HYDROGEN SYSTEM

Timebox 2 – 29-03-2011

This report covers Timebox 2 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

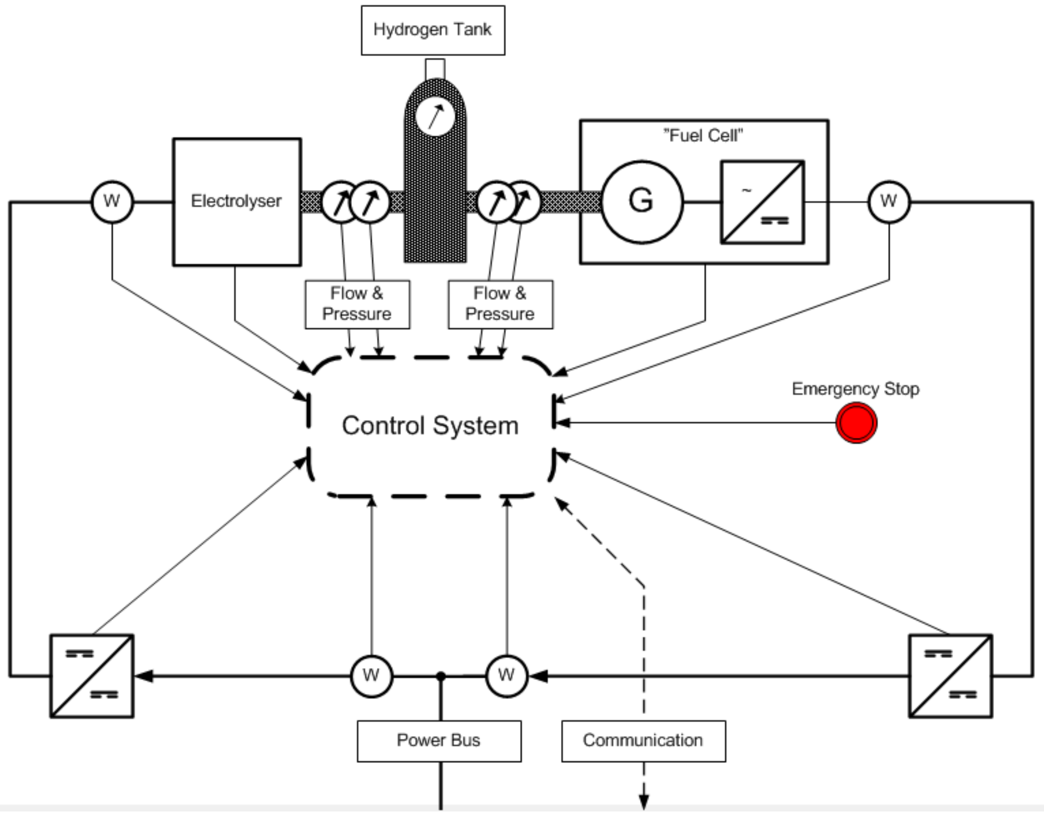
Knud Baastrup

Content

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. Instead it has been supplied with a &data, where the data is written. This makes us able to track errors. getSensorData(int SensorId, int &data)

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	<u>Start/Stop Electrolyzer and implement Database:</u> 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. Database design is settled and implemented on lene-lasse.dk Participatory session with low fidelity prototype completed. The product version consists of the following artifacts: • 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	<u>Measure efficiency of the Electrolyzer and implement dynamic web:</u> 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. Dynamic web site can show measured data (data are simulated using scripts). Video presentation to cover the participatory session with low fidelity prototype. The product version consists of the following artifacts: • 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	Gain CAN knowledge and finalize PHP Websites: • 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	Communicate with Hub, introduce Fuel-cell and get started with FPGA support	03-05-2011
5	EMC verification and finalize FPGA support:	17-05-2011

Time box 2 Specification

Knud Baastrup

This time box covers the below requirements from PRO3 launch, which must be realized in order to meet the functionality specified for product version 2 in the deployment plan in Table 1.

ID	Requirements (as specified during PRO3 Launch)
G1.5	The H2 subsystem shall monitor input/output power-bus voltage and input/output power-bus current and notify the Hub and other subscribed observers on changes in voltage or current level.
G1.7	The H2 subsystem should be able to deliver efficiency data measured over certain duration of time that at least should cover up to 24 hours.
E1.8	The H2 subsystem should monitor the input voltage and input current to the electrolyzer to allow calculation of the converter efficiency for the converter required between power-bus and electrolyzer.
E1.9	The H2 subsystem shall monitor the hydrogen flow and hydrogen pressure delivered by the electrolyzer to allow calculation of the electrolyzer efficiency.

The time box will as well include the web design and PHP code required to show the measured data on a dynamic web-site.

Development Plan

All

Timebox 2	Week 11							Week 12						
Task	M	T	W	T	F	S	S	M	T	W	T	F	S	S
Flow Sensor circuit (Mock-up and Diagram)	D	D	D	D	D	D	D	D	D	D	D			
Current Sensor (Mock-up and Diagram)	K	K	K	K	K	K	K	K	K	K			K	K
Voltage Sensor (Mock-up and Diagram)	K	K	K	K	K	K	K	K						
ADC Driver for sensors	D	D			D	D	D	D				D	D	D
H2 application support for sensors	L	L									L	L	L	L
Implement Database				D										
Web design	L	L	L	L	L	L	L	L						
PHP/SQL code to support web	L	L	L	L	L	L	L	L	L	L	L	L	L	L
Video editing for IDE1				D	D									
Component Diagram												K		
EMC reviews										L				

L: Lasse Lykkegaard
K: Knud Baastrup
D: Dennis Thomsen
A: All

Verification Plan

Knud Baastrup

A set of Product Acceptance test that match each requirement were prepared during the Launch project. The following 4 tests were defined for the requirements included in this timebox. Some remarks have been added for some of the tests to suggest some alternative measures that are needed due to some functionality still missing, but planned for later timebox realizations.

The tests shown in Table 2 will be performed in prior to deployment of timebox 2.

Table 2 Product Acceptance test relevant for timebox 2 deployment

ID	Acceptance Tests (as specified during PRO3 Launch)	Remarks
G1.5	The test can be performed using the procedure described below: 1. Start the system 2. Make manual readings of the power 3. Compare the readings to those from the system. If measurements are equal to the data- system monitors input/output power-bus voltage and input/output power-bus current correctly 4. Insure that the Hub and subscribers received notification	Sensor data will be printed on consol. Interface towards Hub not yet implemented.
G1.7	The test can be performed using the procedure described below: 1. Monitor system manually by reading power sensors at random time. 2. Check if the efficiency data is equal to the manually measured data.	Handled in database.
E1.8	The test can be performed using the procedure described below: 1. Start hydrogen production 2. Note down the measurements of the input current and voltage of the Electrolyzer from power sensor. 3. Check if the power sensor measurements are equal to the data stored in data-log(/UI)	Will not be supported.
E1.9	The test can be performed using the procedure described below: 1. Make readings from the actual power and flow sensors 2. Compare readings to the system readings. 3. Repeat multiple times.	Will not be supported.

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

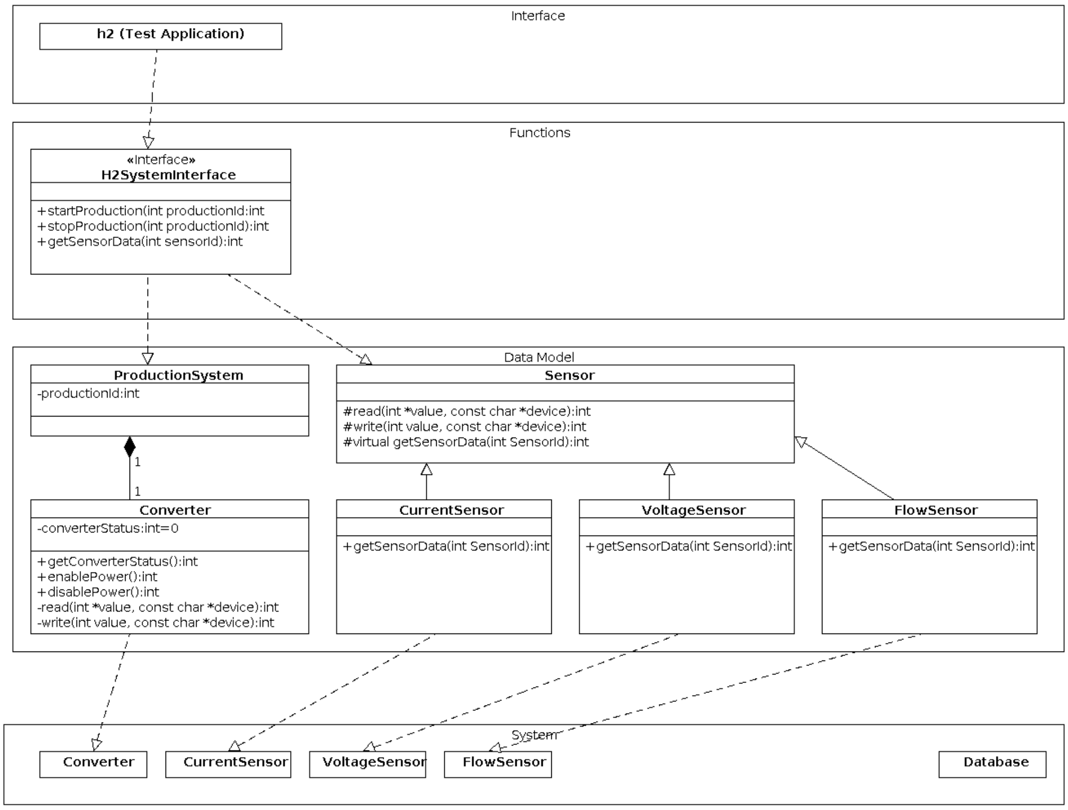
SW Design

Lasse Lykkegaard

The software is written as an extend of previous written classes, keeping the same style and calls as previous classes. An external reader should not tell the difference between who have written what part.

