UiO : Department of Physics
University of Oslo

A Measurement System For Attitude Determination on Sounding Rockets

Master thesis

Bent Anders Furevik

August 8, 2013

UNIVERSITAS OSLOENSIS
MDCCCXI

Abstract

This thesis describes the development of a new integrated sensor measurement system for attitude determination of sounding rockets. The sensor data are to be processed using the attitude determination software developed by Bekkeng (2007), in order to perform post flight data fusion and attitude determination. Measurement principles, sensors, intrument design, calibration and test results are reviewed.

The system comprises a data acquisition (DAQ) board collecting data from two digital sun sensors, a 3-axis rate gyro module, a 3-axis external magnetometercard and a 3-axis magnetometer integrated on the DAQ card.

The two digital sun sensors can be mounted on each side of the top deck (under the ejected nose cone), and the rate gyro module can be mounted inside the payload section of the rocket. The external magnetometer board should be mounted on a deployable boom, in order to avoid disturbances from the rocket due to currents in the rocket harness and the rocket skin.

Acknowledgements

This work was carried out in the period from August 2011 to June 2013, under the supervision of Associate professor Torfinn Lindem and PhD candidate Tore André Bekkeng.

I would like to thank Torfinn Lindem for his constant support and enthusiasm. You have always been more than helpful, both during my bachelor degree and the work with this thesis.

Tore André Bekkeng, your help and motivation has been invaluable. I'm very grateful for everything.

Thanks to the guys at the Electronics lab for all the help with PCB production. I would also like to thank Espen Trondsen for sharing his seemingly infinite experience. Thanks to the guys at the mechanical workshop for all the help with the mechanical designs.

Thanks to Jan Kenneth Bekkeng for your support and sharing your experience. I would also like to thank Ivar Jørstad for his guidance and support.

To Cecilie, Elling, David and Andres at room 303, Lars Jørgen and Espen; thanks for all help, discussion and nonsense during this time.

At last I would like to thank my girlfriend Siri for sticking up with me, even when times have been though.

Contents

Nomenclature

AA	Anti-Aliasing
ADC	Analog to Digital Converter
AMR	Anisotropic Magnetoresistance
ARR	Andøya Rocket Range
BGA	Ball Grid Array
CCD	Charge Coupled Device
$CMOS$	Complementary Metal Oxide Semicondutor
$COTS$	Commercial Off-The-Shelf
$CPLD$	Complex Programmable Logic Device
DAQ	Data Acquisition Card
DC	Direct Current
DMM	Direct Current
DRY	Don't Repeat Yourself
DSS	Digital Sun Sensor
E^2PROM	Electrically Erasable Programmable Read-Only Memory
$FBGA$	Fine Ball Grid Array
FOV	Field of View
$FPGA$	Field Programmable Gate Array
$G-R$	Generation-recombination

GND	Ground
GPS	Global Positioning System
HF	High Frequency
ICI	Investigation of Cusp Irregularities
IGRF	International Geomagnetic Reference Field
JTAG	Joint Test Action Group
LDO	Low Drop Out regulator
LF	Low Frequency
LSB	Least Significant Bit
LVDS	Low Voltage Differential Signalling
MEMS	Micro-Electro-Mechanical Systems
MSB	Most Significant Bit
PCB	Printed Circuit Board
PCM	Pulse Code Modulation
PSD	Position Sensitive Detector
PSRR	Power Supply Rejection Ratio
PWM	Pulse Width Modulation
SAR	Successive Approximation Register
SHA	Sample and Hold Amplifier
SRADS	Sounding Rocket Attitude Determination System
TQFP	Thin Quad Flat Package
UiO	University of Oslo
VHDL	Very high speed integrated circuit Hardware Description Language
VI	Virtual Instrument

List of Figures

Chapter 1

Introduction

This thesis describes the development of a system to measure the 3-axis attitude of a sounding rocket. The measurement data from the attitude sensors will be transmitted to the ground telemetry station by the rocket encoder. Since the attitude of the spacecraft is not controlled during the flight the sensor data are fused in the SRADS post processing estimation algorithms.

The SRADS (Sounding Rocket Attitude Determination System) attitude determination algorithm was developed by Bekkeng (2007). Due to a new sun sensor developed in this work, the original sun sensor model and the corresponding mathematics in the Kalman filter estimator must be modified. However, updating the attitude determination estimator is not part of this work.

1.1 Background and Motivation

Every spacecraft needs to know its attitude (orientation in space) in addition to the position. Attitude information is also needed in order to transform scientific measurements in the reference frame of the spacecraft into a more meaningful frame of reference, e.g. an Earth-fixed reference frame.

The motivation for this work was to make an improved measurement system for attitude determination of the ICI-4 (Investigation of Cusp Irregularities) rocket payload and future rocket launches by the University of Oslo. Measurements made by this system are to be used with the updated SRADS software estimation algorithms produced in the work of Bekkeng (2007). ICI-4 wil carry several scientific experiments in the rocket payload, e.g. a multi-Needle Langmuir Probe (m-NLP) and an Electric Field Instrument developed at the University of Oslo. This sounding rocket is scheduled to launch from Spitsbergen in November/December 2013.

1.2 Goals of the present work

The goal of this thesis was to develop a new integrated measurement system with improved sensors to replace the currently used measurement system developed in the work by Bekkeng (2007). To achieve this, a new sun sensor was developed, due to mechanical accuracy limitations of the old sun sensor design. It was desirable to investigate if a solution based on a Position Sensitive Detector (PSD) was feasible. Novel sensor technology should be utilized in the new measurement system, including a high precision 3-axis rate gyro configured for high angular rates.

Reducing the size, weight and power consumption was a secondary goal. As the currently used system comprises separate sun sensor, magnetometer and rate gyro instruments, reductions should be more than achievable.

Because of magnetometer problems on previous flights it was also a goal to have a magnetometer on a deployable boom, away from the rockets disturbances. While development of the external magnetometer was not part of this thesis, communication with the data acquisition board was implemented.

1.3 Outline

Chapter 2 - Sensors for attitude determination

An introduction to different sensors used for attitude measurements, with a particular focus on the sensor technologies actually used.

