to the site from where the login were requested.

Logout will be handled by the logout.php page where the session variables and the session it selves are cleared and destroyed. During logout the user will via an alert pop-up box be informed that the logout is completed with relocation back to the Front Page (implemented in login_menu.html with a small Java script).

Register with client side validation of input data

Register will be handled by the register.php page and will require the user to register a username and email address and as well provide a password for authentication.

The form onsubmit attribute has been utilized to perform a client side validation of the user inputs before the data is processed by the PHP server. The client side validation is performed using a Java script embedded into the HTML part of register.php.

Performing a client side validation will save some communication between client and server and could for this reason increase response time and save some network load. In most cases it will however not make any difference due to the often plenty amount of available bandwidth and the often very powerful and dedicated web servers.

The client side validation is performed using a JAVA script that checks the user input using regular expressions.

Change Profile with update of email address and password

Change Profile will be handled by change_profile.php and will allow the user to update email and password settings. The inputs are also validated using JAVA scripts.

The email address update will also take affect for the notifications subscribed by the concerned user as these notifications are linked to the specific user_id in the mail_list table and not the email address it selves

H2 Notification Service (Mail List)

Dennis Thomsen

H2 Notification Service is the signup page for the H2 Mail List, where a user can subscribe themselves to any of the sensors attached to the system, and also what conditions should be meet, for an email to sent to the user, it also enables the user to change the conditions of previously made subscriptions, upon making a new subscription, a test mail is sent to the user's email address to test all is in order.

The page consists of dropdown menu from which the sensor, that the user wishes to subscribe to can be selected from.

After a sensor has been chosen an input box appears under the dropdown menu, with the unit of the sensor selected next to it, this box is used to entre in, the condition under which that the system should send a mail to the user, the reason this box is not there from the beginning is, so the units that the condition relates to could be loaded, thereby making it easier for the user to enter the desired conditions.

After choosing the terms of a subscription, the subscription is then checked against the database, to see if the user is already subscribed to that partially sensor. Then the option is giving to update the database with the newly entered conditions, otherwise it creates the subscribing in the database, send the test mail and notify the user of the success of the operation.

```
$data_rows = @mysqli_query($dbc, 'SELECT sensor_id, name FROM sensor_list ORDER BY name');
echo '
    <form action="mail_list.php" method="post">
    <div style="height:150px">
        <select name="sensor_list" onclick="javascript:document.mail_list.php.submit();" > ' ;
        while ($row = mysqli_fetch_array($data_rows,MYSQLI_ASSOC))
        {
            $name = str_replace("_", " ", $row['name']);
            echo "<option ". ( (isset($_POST['sensor_list']) && ($_POST['sensor_list']== $row[
            'sensor_id'])) ? "selected" : "" ) . " value=" . $row['sensor_id'] . ">". $name . "
            </option> \n";
        }
        echo "<input type=submit name=select value=Select><br />";
```

Figure 4: Code for the dropdown menu used to selecting sensor

The dropdown menu is made so, that after a choice has been select and posted to the site, that choice will be preselected for the menu box, so the user be sure, that it is the correct sensor, he or she is signing up to.

In order to preselect an option in a dropdown menu "selected" is used, the following line from the code in Figure 4, is what ensures that is the correct option in the menu that is preselected:

```
( (isset($_POST['sensor_list']) && ($_POST['sensor_list']== $row['sensor_id'])) ?
"selected" : "" )
```

The syntax (? :) is what is known as conditional operator, which is an conditional expression, rather than a conditional statement like an if-else.

The main difference between the two, being that an expression returns a value, while a statement performs an action.

If we looked at it like a pseudo if-else statement it would correspond to the statement below:

```
if(isset($_POST['sensor_list']) && ($_POST['sensor_list'] == $row['sensor_id']))
{
    Then print "selected"
} else {
    Then print ""
}
```

```
$S_name = str_replace("_", " ", $_POST['name']);
$user_q = mysqli_query($dbc, "SELECT mail, name FROM user WHERE user_id = " . $_SESSION['user_id'] . " "); or
trigger_error("Query: $query\n<br />MySQL Error: " . mysqli_error($dbc));
while ($user = mysqli_fetch_array($user_q, MYSQLI_ASSOC))
{
    $query = "INSERT INTO `mail_list` (`user_id`, `sensor_id`, `condition`) VALUES (" . $_SESSION['user_id'] .
    ", " . $_POST['sensor_id'] . ", " . $_POST['number'] . " )";
    $result = mysqli_query($dbc, $query) or trigger_error("Query: $query\n<br />MySQL Error: " .
    mysqli_error($dbc));
    if (mysqli_affected_rows($dbc) == 1)
    {
        echo "<p>
        <h3>Success!</h3><br>
        you where added to the " . $S_name . " mailing list<br>
        a test mail should appear in your mail shortly
        </p>";
        // Test mail
        $email = $user['mail'];
        settype($email, "string");
        $body = 'Dear ' . $user['name'] . ' you where recently added to the ' . $S_name . ' mailing list please
        contact http://lene-lasse.dk if you did not do this yourself';
        $subject = 'Mailing List';
        $header = 'From: Hydrogen subsystem mail list' . "\r\n" . 'Please note: this e-mail was sent from a
        notification-only address that cannot accept incoming e-mail. Please do not reply to this message. ';
        mail($email, $subject, $body, $header);
    }
    else
    {
        echo '<p> something failed in the sign up</p>';
    }
}
```