Class Diagram

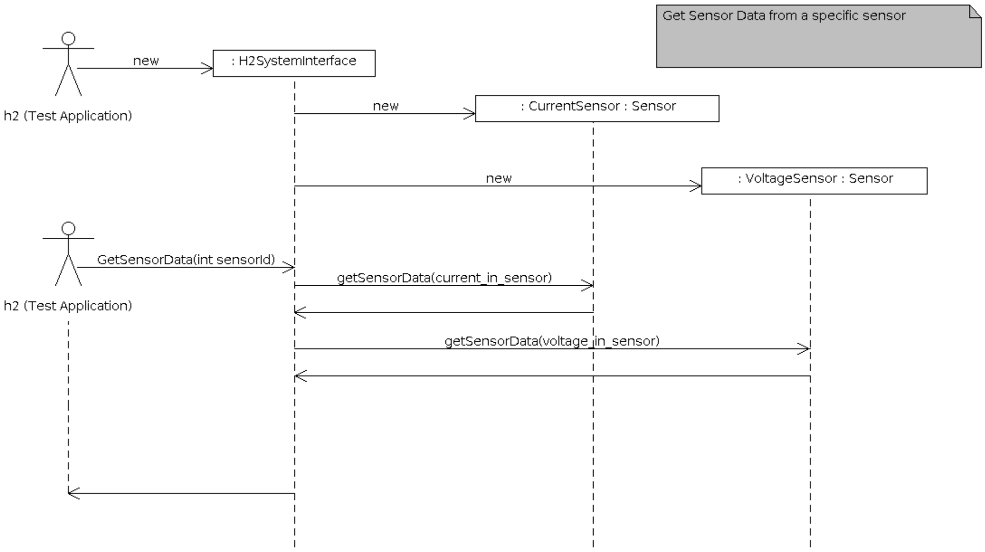
We've made a virtual Sensor class, so that all sensors derive the same interface towards the system interface and inherit the common functions of a sensor.



To overcome the problem about having none-named objects of our classes in the system interface the sensors are implemented in sub-class arrays of pointers, else we should have a function to determine what kind of sensor a *sensorId* is.

Use Case Scenarios

Lasse Lykkegaard



ADC DRIVER FOR UCLINUX

Dennis Thomsen

Rather than starting with writhing the driver from scratch, we used a Embedded Artist adc (ea_adc) driver designed for use with the LPC2478 board and modified it to suit our needs.

The ea_adc needed modification because it was made only to support 4 adc out of the 8 on the board and we need 7 adc's.

In addition the way the data was sent to userspace was also changed from using a function called intostr which converted the data from the adc into a hex value before sending it, it now simple uses copy_to_user and sends a int, which is more suited as calculations needs to be done with the data.

Below are exacts from the code so some of the changes can be seen (EA_ADC refers to the driver made by Embedded Artist, ADC to our modified version)

EA_ADC

```
static u32 enable_bits[] =
{
   (1 << 14), // ad0.0
   (1 << 16), // ad0.1
   (1 << 18), // ad0.2
   (1 << 20), // ad0.3
};
```

ADC

```
static u32 enable_bits[] =
{
   (1 << 14), // ad0.0 P0[23]
   (1 << 16), // ad0.1 P0[24]
   (1 << 18), // ad0.2 P0[25]
   (1 << 20), // ad0.3 P0[26]
   (11 << 28), // ad0.4 P1[30] PINSEL3 0xE002 C00C
   (11 << 30), // ad0.5 P1[31] PINSEL3
   (11 << 24), // ad0.6 P0[12] PINSEL0 0xE002 C000
   (11 << 26), // ad0.7 P0[13] PINSEL0
};
```

EA_ADC From Function adc_open

```
if(chRefCnt[channel] == 0)
{
  file->private_data = (void *) channel;
  m_reg_bfs(PINSEL1, enable_bits[channel]);
}
```

ADC From Function adc_open

```
if(chRefCnt[channel] == 0)
{
  if (channel <= 3)
  {
    m_reg_bfs(PINSEL1, enable_bits[channel]);
  } else if (channel == 4 || channel == 5)
  {
    m_reg_bfs(PINSEL3, enable_bits[channel]);
  } else if (channel == 6 || channel == 7)
  {
    m_reg_bfs(PINSEL0, enable_bits[channel]);
  }
}
```

The added adc's are enabled though other PINSEL registers, if statements are used to swap depending on channel

ADC From function adc_read

```
res = adcRead(channel);
DPRINT("adc_read: %d \n",res);
int result = (int)res;
DPRINT("adc_read: %d \n",result);
if (copy_to_user(p_buf, &result, count))
  { // Some of the bytes could not be copied.
    return  -EFAULT;
  }
```

EA_ADC From function adc_read

```
if(endRead[channel])
{
  endRead[channel] = 0;
  return 0;
}
res = adcRead(channel);
len = intToStr(res, p_buf+*p_pos, count, 10);
if(len < count)
{
  p_buf[len++] = '\n';
}
endRead[channel] = 1;
```

endRead was removed, as all it seemed to do was cause the reading the fail every other time(only other place it could be found in the code was when it was declared) and as said above, intToStr was removed as returning an int was more convenient as it should be used in calculations

HW Design

Knud Baastrup

Figure 1**Fejl! Henvisningskilde ikke fundet.** shows the HW components introduced in Timebox 1 and Timebox 2 where the new components include a combined Voltage and Current Sensor, a Flow Sensor and some HW for gain and level adjustment. The new components are further described in the subsequent sections.

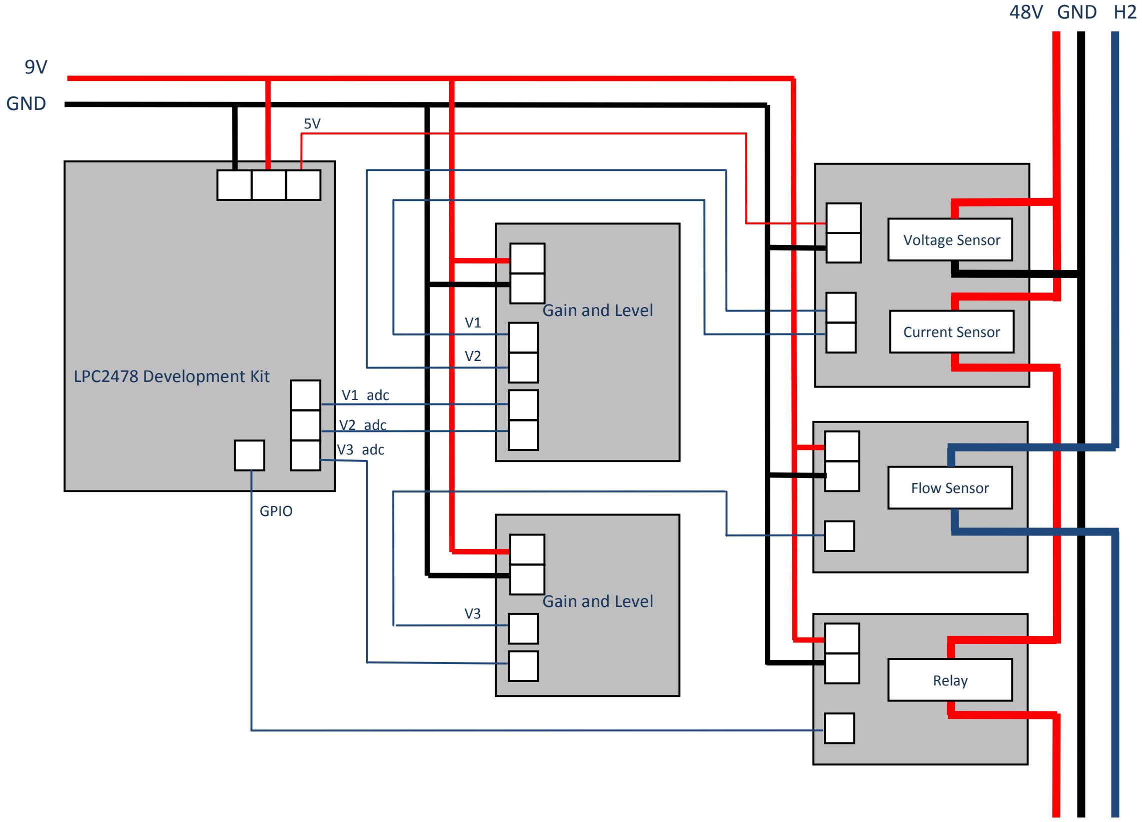


Figure 1 HW components introduced in Timebox 2

Current and Voltage Sensors

Knud Baastrup

The Hydrogen system requires a number of sensor circuits for measuring the current and voltage consumed by the electrolyzer and as well the current and voltage produced by the “fuel Cell”.

The measured current and voltage data are used when calculating the efficiency for the electrolyzer, “fuel Cell” and the power converters and as well to inform the Hub on changes in the voltage and current levels. See also requirements G1.5, E1.8, F1.9.

We will, in the first version of the hydrogen system, include the efficiency for the power converters as part of the complete efficiency for respectively the electrolyzer and “fuel Cell”, i.e. there will not be separate efficiency data for the power converters as stated in the anyway optional requirements E1.8 and F1.9.

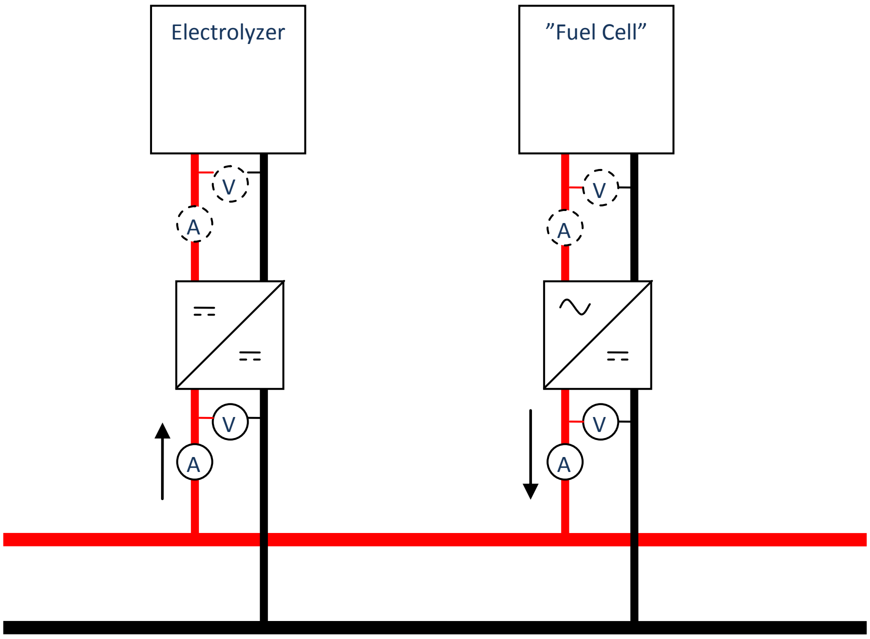


Figure 2 Current and voltage sensors required for the H2 system

We can as shown in Figure 2**Fejl! Henvisningskilde ikke fundet.** manage the measurements using two voltage sensors and two current sensors and for now skip the extra set of sensors (dashed in **Fejl! Henvisningskilde ikke fundet.**) that only will be required when introducing requirements E1.8 and F1.9.