Chapter 3 - Sun sensor background and Theory

A general overview of measurement principles for the sun sensor. It also includes a coverage of noise factors and analysis.

Chapter 4 - System Design

A description of the system requirements, choices made during development and the actual design. The logic design and final PCB layouts are also covered.

Chapter 5 - Mechanical Design

A brief coverage of the mechanical designs for the system.

Chapter 6 - Performance analysis and calibration

An analysis of the systems performance and noise properties. Included is also a calibration routine for the sun sensor and characterization of the analog input stage.

Chapter 7 - Comparing the new vs old systems

A comparison is presented of the measurement system developed in this thesis and the old system of sensors. Weight, cost, power consumption and ease of manufacturing and calibration is taken into account.

Chapter 8 - Conclusion

A summary and critical review on the development process and the final product of the thesis. Suggestions for further improvements in the future are given as well.

Chapter 2

Sensors for attitude determination

To determine the attitude of the spacecraft it is necessary to fuse measurements from several sensors. A general overview of different sensors suitable for these measurements is given in this chapter.

Sensors used for attitude determination include both attitude sensors and inertial sensors. Inertial sensors utilize an internal inertial mass, making them non-dependent on external signals. Attitude sensors are dependant on an external reference, e.g. a magnetic field or the position of the sun. (Wertz, 1978)

2.1 Gyroscope

The operation of a gyroscope is based on the fundamental principle of conservation of angular momentum: *In any system of particles, the total angular momentum of the system relative to any point fixed in space remains constant, provided no external forces act on the system.* (Fraden, 2004)

The classic gyroscope consists of a disc suspended inside two rings called gimbals to give it three degrees of freedom. This allows it to detect rotation in the spin axis of the disc. These units are very accurate, but they are large, expensive and mechanically complex.

MEMS (Micro-Electro-Mechanical Systems) units, however, are mounted rigidly (strapdown) to the object being measured. Usually the term rate gyro is used, since these gyros measure angular rate (angular velocity) and not an angle.

Strapdown gyros can be based on optical (ring laser or fiber optic gyro) principles, but these are relatively large, heavy and expensive. Gyros based on MEMS technology are usually based on the Coriolis effect (Coriolis, 1835), since it is difficult to create a rotating element using MEMS technology. A MEMS rate gyro has a mass suspended in a frame vibrating in one axis, illustrated in Figure 2.1, creating a velocity of the inertial mass. When a rotation is applied the mass will experience a force that is orthogonal to the velocity vector v and the angular velocity (rate) vector ω, given by Equation (2.1).

$$v = w \cdot r \tag{2.1a}$$

$$F = 2m\omega \times v \tag{2.1b}$$

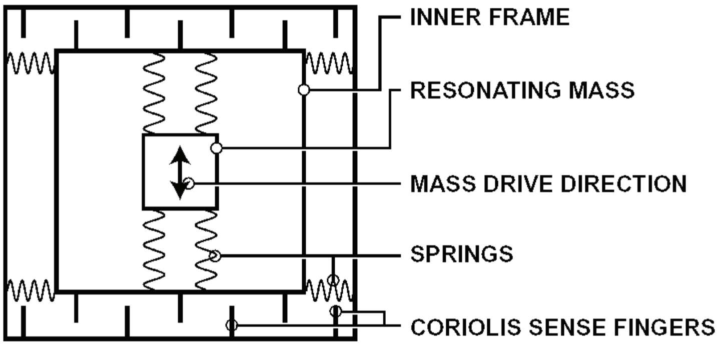


Figure 2.1: MEMS gyro principle. (Zumbahlen, 2007)

A practical advantage of a gyro is that it doesn't have to be located in the spacecrafts geometrical center to make correct measurements. The angular rate is the same even if the gyro is located near the edge of the spinning spacecraft. However, this requires that the rate gyro design is insensitive to translation or rotational (centripetal) acceleration. A vehicle with high rotational dynamics, such as a souning rocket, requires rate gyro capable of measuring high angular rates. Rate gyros for high rate applications, above 400°, is treated as military equipment and have export restrictions.

2.1.1 Sensonor STIM210

For this thesis the STIM210 gyro has been chosen (further discussed in Section 4.4). It employs the ButterflyGyro concept of Sensonor, which basically means that it uses two vibrating masses operated in a balanced anti-phase movement. This balanced vibration of both the excitation mode and the detection mode makes the gyro insensitive to environmental vibrations and limits offset-causing effects (Kvisterøy et al., 2004). The deflection is measured using a capacitive measurement technique, which is a common principle used in MEMS gyros.

The sensor is also temperature compensated, which is important to avoid temperature drift affecting the measurements.

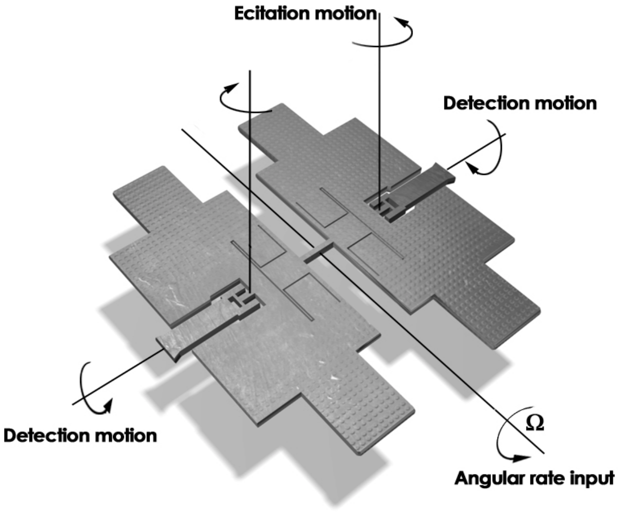


Figure 2.2: Functional principle of Sensonors Butterfly gyro concept. (Kvisterøy et al., 2004)