Figure 5: Inserts new subscription into “mail_list” table and sends test mail to user

The creation of new subscriptions is handled in the code from Figure 5, the user’s “mail” and “name”, is pulled from the “user” table, and the new subscription is then inserted into the “mail_list” table in the database, if successful, the user is notified of the success, and a test mail is send to the user’s email, in the case of a failure happing when inserting into the database a error message is printed to the user.

VERIFICATION

Lasse Lykkegaard

The verification for timebox 3 includes a number of functional tests and as well the product acceptance tests required prior to deployment.

Functional tests

The following test cases were executed to verify the functionality delivered in timebox 3

Test Case ID	3.1		
Description	Verify that the system can send out an email		
Preconditions	User logged in on website, access to MySQL - database		
Command/Action/Steps	Expected Result/Verification	PASSED	
Select Sensor	Condition box with unit	OK	
Enter condition and press subscribe button	<p>if not already subscribed to the sensor , confirmation will pop up if adding to database was a success, and a test email is sent to the users email.</p> <p>Check email shows up in the users mailbox, and the subscription has been added to the database</p> <p>if the user is already subscribed to that mail list then update option shows up with the current conditions for the subscription</p>	OK	
Press update button	<p>System Notice that the update was successful</p> <p>Check the database to confirm</p>	OK	

Test Case ID	3.2		
Description	Verify that it is possible to log in and keep that session		
Preconditions			
Command/Action/Steps	Expected Result/Verification	PASSED	
Register a new user at: http://www.lene-lasse.dk/pages/register.php	<p>Thank you for registering!</p> <p>You should now be able to login: Login</p>	OK	
Log in	You will be directed back to the page from where	OK	

	you requested the login or the requested page that forced you to perform a login.	
Visit different sites at www.lene-lasse.dk/ without closing the browser.	Notice if the “log out” is vanishing If “Log out” stays there – a session is still intact.	OK
Press log out or close browser.	Session should now be lost. Try open the site again, it should now say log in instead of log out.	Ok

Test Case ID	3.3		
Description	Verify that CAN is working on the boards		
Preconditions	Two LPC2478 boards		
Command/Action/Steps	Expected Result/Verification	PASSED	
\$ Verify that the boards only are connected via CAN interface			
\$ Send test msg from board 1			
\$ Recive test msg on board 2	Mgs received and displayed on console		
\$ Send test msg from board 2			
\$ Recive test msg on board 1	Mgs received and displayed on console		

INTERACTIVE DESIGN

Dennis Thomsen

Raw Video footage from interview with Mr. Esben Wolf regarding interactive design of the maintenance part of the system, has been edited into a 9 min long short film along with what we observed of noteworthy “gold pieces” so to speak, and divided into about 10 short sequences depending on topic, the editing was done using the Sony “Vegas Movie Studio HD”

The video can be found at <http://lene-lasse.dk/files/IDE/> named IDE.wmv

From this session we gained a great deal knowledge about what worked and what could be improved upon both in regards to the GUI prototype for the current project, but also for other such sessions in future projects.

In regards to future video interviews like this:

- Start by shooting a real introduction for voice recognition
 - It is important to be able to identify who says what in the video and it can't be assumed that the viewers know the people in the video.
- Full focus on design with text gibberish

- Prototype in scaled version
 - Having the prototype in 1:1 scale worked very well.
- Use a tripod when filming a object that small
- Be aware of external interference
 - Do a proper stake holder analysis before visiting the Customer, so you are able to prioritize inputs from different people that might share their view of the product.

As for this project we learned that:

- Priority of Errors
 - It must be possible to distinguish errors in terms of priority.
 - The priority must be visualized with a number or a color.
- Click-ability
 - Solution with dashed polygon combined with red color to indicate an clickable area were confirmed, but red color should then not be used if something cannot be clicked.
- Video Guide
 - Extra focus and guidance on difficult tasks archived via video guide.
- Graph well received
 - Have consistency for the buttons on front page was good, and the ability to see what has going on when the error was reported.
- Maintenance History
 - Customer wants popup maintenance history with relevant product info.
- Time Estimation
 - Estimation for the time needed to complete a maintenance job, should be present under the maintenance menu.
- Conclusion of session
 - The low fidelity prototype kept focus on functionality instead of design