Current Characteristics

The electrolyzer will initially use the 48VDC power bus voltage, but it might later be replaced by a 48VDC to 60VDC power converter as the electrolyzer will be a lot more efficient at 60VDC.

We know from measurements that the electrolyzer will consume power as below:

	Input Power	Input Current
Electrolyzer @ 48VDC	288 W	6 A
Electrolyzer @ 60VDC	1500 W	31 A

We do not yet know how much power we can expect from the “Fuel Cell”, but the generator that simulates the “fuel Cell” can according to its specification deliver up to 2000W (we expect it to me significant lower, but we will use this figure for now as a worst-case):

	Output Power	Output Current
“Fuel Cell”	~2000 W	~40 A

According to the above characteristics we need to measure a range from 6A – 40A.

Current Measurement

The current can be measured either by using a shunt resistor or by using a current transducer.

The shunt resistor is a resistor with a very accurate value of a few milliohms. The current to be measured will flow through the shunt and provide a voltage drop that is used as an expression for the amount of current. The accuracy is typical around ±0.25% or even ±0.1%

The current transducer utilize the Hall effect, i.e. it creates a magnetic field around the current to be measured, which due to the Hall effect, separates the charges and produce an electrical field that is used as an expression for the amount of current. The current transducer is not as accurate as the shunts, but it provides on the other hand an important galvanic isolation.

Galvanic isolation is important for the measurement as that will allow us to keep a separate ground level for the powerbus and by that prevent ground loops where an unwanted current in a shared ground could introduce noise into the more sensitive circuits.

Current Transducer LTS 15-NP

LEM manufactures some quite cheep and accurate current transducers that become a preferable choice due to the included galvanic isolation.

The current transducer LTS 15-NP is a current transducer that, according to the enclosed datasheet, utilizes the Hall Effect to measure current in the range from 0 - ±45A, which fully covers our required range from 0 - ±40A. It provides an analogue output in the range from 2.5 - 2.5 ± 1.875 V with an accuracy of ±0.7%. The supply voltage is 5V from where it will consume around 45mA.

Figure 3 shows a diagram of the current transducer where the secondary current (I_S) is depicted as a current generator that is followed by an I-to-V amplifier that produce an analogue output.

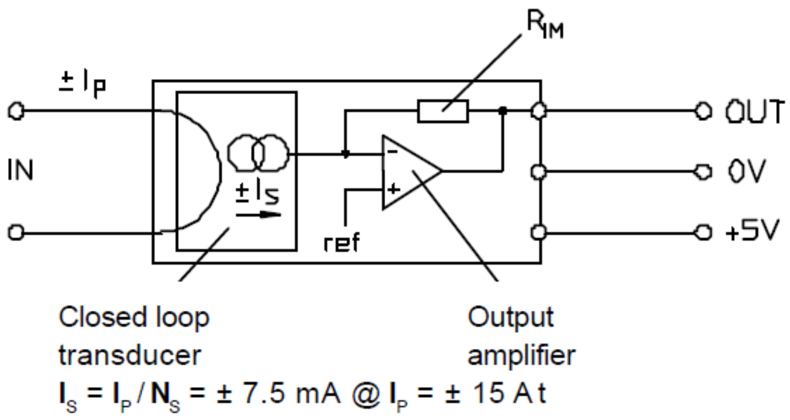


Figure 3 Diagram of the Current Transducer LTS15-NP

The output from the current transducer will require some gain and level shifting, which is covered in the section named Gain and Level Adjustment using Differential Amplifier.

Current Sensor Implementation

Figure 4 shows the schematic for the Current Sensor, which besides the LTS15-NP just include a 1 kΩ resistor to avoid impact from capacitive loads caused by bypass/decoupling capacitors. The 5V power supply will be delivered from the LPC2478 Development kit.

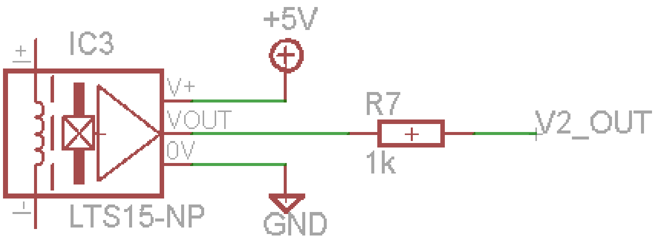


Figure 4 Schematic for Current Sensor

The Current Sensor will be located together with a Voltage Sensor to be described in coming sections. The first version will be realized on a PCB with pre fabricated holes, which is quick way to realize the sensors and as well provide some physical stability for a test that involves direct interaction with the powerbus.

Voltage Characteristics

The nominal voltage of the power bus is 48 VDC, but can according to the Power Bus Voltage Specification, vary in the range from 45 - 51 V, which includes ±2 V RMS and some possible AC ripple.

Voltage Measurements

Voltage can be measured using a simple voltage divider or by using a voltage transducer.

The voltage divider will provide a linear output with an offset, due to the measuring range from 45 – 51V. The voltage divider will however not provide any galvanic separation, which must be handled by a separate linear opto-coupler.

The voltage transducer will use the same principles as previously described for the current transducer where it now just measure a current proportional to the voltage to be measured. The voltage transducer will like the current transducer provide a galvanic separation.

Voltage Transducer LV 25-P

LEM manufactures a voltage transducer that measures a current in the range from 0 - ±14mA which is defined via an external resistor placed across the voltage to be measured. The required supply voltage is unfortunately 12V – 15V, which we do not have available and the voltage transducer is as well more than 3 times the cost of the LTS15 NP current transducer.

Voltage divider with IL300

A lot cheaper solution is the usage of a simple voltage divider together with the IL300 linear opto-coupler. Figure 5 shows the schematic for the simple voltage divider that is followed by a Voltage Follower to ensure low impedance and prevent any load from the IL300 based amplifier that follows.

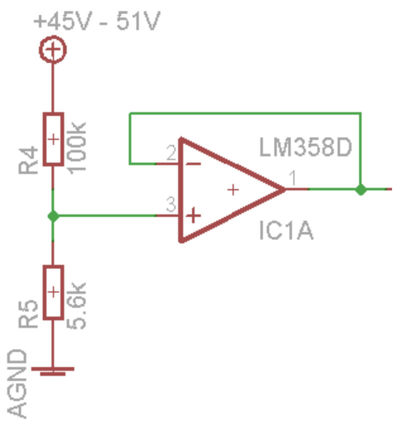


Figure 5 The voltage divider used in the Voltage Sensor.

We can find resistor values that provide an offset around 2.5V, which is the same offset provided by the LTS 15-NP current transducer. Using the same offset will make it easier to make a more general solution for the required gain and level adjustment.

Figure 6 show the galvanic isolation achieved using the IL300 linear opto-coupler. IL300 consists of a LED and two photodiodes where one photodiode is used to provide feed-back for improved linearity and the other to produce the galvanic separated output current. The output current is converted into a voltage in the I-toV amplifier where the IL300 output current creates the voltage across the R2 resistor that becomes the output voltage Vout.

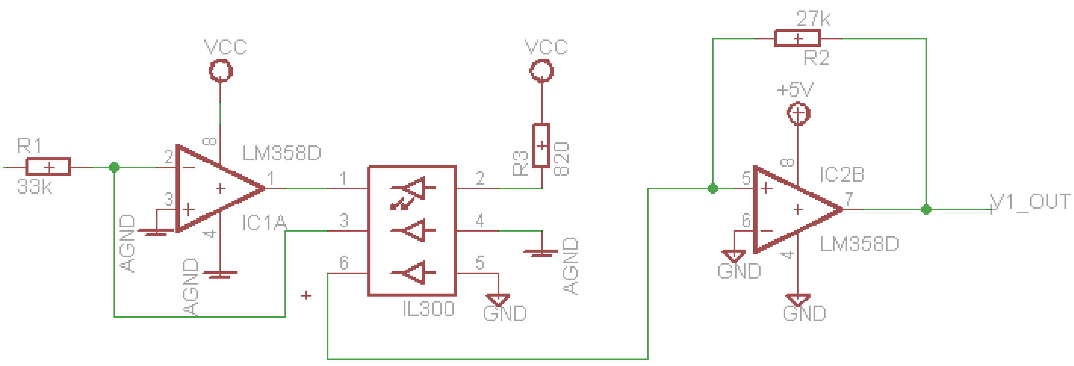


Figure 6 The galvanic separation achieved using IL300

The photodiodes can be either reversed biased for a photoconductive mode (like a zener diode) or zero biased for a photovoltaic mode (like a solar cell). According to datasheet, the best linearity is achieved in the photovoltaic mode on behalf of reduced bandwidth. In this case we do not require any bandwidth and the best option would be the photovoltaic mode, which explain the zero biasing performed for IL300 on pin 4 and 5 in Figure 6.

The best linearity is according to datasheet achieved by using a LED current between 5mA – 20mA, which match well with the sink current capabilities of the LM358 op amp.

R1 can be calculated as below based on the feed-back current. According to the datasheet the feed-back current is 75 uA when the LED current is 10 mA.

$$R1 = \frac{V_{in}}{I_{p1}} = \frac{2.5V}{0.075mA} = 33.33k\Omega$$

We choose the value 33kΩ for R1 that is the best match from the E24 range of resistors.

We have the possibility to add some gain when setting the value for R2, but we will however just target for unity gain and perform a separate level shift and gain adjustment.