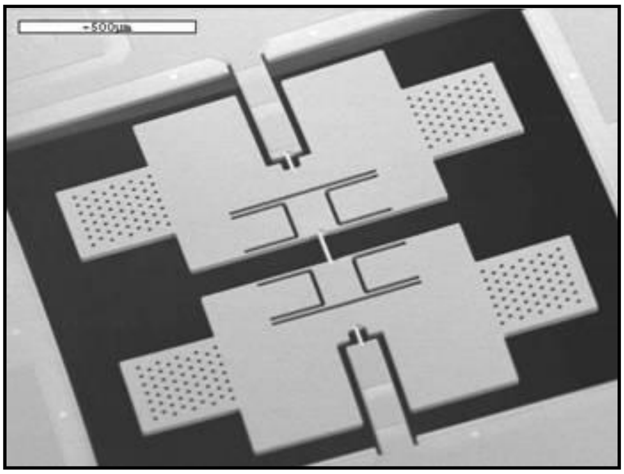


Figure 2.3: Structure of the Sensonor SAR150 die. (Sensonor, 2010)

2.2 Magnetometer

Magnetometers are widely used as spacecraft attitude sensors (Wertz, 1978). A magnetometer measures the magnetic field vector and they are known to be reliable, lightweight and having low power consumption. They are also operable over a wide temperature range and have no moving parts, which is a big advantage in a rocket considering the large g-forces and vibration the sensors are subjected to during launch.

The Earth's magnetic field strength decreases by an $1/r^3$ dependence with distance. This limits the use of magnetometers in spacecrafts to altitudes below 1000 km because internal magnetic fields of the spacecraft may start to dominate the measurements above that. (Wertz, 1978) As the ICI sounding rockets will fly to approximately 350 km this should not pose a significant problem for the application.

The downside when using a magnetometer is that the measurements must be compared to a model of the Earth's magnetic field. A model called International Geomagnetic Reference Field (IGRF) is one of the most commonly used for this (Finlay et al., 2010). This is a standardized model with an accuracy of 0.1 nT. Today's model is a 13th order formula. Revised every fifth year, the model is now in its 11th generation and is widely recognized as a standard.

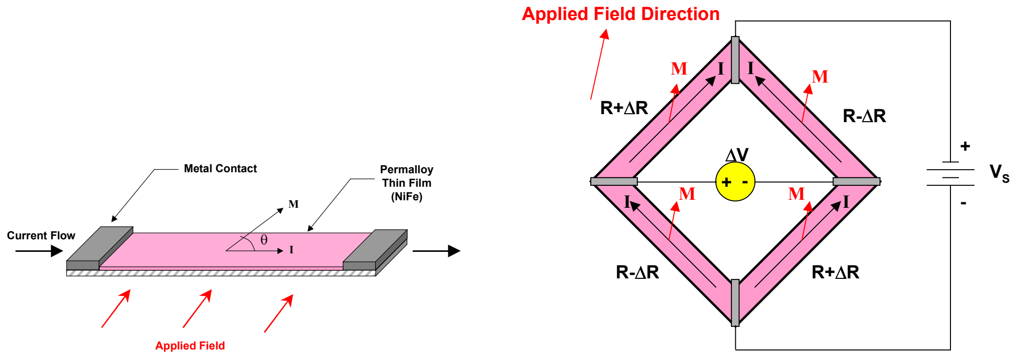
The field contributions of the Earth's crust, current flows in the ionosphere and magnetosphere, and induced currents in the sea and the ground are not included in the model, as they are difficult to predict. Since the ICI sounding rockets are launched into strong ionospheric events, the IGRF model error must be assumed to be much larger. The error can probably be in the range of several hundred nT. (Bak, 1999)

2.2.1 Magnetoresistive Sensor (Honeywell HMC5983)

The Honeywell HMC5983 has been chosen for this thesis. It is based on Anisotropic Magnetoresistive (AMR) technology. Measured data are provided digitally through a 12-bit ADC, so that no external conversion is required.

When a magnetic field is applied perpendicular to the current flow in a thin strip of ferrous material, there is a change in the strips resistance (McGuire and Potter, 1975). This is called anisotropic magnetoresistance.

In Honeywells sensors a ferrous material called Permalloy (NiFe) is used to create four magnetoresistive elements. These are placed in a diamond shape forming a wheatstone bridge to maximize sensitivity, as seen in Figure 2.4(b). The four magnetoresistors have the same resistance. When a magnetic field



(a) AMR Element (b) AMR Bridge

Figure 2.4: Magnetoresistive sensor properties (Honeywell, Honeywell)

is applied, a differential voltage can be measured across the side contacts of the bridge. This voltage is a function of supply voltage, magnetization ration and the angle between the magnetization and the current flow in the sensor.

2.3 Horizon sensor

When operating near the Earth, the Earth is the second brightest celestial object. Compared to the sun and the stars, which can be approximated as point sources, the Earth is an extended target. A sensor detecting only the presence of the Earth is not sufficient in most applications. Nearly all horizon sensors are therefore designed to detect the Earth's horizon. For a spinning spacecraft, e.g. a spin stabilised rocket, the sensor will detect a rapid change in intensity crossing the Earth's horizon (Wertz, 1978).

According to Wertz (1978), the Earths horizon is least ambiguous in the infrared spectral region around 15 μm. Such a long wave infrared sensor with good performance could require cooling, making the sensor large, heavy and expensive.

2.4 Star sensor

A star sensor, or star tracker, comprises an image sensor to image the sky and a star catalog to identify the stars in the sensors field of view. Although star sensors are known to be the most accurate attitude sensors, they are very sensitive to large angular velocities. Star sensors typically have an update rate of less than 5 Hz and operate up to an angular velocity of a few degrees per second (Wertz, 1978). This makes them pretty much useless for a spin-stabilized rocket with a typical spin frequency of 4 RPS. They are also very complex, large and expensive.

Liebe et al. (2004) describes a stellar gyroscope capable of operating at angular velocities of 50°/s. Although this is much higher than the traditional star trackers, it is still much lower than the requirements of an ICI sounding rocket.