Using the same value as R1 will however not guarantee a transfer gain of one as the transfer gain according to datasheet can vary from 0.59 to 1.53. IL300 components have, for this reason, been grouped into 10 groups (group A to J), where each group represents a certain gain within ±5%. The IL300 used for the Voltage Sensor is in group H, which represent a typical gain of 1.24. We can based on this calculate the adjusted value for R2.

$$R2 = \frac{R1}{1.24} = \frac{33k\Omega}{1.24} = 26.61k\Omega$$

We choose the value 27kΩ for R2 that is the best match from the E24 range of resistors.

The LM358 op amp must be supplied from the powerbus due to the galvanic separation. The required supply current includes the LED current for IL300 and the supply current for the op amp it selves. We estimate the maximum required current to be around 15 mA (which include some safe margin).

We choose a voltage regulator based on a zener diode to provide the supply voltage, which is a simple solution that works well for small amounts of load current.

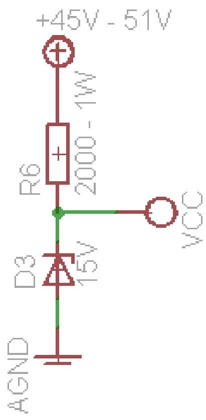


Figure 7 Voltage regulator base don zener diode

We choose a very common 15V zener diode, which should be rated to 15V * 15mA = 225mW.

There will be from 30V to 36V across the R6 resistor, which is the powerbus voltage minus the zener voltage. We can calculate the value of R6 as 30V / 15mA = 2kΩ.

R6 must be an effect resistor that can manage $(36V)^2$ / 2kΩ = 648mW. We can still manage this with ordinary 0.6W resistors by splitting R6 into two 1kΩ resistors.

Voltage Sensor Implementation

Figure 8 shows the implemented schematic with the calculated resistor values.

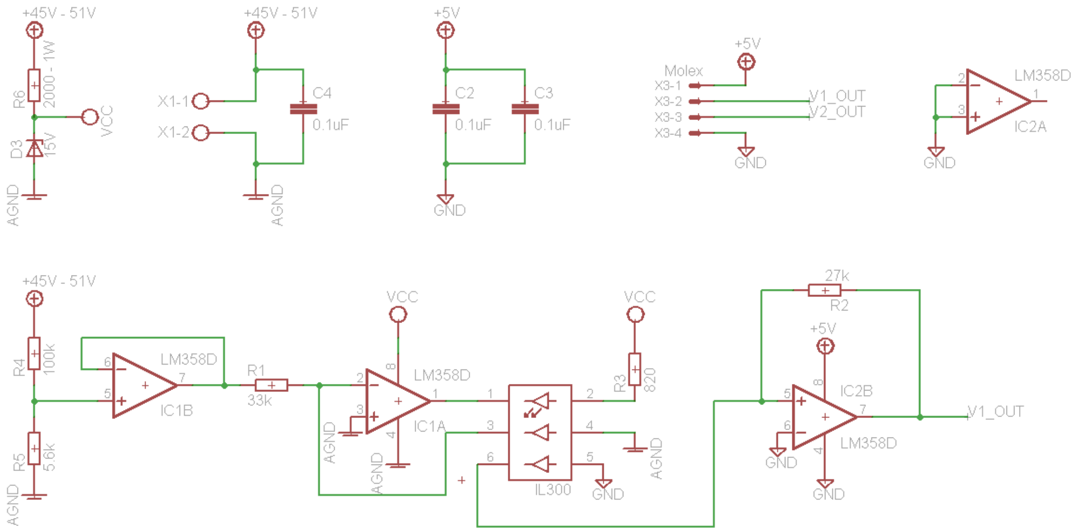
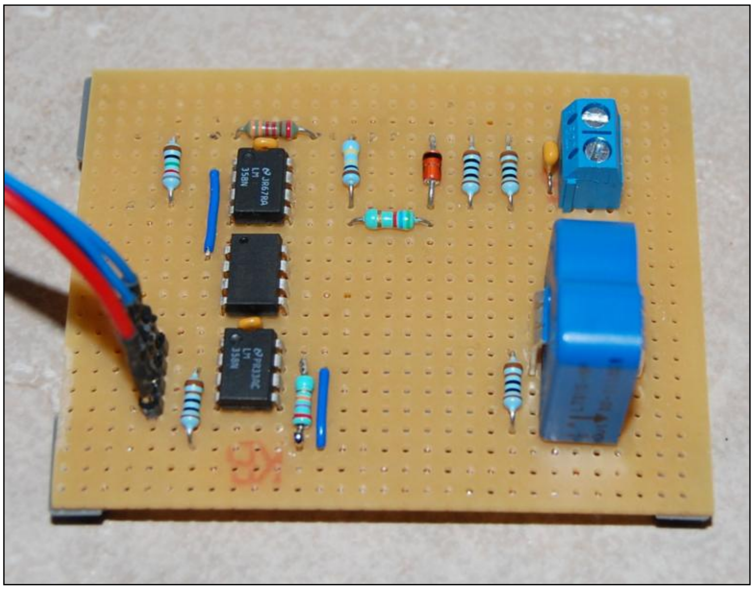


Figure 8 The complete schematic for the Voltage Sensor

The Voltage Sensor is realized on a PCB with pre fabricated holes along with the Current Sensor as shown in the picture below.



Gain and Level Adjustment using Differential Amplifier

Knud Baastrup

The analog outputs from the sensor circuits will require some gain and level adjustment in order to provide a range from 0 – 3.3V as required by the analog to digital converters located on the LPC2478 Development kit.

The level shift and gain adjustment can be performed by using a differential amplifier or preferably an instrumentation amplifier as shown in Figure 9.

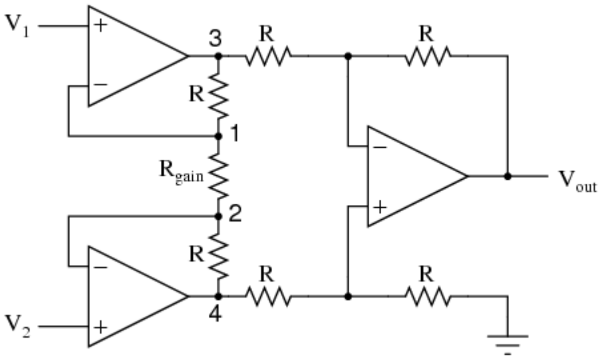


Figure 9 Instrumentation Amplifier

The instrumentation amplifier extends the differential amplifier by adding a non-inverting amplifier on each input to increase the input impedance and make it fully independent of the sensor circuits. It is as well designed so the two non-inverting amplifiers share the current through the feed-back resistors, which allow the gain to be adjusted on the single resistor R_{gain}.

Level Shift

The level shift (or offset adjustment) is performed by using a reference voltage that match the level shift to be performed. The reference voltage is established with a simple voltage divider combined with a potentiometer for calibration as shown in the complete diagram in Figure 12. The reference voltage is established with R1 and R2 and the calibration is performed with R3 and R4.

The current sensors and the voltage sensors will deliver an offset voltage of 2.5V, which then require a reference voltage of 2.5V in order to get a 0V output. We can implement the voltage divider with a current range in uA as the load from the instrumentation amplifier will be very low due to the high input resistance. We choose R1 to be 100kΩ, which will limit the current to be less than 90 uA (9V/100kΩ). Knowing R1, we can calculate R2 as below:

$$\mathrm{Vin} := 9\mathrm{V} \qquad \mathrm{Vo} := 2.5\mathrm{V} \qquad \mathrm{R1} := 100\mathrm{k}\Omega$$

$$\mathrm{R2} := \frac{\mathrm{R1}}{\frac{\mathrm{Vin}}{\mathrm{Vo}} - 1} = 38.461\mathrm{k}\Omega$$

We choose the value 39kΩ for R2 that is the best match from the E24 range of resistors.

R3 and R4 are used for calibration and allow some fine tuning of the reference voltage, which will be needed due to component tolerances and due to the 9V supply voltage that might not be exactly 9V.

R3 defines the maximum current that can be added or subtracted from the voltage divider when the potentiometer is in it outmost positions. Figure 10 and Figure 11**Fejl! Henvisningskilde ikke fundet.** shows a Thevenin equivalent for the voltage divider with the potentiometer, R4, in its two outmost pos

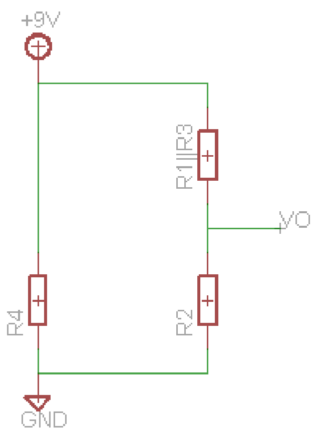


Figure 11 Thevenin to represent Vo_min

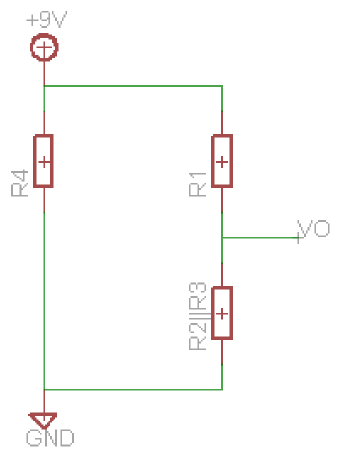


Figure 10 Thevenin to represent Vo_max

We can then as shown below calculate the value for R3 that allow Vo = 2.3V, which leave us 200mV for calibration in the downward direction.

$$Vin := 9V$$

$$R1 := 100k\Omega \qquad R2 := 38.3k\Omega$$

$$Vo_{min} := 2.3V$$

$$\text{Given} \qquad R2_{R3} = \frac{R2 \cdot R3}{R2 + R3} \qquad Vo_{min} = Vin \cdot \frac{R2_{R3}}{R2_{R3} + R1}$$

$$\text{Find}(R2_{R3}, R3) \text{ simplify } \rightarrow \begin{pmatrix} \frac{2300 \cdot k\Omega}{67} \\ \frac{880900 \cdot k\Omega}{2661} \end{pmatrix} = \begin{pmatrix} 34 \times 10^3 \\ 331 \times 10^3 \end{pmatrix} \Omega$$

We set R3 to be 330 kΩ, which is the best match from the E24 range of resistors. Knowing R3, we can as shown below calculate the corresponding maximum Vo in the upward direction to be Vo = 3V, which leave 500 mV for calibration.

$$Vin := 9V$$

$$R1 := 100k\Omega \qquad R2 := 38.3k\Omega \qquad R3 := 330k\Omega \qquad R1_{R3} := \frac{R1 \cdot R3}{R1 + R3}$$

$$Vo_{max} := Vin \cdot \frac{R2}{R1_{R3} + R2} \rightarrow 2.996 \cdot V$$

Gain Adjustment

The gain for the instrumentation amplifier shown in Figure 12 is given by below.

$$\frac{V1\_adc}{V1 - Vref} = \left(1 + \frac{2 \cdot R5}{R11}\right) \cdot \frac{R8}{R7} \quad \text{or simply} \quad \frac{V1\_adc}{V1 - Vref} = \frac{2 \cdot R5}{R11} \quad \text{if we choose R7=R8}$$

We set R7=R8=R9=R10= 100 kΩ and can then just focus on R5/R6 and R11 for the gain adjustment.