2.5 GPS (Global Positioning System)

Although originally designed to determine position, velocity and time, the Global Positioning System can also be used for three-axis attitude determination. The most common technique for precise GPS attitude determination uses an array of three or more antennas fixed to the vehicle. By measuring the carrier-phase difference between two antennas, a basic measurement of the range between the antennas and a satelite can be determined. For a three-axis attitude determination, a minimum of three antennas and two GPS satelites are needed. (Psiaki, 2006)

An accuracy of 4° or better was achieved by Psiaki (2006) using only two GPS satelites and three antenas, with a 1 m baseline.

2.6 Sun sensor

A sun sensor measures the angle of the incident sunlight and is the most used reference sensor on spacecrafts. (Wertz, 1978)

To measure the angle of the sun relative to the spacecraft, a sensor that is able to detect the angle of the sunlight is needed. The Sun can be treated as a point-source at infinite distance, producing parallel rays of light. Its angular radius i approximately 0.53° seen from the Earth. As the intensity of the sun is much higher than any other object in the vicinity of the Earth, it is simple to discriminate. This means that simple equipment can be used for the measurements, without having to discriminate between different sources.

Since the development of the new digital sun sensor is a main goal of the thesis, it is gone through in greater detail than the other types of sensors. A thorough coverage of the theory behind the sun sensor is found in Chapter 3 and the design in Section 4.3.

Chapter 3

Sun sensor background and Theory

This section covers important principles and theory applied in the development of the measurement system in this thesis. Different sun sensor principles are described, followed by a coverage of different light detector technologies. A description of current measurement methods is also provided, as well as a noise analysis of a photodetector circuit.

3.1 Sun sensor principles

There are several ways to do this, including, but not limited to, V-slit sensors, cosine detectors, sun presence detectors, fine sun sensors and digital sensors. This section will go through a few of these.

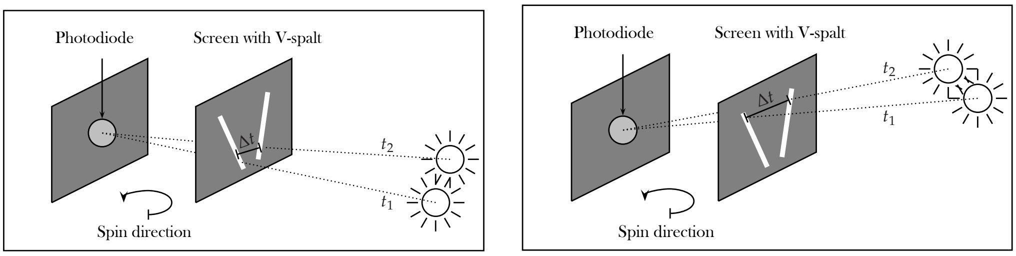
3.1.1 Cosine detectors

The cosine detector is a common type of analog sensor. It is based on a silicon solar cell which gives a sinusoidal variation in output current with the changing sun angle. The output current is proportional to the cosine of the sunlights angle of incidence, as in Equation (3.1). (Wertz, 1978)

$$I = I_0 \cdot cos\theta \tag{3.1}$$

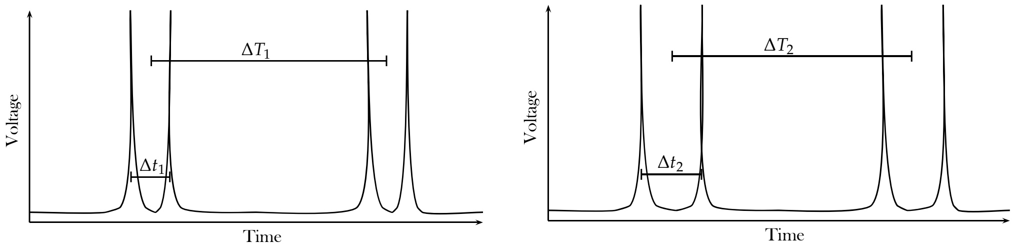
3.1.2 V-slit sensor

A V-slit sensor comprises a photodiode behind a screen. The screen has two slits in the shape of a V, letting the sunlight through. When the spacecraft spins the light will hit the photodiode twice, once through each slit. The angle to the sun is calculated from the time between each hit, combined with the spin velocity. The spin velocity can be found from the time between two pulse pairs. (Sollien, 2006) Alternatively, the V-slit sensor could be combined with a gyro to enable a more accurate calculation of spin velocity and thereby possibly increaing the accuracy of the calculated angle.



(a) (b)

Figure 3.1: Principle schematic of V-slit sensor. The sun is first detected at t_1 and then later at t_2. The time difference is given by $\Delta t = t_2 - t_1$. When the sun is higher, like in (b), the Δt will be larger. By knowing the rotational velocity of the spacecraft the angle can be calculated from Δt and the slit geometry. (Sollien, 2006)



(a) (b)

Figure 3.2: Typical signal from a V-slit sensor. Although Δt_1 and Δt_2 are different when the sun is higher, the rotational velocity remains constant and $\Delta T_1 = \Delta T_2$. (Sollien, 2006)

3.1.3 Pinhole camera

A pinhole camera has been implemented in this thesis, based on the experiences made by Sollien (2006). In a pinhole camera the sunlight passes through a pinhole which will decrease the amount of incident light and create a small light spot on the detector. The image sensor, in this case the PSD, detects the position of the light spot on its active area.

By using trigonometry, the angle of the incident light, the solar angle, can be calculated as in Equation (3.2). When using a 2D detector, this must be calculated twice to get both the X and Y angle.

$$\alpha = tan^{-1}\left(\frac{L}{d}\right) \tag{3.2}$$

A 2D measurement is not neccessary for making angular measurements, as the spacecraft is spinning. The mechanical alignment and added ease of calibration are the main reasons to use a 2D sensor. In practice the sensor will act as a linecamera, and the added accuracy obtainable by using a 2D sensor is negligable. A comparison of the newly developed sun sensor (DSS2) and the old sunsensor (DSS) is found in Section 7.1.

The compensation for offset in a 2D sensor, compared to a sensor of one dimension, can be done completely in software, requiring none of the sensors mechanical parts to be moved. While one would measure the displacement of the 1D sensor manually, the easiest way to do this with a 2D sensor is to make a measurement of a 0° sun angle, with the light source perpendicular to the detector. In post processing, the displacement of the measured center point can easily be compensated for.