We would like the possibility to attenuate the signal if it is above the 3.3V that is the maximum voltage to be used for the A/D converters. We will therefore choose some resistors that are less than half the value of R11. If we choose a 10kΩ potentiometer for R11, we can calculate the value for R5/R6 that allows 10 dB attenuation, which should be sufficient.

$$A_{dB} = 20 \cdot \log\left(\frac{2 \cdot R5}{R11}\right) \quad \Leftrightarrow \quad R5 = \frac{10^{\frac{A_{dB}}{20}} \cdot R11}{2} = \frac{10^{\frac{-10}{20}} \cdot 10k\Omega}{2} = 2.5k\Omega$$

We choose R5/R6 = 2.40kΩ, which is the best match from the E24 range.

Implementation

Figure 12 shows the implemented schematic with the calculated resistor values for level shift and gain adjustment.

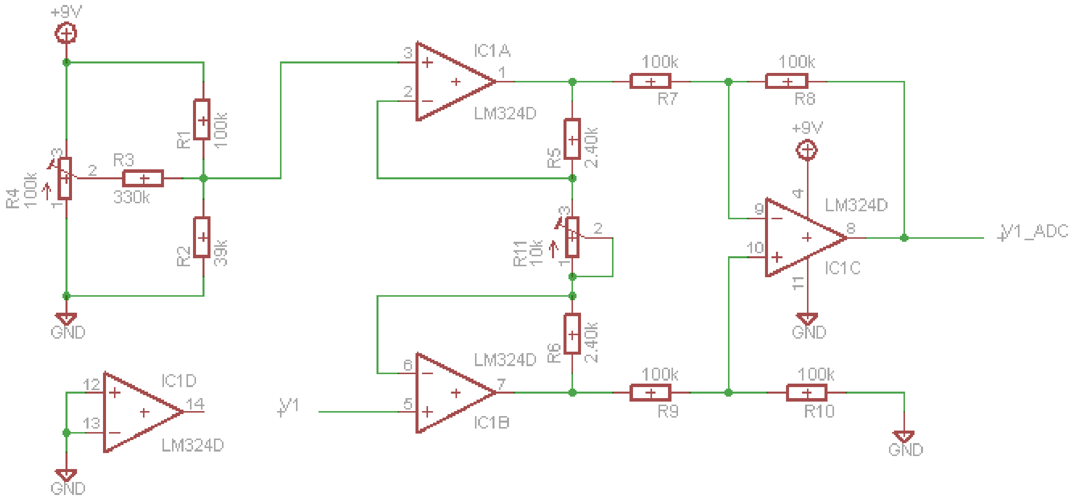


Figure 12 Schematic for the gain and level adjustment

We will have seven sensors that need some gain and level adjustment, which then require seven copies of the above circuit.

Figure 13 shows the PCB layout where two Gain and Level Adjustment circuits have been located on the same PCB. The PCB layout includes as well the bypass capacitors where a bypass capacitor have been added for each chip and as well for the incoming 9V supply.

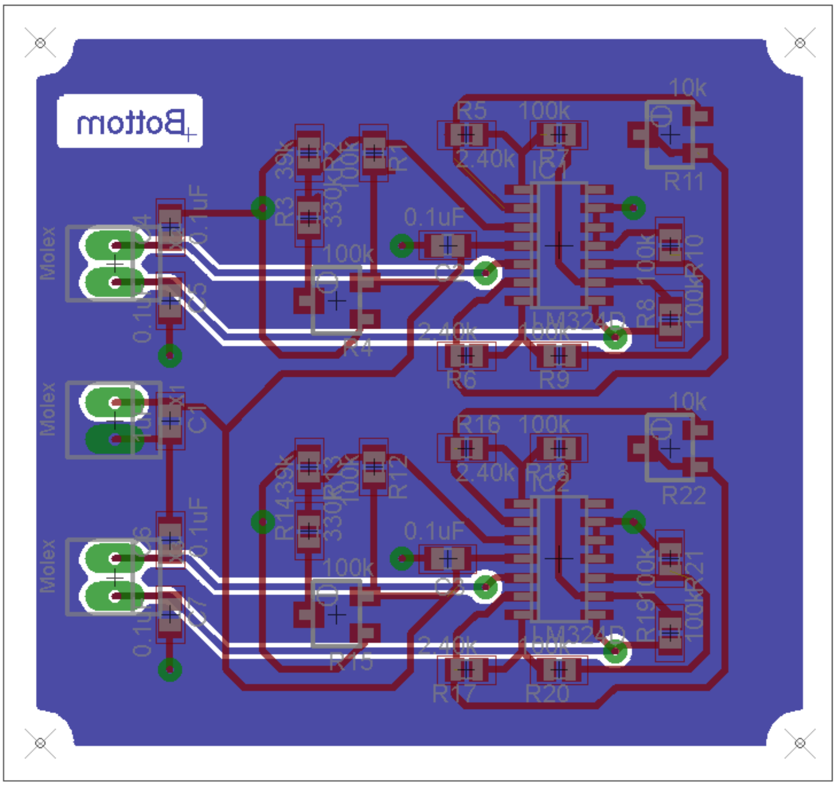


Figure 13 PCB layout for the gain and level adjustment

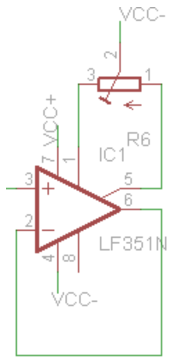
The following considerations were done during the PCB layout:

- Bypass/decoupling capacitors for the chips (C2 and C3) have been placed as close as possible to the supply pins. The capacitors will act as a local battery and prevent the voltage drop caused by the self inductance in the potential long wires. The capacitors will also decouple the high frequency noise.
- All incoming and outgoing connections have been placed in the same side to ensure that return current follow supply current and minimize loops/antennas.
- The incoming 9V power (upper left corner) is distributed to each circuit group in a star topology.
- Ground plane almost intact for a 2 layer PCB. All components requiring a ground have been directly connected to ground plane using a via.
- Sharp corners avoided, which could suffer acid damage.

Gain and Level Adjustment using Alternate ways

Dennis Thomsen

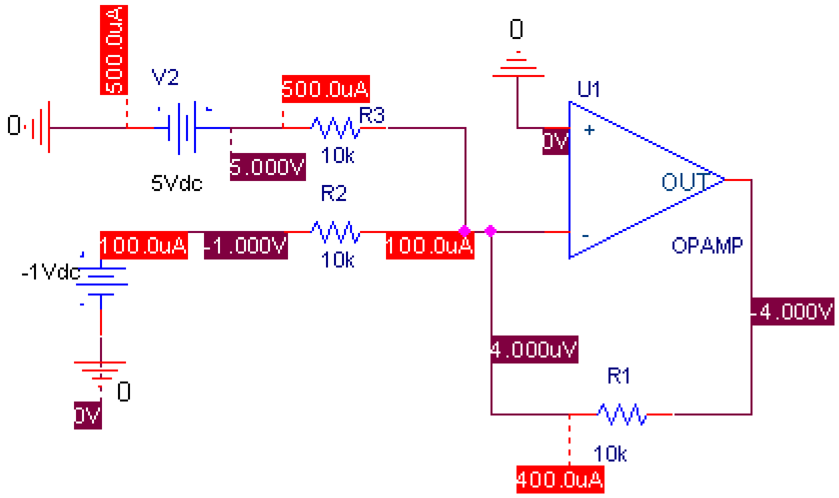
Before deciding on the use of the instrumentation amplifier for making the level adjustment need for the sensors, a number of diffident ways to archive this was discussed.



First tried was a non-inverting amplifier with a gain of 1 more commonly named a Voltage follower with NULL offset adjustment, to this was used the LF351N, but on the breadboard test showed that the NULL point could not move than 10% of ±Vcc, meaning that at least to opamp in series would be need as the system gets +9V from the hub and from that a ±Vcc of 4.5V would be the maximum without the use of a voltage doubler or something like that.

After this idea others were tested but they did not work as intended while being able to indeed drag 1 voltage down to 0V the output did not follow linear when the input voltage was raised, also around that time it was decided that the opamps used would be single supply, as to avoid making the negative supply needed for normal dual supply's ourselves.

Summing amplifier with a negative input voltage equal to offset voltage that need to be removed and by giving all the resistors on the inputs the same value, the amplification of each signal will be the same and with the feedback resistor also sharing that value, the amplification of the system will be one, though the output will be inverted, so it would need to go through a inverting amplifier.



Though this set could be made with a single supply opamp, the point of going with that rather than dual supply was to avoid having to make the negative voltage, which is why ultimately this design was not used.

Gain and Level Board for flow sensor to electrolyzer

Dennis Thomsen

Though it has been decided to the instrumentation amplifier for the different sensors obviously the level shift and gain needed for each sensor differ, so recalculations where needed to match the circuit to the sensor.

The flow sensor for the electrolyzer system is the FTAL020NU from SensorTechnics which has offset from 0.95 – 1.05 V DC and a maximum output of 4V DC.

Calculations where first carried out in mathcad to find the values of the resistors needed for this circuit, and following that the circuit was simulated in orcad using pspice, before it was built on a bread board.

Instrumentation amplifier Voltage divider 9V to 1V

$v_o := 1V \qquad v_{.in} := 9\ V$

$R1 := 1\Omega$

Starting out R1 is simply set to 1 so what will be found is simply the difference between R1 and R2

$$R2 := v_o = \frac{R2}{R1 + R2} \cdot v_{in} \ \text{solve}, R2 \ \rightarrow \frac{\Omega}{8}$$

Using that, we run though all the first values of the E12 series as R1 and see if what R2 values lies closes to the series as well

$$E12 := \begin{pmatrix} 1 \\ 1.2 \\ 1.5 \\ 1.8 \\ 2.2 \\ 2.7 \\ 3.3 \\ 3.9 \\ 4.7 \\ 5.6 \\ 6.8 \\ 8.2 \end{pmatrix} \qquad R1 := E12$$

R2·E12 = (Ω)

	0
0	0.125
1	0.15
2	0.188
3	0.225
4	0.275
5	0.338
6	0.413
7	0.488
8	0.588
9	0.7
10	0.85
11	1.025

0.15 is E12 value so using a multiplicity of 10 of 1.2 and 0.15 it should have a very good voltage divider

$R1 := 12k\Omega \qquad R2 := 1.5k\Omega$

$$v_o := \frac{R2}{R1 + R2} \cdot v_{in} \qquad v_o = 1\,V$$

$$I_{.1} := \frac{(v_{.in} - v_{.o})}{R1} = 0.667 \cdot mA \qquad I_{.2} := \frac{v_{.o}}{R2} = 0.667 \cdot mA$$

This voltage divider will include the potentiometer as well as R3, as seen under Level Shift section of Gain and Level Adjustment using Differential Amplifier

Instrumentation amplifier Gain calculations

$$\frac{V_{out}}{V_2 - V_1} = \text{Gain} = \left(1 + \frac{2R_1}{R_{Gain}}\right) \cdot \frac{R_3}{R_2}$$

as an Gain of 1 is need for this sensor then $\left(1 + \frac{2R_1}{R_{Gain}}\right) \cdot \frac{R_3}{R_2} = 1$

meaning that idealy that if the value inside the parentens are 2 then $\frac{R_3}{R_2} = 0.5$

That is R2 is double the value of R3

For this to work then $\frac{2R_1}{R_{Gain}} = 1$ that is RGain should be double the Value of R1

of couse from the E12 series there cant be found perfect values for R3 and R2 but as a potentiometer is well suit for use as RGain, it can be adjusted to make up for this differens

$R_2 := 6.8k\Omega \qquad R_3 := 3.3k\Omega \qquad R_1 := 10k\Omega$

$$\left(1 + \frac{2R_1}{R_{Gain}}\right) \cdot \frac{R_3}{R_2} = 1 \text{ solve}, R_{Gain} = 18.857142857142857143k\Omega = 18.857 \cdot k\Omega$$

Instrumentation amplifier Voltage divider 4V to 3.3V

$v_o := 3.3V \qquad v_{in} := 4V$

$R1 := 1\Omega$

$$R2 := v_o = \frac{R2}{R1 + R2} \cdot v_{in} \text{ solve}, R2 \rightarrow 4.7142857142857142857\Omega$$

The same method used for finding the Offset voltage divider resistors values is used here

$$E12 := \begin{pmatrix} 1 \\ 1.2 \\ 1.5 \\ 1.8 \\ 2.2 \\ 2.7 \\ 3.3 \\ 3.9 \\ 4.7 \\ 5.6 \\ 6.8 \\ 8.2 \end{pmatrix} \qquad R1 := E12 \qquad R2 \cdot E12 = \begin{array}{c|c} & 0 \\ \hline 0 & 4.714 \\ 1 & 5.657 \\ 2 & 7.071 \\ 3 & 8.486 \\ 4 & 10.371 \\ 5 & 12.729 \\ 6 & 15.557 \\ 7 & 18.386 \\ 8 & 22.157 \\ 9 & 26.4 \\ 10 & 32.057 \\ 11 & 38.657 \end{array} \cdot \Omega \qquad R2 := \begin{pmatrix} 4.7 \\ 5.6 \\ 6.8 \\ 8.2 \\ 10 \\ 12 \\ 15 \\ 18 \\ 22 \\ 27 \\ 33 \\ 39 \end{pmatrix}$$

all the different R2 values were assigned the E12 value closest to them,

$$v_o := \frac{R2}{R1 + R2} \cdot v_{in}$$

$$v_o = \begin{array}{c|c} & 0 \\ \hline 0 & 3.298 \\ 1 & 3.294 \\ 2 & 3.277 \\ 3 & 3.28 \\ 4 & 3.279 \\ 5 & 3.265 \\ 6 & 3.279 \\ 7 & 3.288 \\ 8 & 3.296 \\ 9 & 3.313 \\ 10 & 3.317 \\ 11 & 3.305 \end{array} \cdot V$$

Picked the 9'th output so $R1 := 56k\Omega$ $R2 := 270k\Omega$

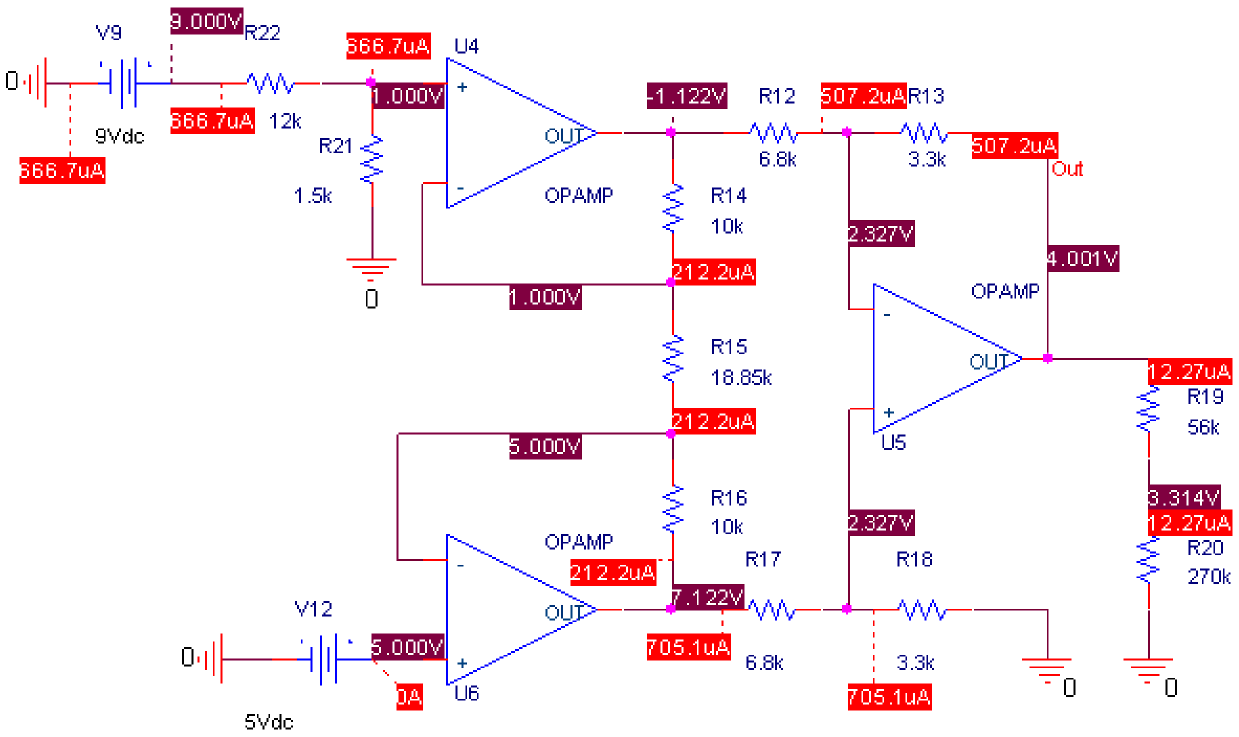
$$v_o := \frac{R2}{R1 + R2} \cdot v_{in} \qquad v_o = 3.313 \cdot V$$

$$I_1 := \frac{(v_{in} - v_o)}{R1} = 0.012 \cdot mA \qquad I_2 := \frac{v_o}{R2} = 0.012 \cdot mA$$

There is no need for calibration on this voltage divider as the gain of the Instrumentation amplifier can be adjusted.

Testing of Circuit

Using Orcad below circuit was built and then run, as can be seen the circuit behaves as expected



However when built on breadboard the circuit did not act as expected, the output from the circuit was lower than expected with a maximum lying at around 2.8V, the first change done was to replaced all the E12 resistors in the amplifier part with E96 resistors that had 1% error margin rather than the 5%, the first resistors had. Though it didn't help much a few millivolts at best, the difference between the resistors, made it unlike that the system could get a gain of precisely 1, as such the 6.8k and 3.3k resistors where giving the same values, in the hope of setting $\frac{R_2}{R_3}$ as close to one as possible and then getting $\frac{2R_1}{R_{Gain}}$ as close to 0 as possible after all a overall gain of 1.001 or even 1.01 would be good enough.