The sketch in Figure 3.4 demonstrates the incident parallell light from the sun striking the face of the sun sensor, passing through the pinhole and illuminating a spot on the detector.

The operation of the sun sensor is illustrated in Figure 3.5. Note that this is a model of the operation in time, not a geometrical model.

In the figure, the sensor scans the sky from right to left, because of its counterclockwise rotation. The solar angle is measured at a rate of 5787 Hz, which is the data rate assigned to the sensor in the rocket encoder (see Section 4.2.5). Assuming a maximum spin rate of 6 RPS this results in a minimal radial distance of 0.37° between the measurements. The results are fed directly to the DAQ (see Section 4.2 for information about the DAQ).

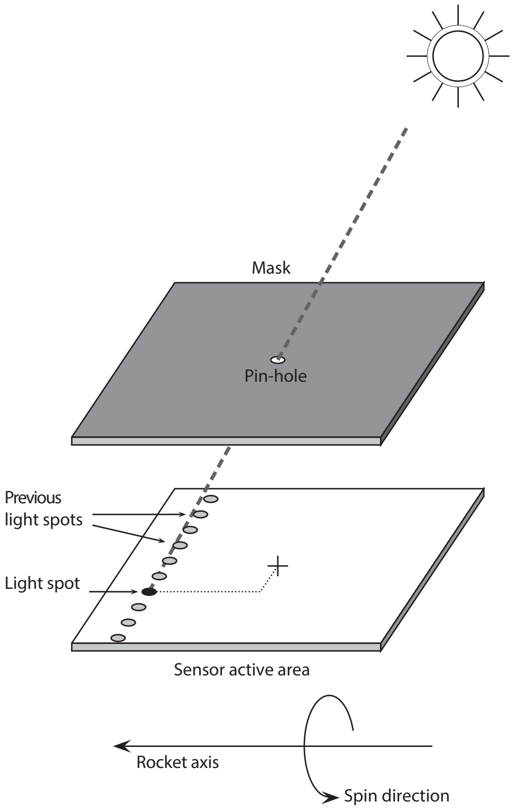


Figure 3.3: Principle schematic of the pinhole camera. Depicted is the movement of the light spot on the sensor surface when the rocket is spinning. (Sollien, 2006)

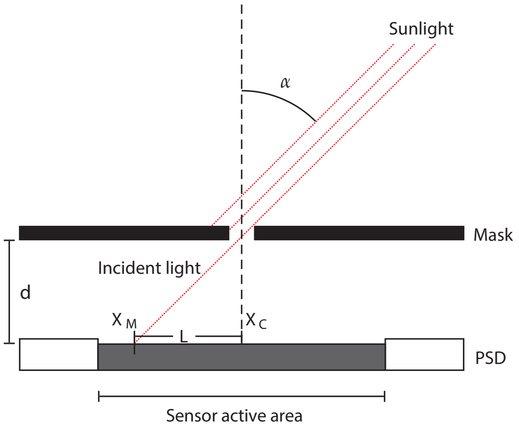


Figure 3.4: Schematic overview of the pinhole camera (Sollien, 2006)

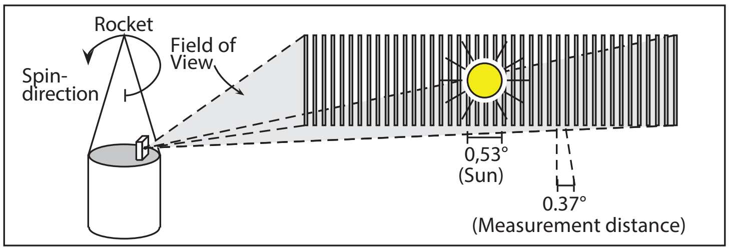


Figure 3.5: The operation of the sensor at 5 rpm spin rate (Sollien, 2006).

The optimal distance d (illustrated in Figure 3.4) between the detector and pinhole is calculated from the field of view, X, and the length of the active area, A, as in Equation (3.3). (Sollien, 2006) Inserting the data for the Hamamatsu S5991 PSD and a field of view of 60° results in:

$$\begin{aligned} d &= \frac{A}{tan(X)} \\ d &= \frac{9\ mm}{tan 60^\circ} = 2.6\ mm \end{aligned} \tag{3.3}$$

A wavelength of 1000 nm can be assumed, as the Hamamatsu S5991 PSD is most sensitive in the 900-1000 nm range. The optimal pinhole diameter can be calculated as in Equation (3.4). (Sollien, 2006)

$$D_{optimal} \approx \sqrt{4\lambda d} \approx 136\ \mu m \tag{3.4}$$

3.2 Light detector technologies

In this section the relevant light detector technologies for the sun sensor are covered.

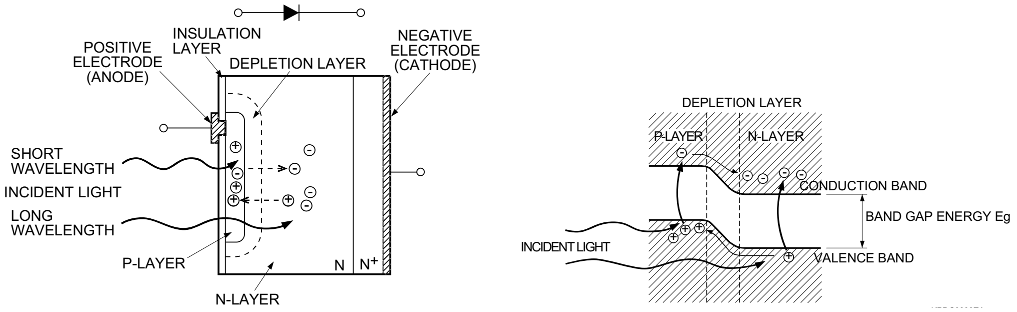
3.2.1 Position Sensing Detector (PSD)

Position Sensing Detector is another name for lateral-effect photodiodes. Before the principle of the lateral-effect photodiode can be understood, it is important to understand the principle and operation of a regular photodiode. It is also relevant for CMOS and CCD, although the photodiodes are usually not operated directly in these technologies.