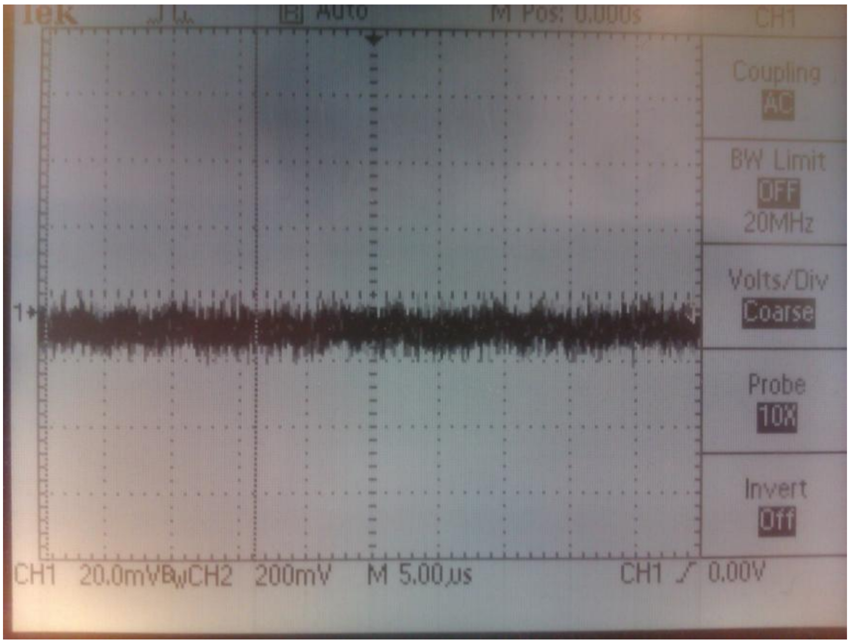
$R_2 := 100k\Omega \qquad R_3 := 100k\Omega \qquad R_1 := 10k\Omega$

$$\left(1+\frac{2R_1}{R_{Gain}}\right)\cdot\frac{R_3}{R_2} = 1.01 \,\text{solve}, R_{Gain} \;\rightarrow\; 2000.0\cdot k\Omega = 2\times 10^3\cdot k\Omega$$

By rerunning the calculations it can be seen that a potentiometer of about 2MΩ would be need to set the output at 3.3 V but in actually as 4.7M potentiometer was needed for that, in this setup the voltage could not go under 3.3V, as such the 10K Ω resistors where change to 2.2K Ω which did allow to output to be calibrated from 2.292V to 5.7V

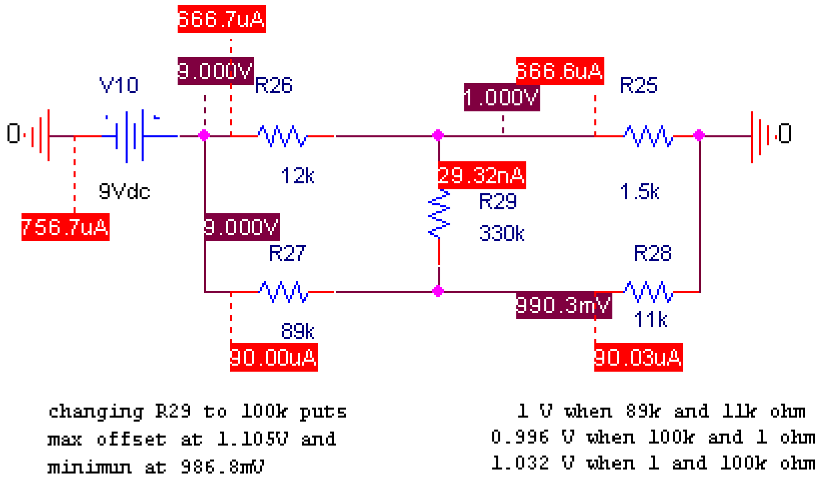
The Null offset can't go all the way down to 0V, as it is a single supply opamp, more precisely 0.015V is the lowest output.

The output was also checked for noise using an oscilloscope



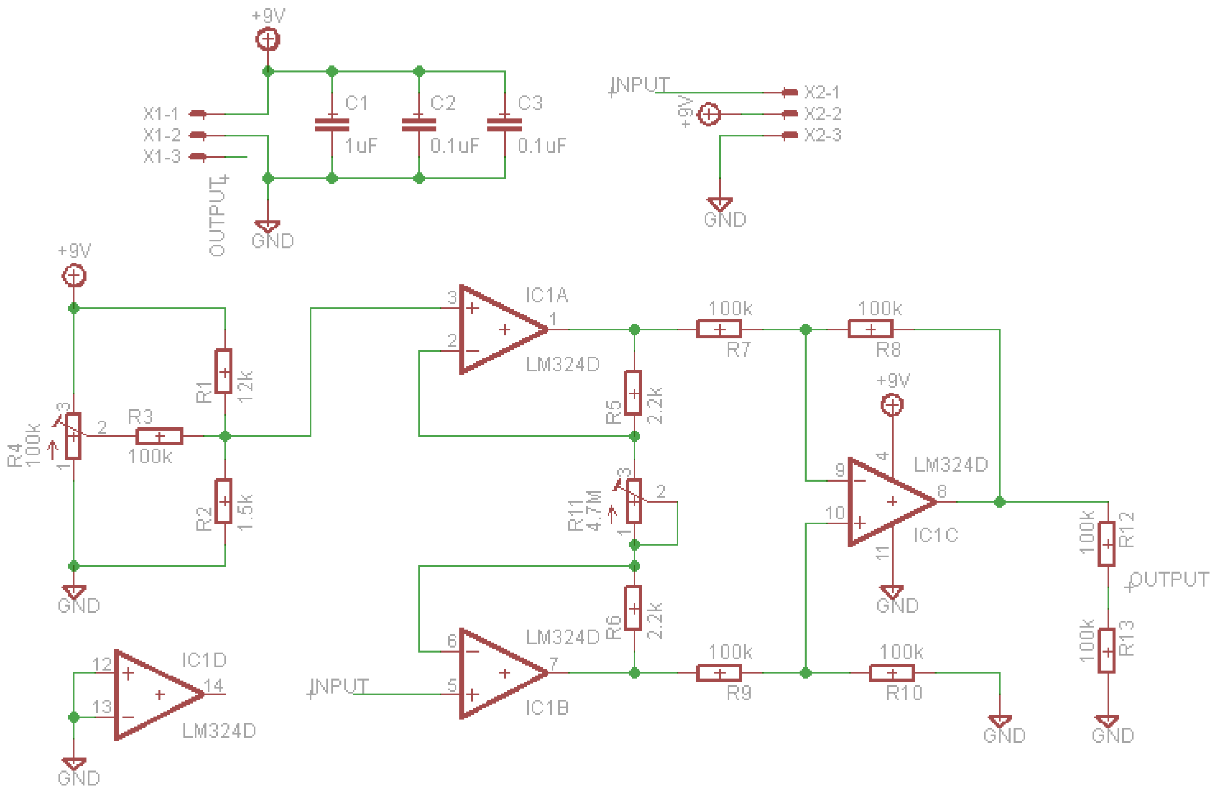
As can be seen, the signal noise is very high frequent and it don't exceed an amplitude of 10mV, getting the circuit of breadboard, and adding a couple of capacitors for decoupling, in addition to the EMC review that circuit is getting, should ensure it don't become a problem.

Further simulation was done on the offset voltage divider, to see how the potentiometer of 100k Ω affected the output



Which brought with a change of the resistor value of R29, as called in this simulation, to give a higher calibration range.

Current Version can be seen here under



Improvement can be made to the circuit by using a rail-to-rail (r2r) single supply opamp, rather than the LM324 currently used, as the output can go much close to 0V output than normal opamps(normally this is done by using MOS transistors in opamp rather than voltage controlled transistors), We are considering ordering home a few TSV624, which is a r2r opamp, to test how it could improve the circuits, it would be an easy swap as it as the same pin layout as the LM324.

Verification

Knud Baastrup

The verification for timebox 2 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 2

Test Case ID	**2.1**	
Description	Verify that it is possible to read out the actual input current, input voltage and output flow for the electrolyzer. The test should include typical, maximum and minimum value.	
Preconditions	Electrolyzer started.	
Command/Action/Steps	**Expected Result/Verification**	**PASSED**
`$ h2 3 2`	The actual input current value printed on console with a value within ± 1% of control measurement done with Fluke 73 Series II Multimeter. `$ xxxx mA`	TBD
`$ h2 3 3`	The actual input voltage value printed on console with a value within ± 1% of control measurement done with Fluke 73 Series II Multimeter. `$ xxxxx mV`	TBD
`$ h2 3 4`	The actual output flow printed on console with a value within ± 1% compared to measurement performed with graduated cylinder. `$ xxxxx mL/s`	TBD
Repeat above steps for typical, max. and min value.		

Test Case ID	**2.2**	
Description	Verify that it is possible to read-out the current, voltage and flow value when the electrolyzer is stopped.	
Preconditions	Electrolyzer stopped.	
Command/Action/Steps	**Expected Result/Verification**	**PASSED**
`$ h2 3 2`	`$ 0 mA`	TBD
`$ h2 3 3`	The actual input voltage value printed on console with a value within ± 1% of control measurement done with Fluke 73 Series II Multimeter. `$ xxxxx mV`	TBD

`$ h2 3 4`	`$ 0 mL/s`	TBD

Test Case ID	**2.3**	
Description	Verify that maximum value is returned when input current, input voltage and output flow is above the specified maximum values.	
Preconditions	Electrolyzer started.	
Command/Action/Steps	**Expected Result/Verification**	**PASSED**
`$ h2 3 2`	`$ 10000 mA` Even though the actual current value is above 10A. (verified with Fluke 73 Series II Multimeter).	TBD
`$ h2 3 3`	`$ 51000 mV` Even though the actual voltage value is above 51V. (verified with Fluke 73 Series II Multimeter).	TBD
`$ h2 3 4`	`$ 20000 mL/s` Even though the actual flow value is above 20 L/s. (verified with graduated cylinder).	TBD

Product Acceptance tests

The product acceptance tests failed due to an offset that we could not explain during the deployment. The issue will be investigated and the tests repeated for next deployment.

EMC REVIEWS

Lasse Lykkegaard

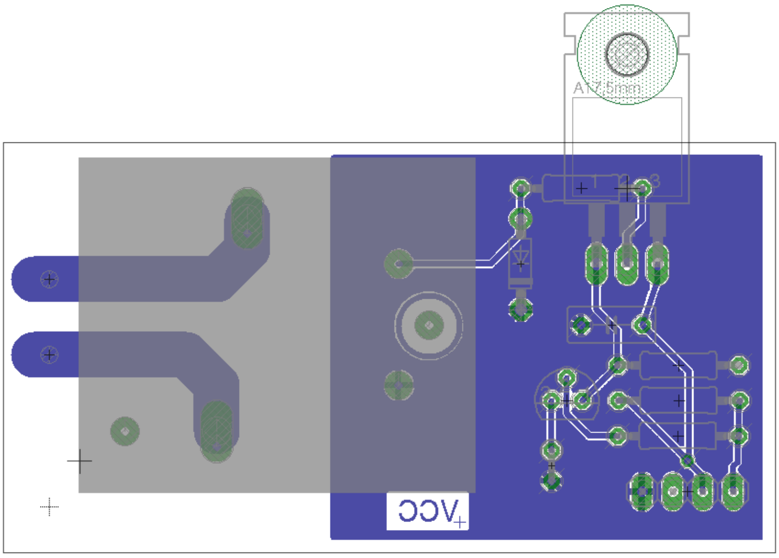


Figure 14: PCB for Relay circuit

In the first edition, resistors were placed in a 45° angle, to safe space. This is bad design practice, due to the fact that soldering machines will not be able to maintain a good quality soldering, the tin solder would be stuck in the angles.