Photodiodes

A photodiode can detect light because of the photovoltaic effect. The photovoltaic effect is the generation of voltage or current when a material is exposed to light. In the photodiode the current flow is increased when the P-N junction of the diode is illuminated by light.

When a light photon strikes the substrate it can excite an atom to dislogde an electron. This creates a hole-electron pair. The electron will migrate towards the positively charged electrode and the hole will migrate towards the negative electrode, thus creating a small flow of current.



(a) Cross section

(b) Junction state

Figure 3.6: Photodiode properties. (Hamamatsu, 2005)

We can bias the diode to modify it's behaviour by applying a bias voltage. When reverse-biased, applying a positive voltage to the cathode relatively to the voltage on the anode, the diode operates in the photoconductive mode. This means that the diode acts like an insulator, blocking current from passing through. The capacitance of the PN-junction depends on the thickness of the variable depletion region. Increasing the bias voltage increases the thickness of the region and thus lowering capacitance. This is turn enables faster response times as the lower capacitance makes for a more stable sensor at high speeds. The increasing bias voltage also increases the dark current of the sensor, which in turn lowers the dynamic range slightly since the noise floor is heightened. (Hamamatsu, 2005)

When the diode is zero-biased it operates in photovoltaic mode. Having no voltage across the diode means that there is no current flowing through the resistor of the amplifier, and the result is no dark current. Thereby a whole error term is removed.

Lateral-effect photodiode (PSD)

A lateral-effect photodiode is a large-area photodiode where the P-layer is made to have a uniform resistance with an output on both sides, usually two anodes with a common cathode. When a light spot strikes the surface, an electric charge proportional to the intensity is generated at the illuminated position. The charge travels through the resistive P-layer before being picked up by the two anodes as current, as shown in Figure 3.7. As the charge travels through the resistive layer, the output currents, I_{X1} and I_{X2}, will be divided in inverse proportion to the distance traveled. Expressions for position calculation are given in Equation (3.5), where Equation (3.5a) is the displacement from the center electrode and Equation (3.5b) is the displacement from the X_1 electrode.

$$\frac{I_{X2} - I_{X1}}{I_{X2} + I_{X1}} = \frac{2X_A}{L_X} \tag{3.5a}$$

$$\frac{I_{X2}}{I_{X2} + I_{X1}} * Lx = X_B \tag{3.5b}$$

(Hamamatsu, 2002)

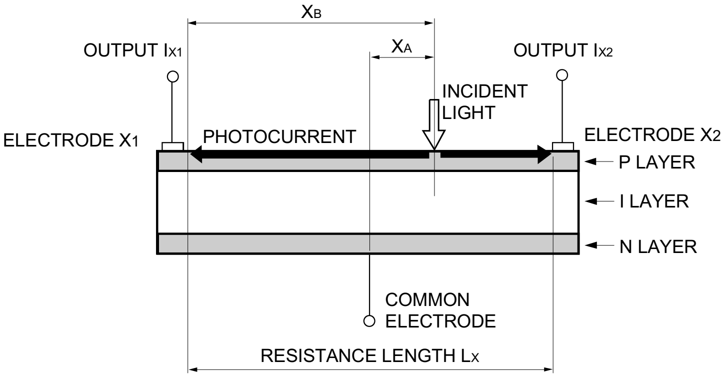


Figure 3.7: Sectional view of a Hamamatsu PSD. (Hamamatsu, 2002)

Duo-lateral PSD

A duo-lateral PSD is in essence two lateral diodes placed on top of each other. On the top layer the diode is structured with two anodes on top and two cathodes on the back surface. In this way the PSD can measure the position in two directions. (Hamamatsu, 2002)

Tetra-lateral PSD

A tetra-lateral diode is manufactured with one single resistive layer. It is structured as four cathodes with a common anode. Compared to a duo-lateral type, the tetra-lateral PSDs tend to have more interaction between the electrodes towards the edges of the sensor. On the other hand it has smaller dark currents and typically a faster response. It is also easier to apply a reverse bias to it. (Hamamatsu, 2002)

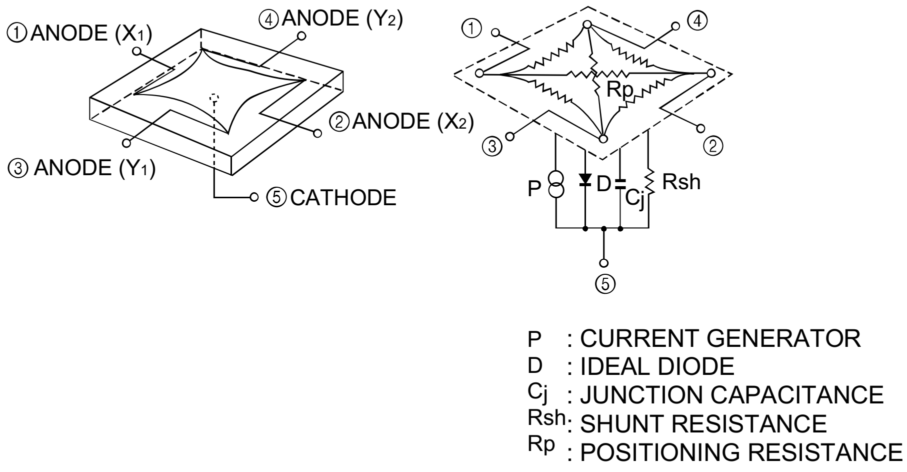


Figure 3.8: The basic structure of a Hamamatsu pin-cushion type PSD (Hamamatsu, 2002)

Pin-cushion type PSD

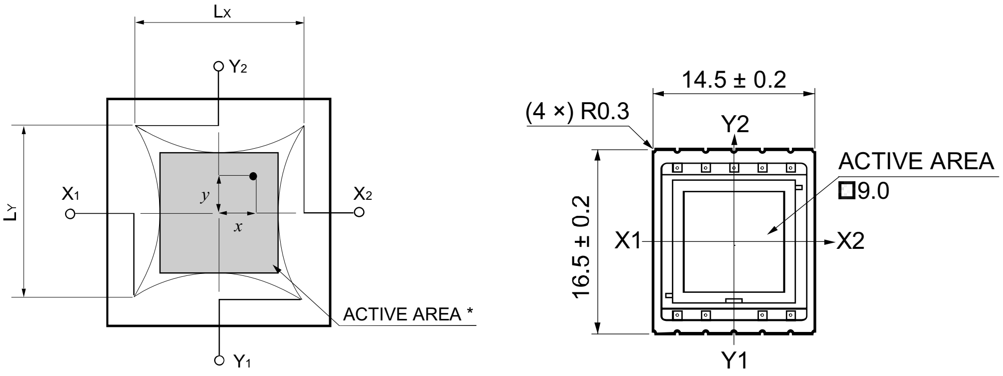
The pin-cushion type PSD is an improved tetra-lateral type. It has an improved active area and the interaction between the electrodes is greatly reduced. It is structured as four cathodes with a common anode, as seen in Figure 3.8, making it very easy to apply a reverse bias. In comparison to the tetra-lateral diode a pin-cushion type also has big improvements in linearity of detection towards the edges. (Hamamatsu, 2002)

Calculating the light spot position on a pin-cushion PSD is a bit more complex than for the simpler PSD types (see Equation (3.6)).

$$\frac{2x}{L} = \frac{(I_2 + I_3) - (I_1 + I_4)}{I_1 + I_2 + I_3 + I_4} \tag{3.6a}$$

$$\frac{2y}{L} = \frac{(I_2 + I_4) - (I_1 + I_3)}{I_1 + I_2 + I_3 + I_4} \tag{3.6b}$$

The PSD is a purely analog detector. This means that the resolution isn't set by a definite number of pixels, but rather by the minimal change in position that will trigger a change in the output. For the sensor used in this thesis the minimal detectable change is 1.5 μm, with a light spot size of 200 μm.



(a) Hamamatsu pin-cushion PSD

(b) Hamamatsu S5991

Figure 3.9: Active area of Hamamatsu PSD. (Hamamatsu (2002), Hamamatsu (2007))

3.2.2 CMOS and CCD image sensors

A CMOS (Complimentary Metal-Oxide Semiconductor) image sensor consists of a matrix of photodiodes. In a CMOS sensor every photodiode has its own amplifier and is usually coupled to a common ADC.

A CCD (Charge-coupled Device) sensor is structured in much of the same way as a CMOS sensor, except that instead of having a dedicated amplifier for each pixel there is one common amplifier for all the pixels. The way this works is by shifting the charges of each pixel through all the other pixels until all have reached the amplifier and ADC. This means that a specific pixel can be read and investigated in a CMOS sensor, while all the pixels would have to shift through the sensor to do the same in a CCD device.

Calculating the displacement of a light spot on this type of device will require a custom search algorithm to be implemented, as the light spot must be found manually in the image taken by the sensor.

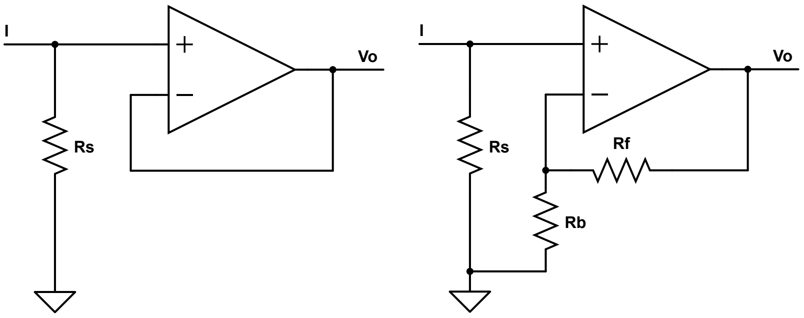
These sensors are sensitive to light with wavelenghts up to about 1000 nm, which is near-infrared light, and the response can be trimmed with a light filter.

3.3 Principles of Current Measurement

Being able to accurately measure small currents is vital for many sensor implementations, like the sun sensor. To measure this techniques are employed to convert the current signal to voltage, which is easier to digitize, as most ADCs measure voltage. This section will go through the basics of the two most common circuits for measuring currents followed by a description of relevant noise sources. An analysis of noise in a photodiode detector (a photodiode coupled to a transimpedance amplifier) is also provided.

3.3.1 Shunt Ammeter

The shunt ammeter, in its simplest form, is essentially a voltage follower with a shunted input. By shunting the input towards ground a voltage potential will form over the resistor, thus creating a voltage potential between ground and the input of the amplifier as depicted in Figure 3.10(a). The output of the circuit is a function of the voltage, V_{in}, across the shunt resistance, R_s, created by the current, I. In the simple form of a shunted voltage follower, the output voltage will be the same as the input voltage.



(a) Simple ammeter (b) Ammeter with gainstage

Figure 3.10: Different ammeters.

The shunt ammeter is seldom used in this most simple form as one usually adds a gain stage, as depicted in Figure 3.10(b), turning the voltage follower into a regular voltage amplifier. The output voltage, V_o, becomes a function of the shunted input voltage and the gain of a non-inverting amplifier, as seen in Equation (3.7b). This also reduces the shunt resistor, R_s, value needed.

$$V_{in} = IR_s \tag{3.7a}$$

$$V = V_{in}(1 + R_A/R_B) \tag{3.7b}$$

It is generally an advantage to use the smallest possible value for R_s. This is because low value resistors have better accuracy in addition to better time and temperature stability, as well as lower thermal noise. Choosing a low resistor value will also reduce the input time constant which leads to a faster response time for the circuit. The disadvantage of choosing a low resistor value is a reduced signal-to-noise ratio as a result of the lower voltage drop created. (Keithley, 2004)

The main disadvantage of a shunt ammeter is the large shunt resistor required to create a sufficient voltage drop across R_s. This leads to a high input resistance which increases the voltage burden of the circuit, meaning that the current measured will be reduced. As a result of adding a gain stage one can reduce the shunt resistor value accordingly, but it is still a major drawback.