It was noticed that the diode was placed close to the coil, but the connection from the external-pin should not be routed to the coil-pin and then to the diode, it should be the other way around. Though is the correction meaningless after a complete VCC-plane was added. The other layer is a GND-layer, making a small uniform electric field between planes. The planes are not laid out beneath the relay. When switching a large current the change in the magnetic field around the current could propagate into the controlling electronic which compared to a 60A circuit is of higher impedance.

With no high frequency communication, there was no need for other adjustments. The high-current connection is galvanic separated from the rest, therefore no single-point entry. On the board is needed.

Sharp edges are avoided as much as possible, to avoid searing failures. There is no high speed frequencies present on the board that is affected by 90° bend.

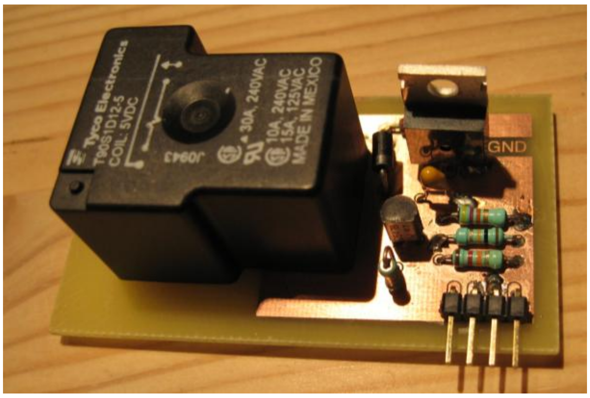


Figure 15: The final relay circuit

Dynamic Website

Lasse Lykkegaard

The overall structure of the webpage has not altered much since the skeleton was made. The MySQL logon script is put aside outside public, Figure 16. Seen from an ftp connection and from the team4-user's point of view the script is nowhere to be seen. A chroot has forced the root of the user of team4 to /var/www/lene-lasse.dk/ The script is located in /var/www/ and therefore inaccessible. Since the server is running UNIX a script owned by root and with permissions set to owneronly would do the same. When a server executes php code it is done as root.

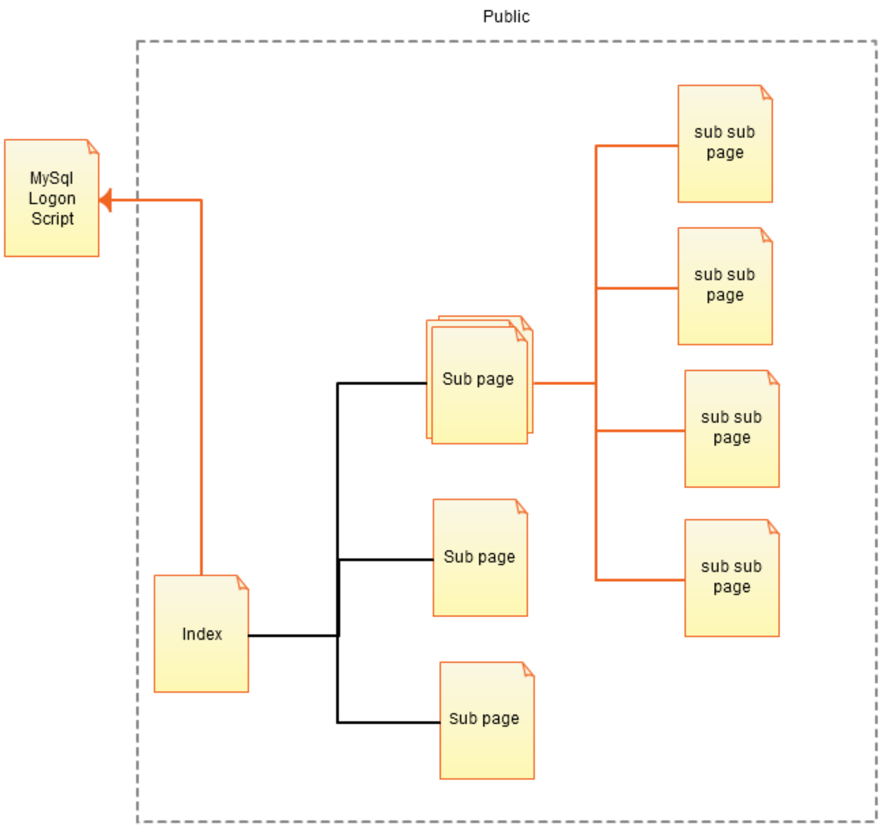


Figure 16 – Dynamic web configuration

All pages have been edited into active php-files. Making an *include* of common elements such as footer box, header-box etc.

In the previous non-active versions, you needed to implement copies of menus and footers on ALL pages. The work of making a small correction on every page was very tedious.

Our php pages follow the on Figure 17 shown generation.

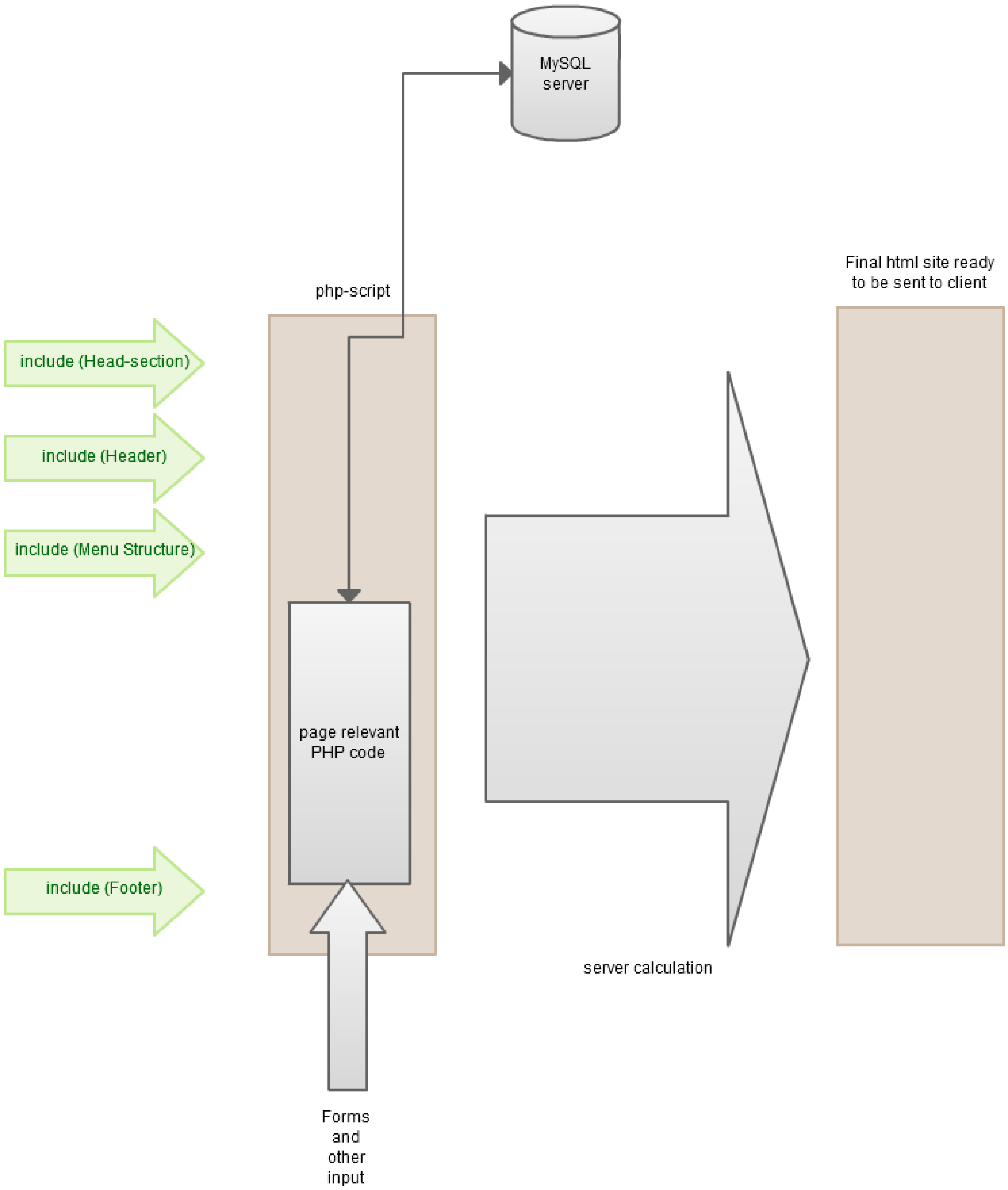


Figure 17– PHP generation

We need to use the tools available at the current php/SQL-programming experience, so the code tends to be cumbersome and if expanded in a large scale: slow and ineffective.

The page of the H2 Production status pages are made active and interact with our MySQL-server. The MySQL server is getting live data from the hydrogen system (when it is turned on).

One of my major problems was to make SQL multiply two rows by each other, when not know how many times it should do it…

SELECT EXP(SUM(LOG(data))) AS power

does the job since data is a colum and log (loge or ln)is taken to all samples, alle these are summed and raised of e. With this syntax we exploit the rules of logarithmic and powers.

One place a SQL-statement caused many troubles (other experts still working on why), and we therefore needed to calculate average "by hand." But to protect for overflow of large numbers the adding arithmetic way of calculating is used. Summing a 1000 floats and dividing by 1000 may cause overflow.

```
$avg= ($i*$avg + $new_data']])/ ($i+1);
$i++
```

This way we can implement the average one sample at a time.

REFERENCES

- EUDP: http://www.eudp.net
- Datasheet: Current Transducer LTS 15-NP
- Datasheet: Voltage Transducer LV 25-P
- Datasheet: Linear Optocoupler IL300
- Application Note: Linear Optocoupler IL300
- W3schools.com: www.w3schools.com