3.3.2 Transimpedance Amplifier (Feedback Ammeter)

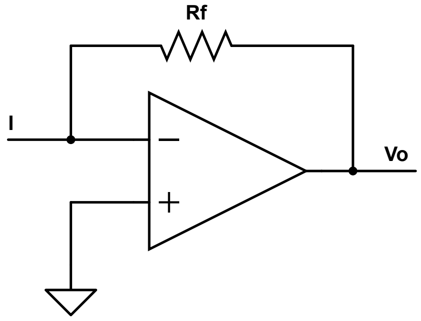


Figure 3.11: The transimpedance amplifier.

The transimpedance amplifier, or feedback ammeter, is another way of making an ammeter. In this configuration the input current I flows through the feedback resistor R_f, giving the the output voltage as V_o.

$$V_o = -IR_f \tag{3.8}$$

The low offset current of the amplifier alters the current, I, by a negligible amount. As the opamp's high gain will force the input voltage to be nearly zero, the voltage burden of the circuit is very low. This also leads to a fast rise time.

It's equally easy to add a gain stage to this type of ammeter, but it's not neccessary, like it is for a shunt ammeter, unless it is used to amplify very low signal levels.

3.3.3 Noise Sources

Noise can be defind as "any unwanted disturbance that obscures or interferes with a desired signal". (Motchenbacher and Connelly, 1993) This section will go through the three most common sources of noise present in an electronic system. These are important effects to consider when designing analog circuits.

Thermal noise (Johnson-Nyquist noise)

If we disconnect the electrical power from a circuit, one might believe that there would be no voltages present in the system. This assumption would be erronous as there would still be thermal noise, although it would be very small.

Thermal noise is caused by the random thermally excited vibration of the charge carriers in a conductor. Each electron carries a charge of $1.602 * 10^{-19}C$, which will induce many small current surges as the electrons move randomly about in the material. As the temperature rises, the random movement of electrons will increase, consequently increasing the thermal noise. (Johnson, 1928)

The noise power of thermal noise is constant (white) for all frequencies, leading to the relation for the noise current I_t and noise voltage E_t:

$$I_t = \sqrt{\frac{4kT\Delta f}{R}} \tag{3.9a}$$

$$E_t = \sqrt{4kTR\Delta f} \tag{3.9b}$$

where k is Boltzmann's constant ($1.38 \cdot 10^{-23}\ Ws/K$), T is the temperature of the conductor in Kelvin (K) and Δf is the noise bandwidth of the measuring system in hertz (Hz). (Nyquist, 1928)

This expression will equate to an infinite noise voltage in an open circuit, but in the real world there will always be some shunt capacitance that limits the voltage. We can still use the given formulas as an approximation in a closed system. (Motchenbacher and Connelly, 1993)

Flicker noise (1/f or semiconductor noise)

Flicker noise has a "pink" power density, meaning that it is inversly proportional with frequency. The noise power typically follows a $1/f^{\alpha}$ characteristic, with α usually being 1, but sometimes ranging from 0.8 to 1.3 in some devices.

The major cause of 1/f noise in semiconductors originates from the surface properties of the material, although improved surface treatment in manufacturing has decreased the magnitude to some degree.

$$I_{1/f} = K_i \sqrt{ln \frac{f_{max}}{f_{min}}} \tag{3.10}$$

where K_i represents I_n at 1 Hz. f_max and f_min are the maximum and minimum frequencies in Hz. (Mancini, 2002)

Since 1/f noise increases with decreasing frequencies, it should theoretically be infinite at DC. Although this might be a theoretical possibility it is generally manageable as the low frequency cutoff in an amplifier is set by the time it has been turned on, effectively attenuating the 1/f noise at extremely low frequencies. (Motchenbacher and Connelly, 1993)

Shot noise

Shot noise is a phenomena appearing in the pn junctions of semiconductor devices. Its power density is the same as that of white noise.

Each hole and electron has a charge. When they pass from the anode to cathode (or vice versa) it results in a small current impulse, a change in the current flow. Since the noise currents are created by the arrival of holes and electrons it is directly related of the DC current flow, as seen in the formula:

$$I_{sh} = \sqrt{2qI_{DC}\Delta f} \tag{3.11a}$$

$$E_{sh} = \sqrt{2qI_{DC}\Delta f} * R_f \tag{3.11b}$$

where q is the electronic charge ($1.6 \cdot 10^-19$ coulomb), I_{DC} is the forward DC current in amperes and Δf is the noise bandwidth in hertz (Hz). (Schottky (1918), Motchenbacher and Connelly (1993))

Generation-recombination noise (G-R)

Generation-recombination noise is often a dominant noise mechanism in photodetectors. It often appears to be similar to thermal noise, but is actually caused by fluctuations in the conductivity of the material. This is the result of fluctuations in the generation, recombination and trapping of carriers in the semiconductor. The spectral response of G-R noise is white up to a frequency determined by the lifetime of the carriers in the photodetector. (Motchenbacher and Connelly, 1993)

Noise analysis of the photodiode detector

Figure 3.12 depicts the photodiode detector circuit used in the thesis. A full noise equivalent circuit is shown in Figure 3.13 followed by a calculation of total system noise.

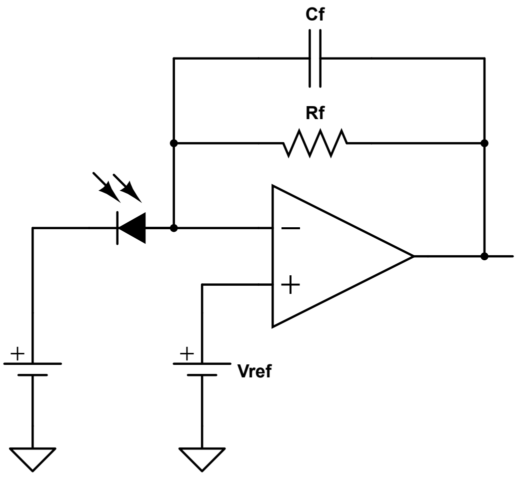


Figure 3.12: Schematic of the photodetector used in the thesis. The PSD is approximated as four photodiodes, with one of the channels shown in this schematic.

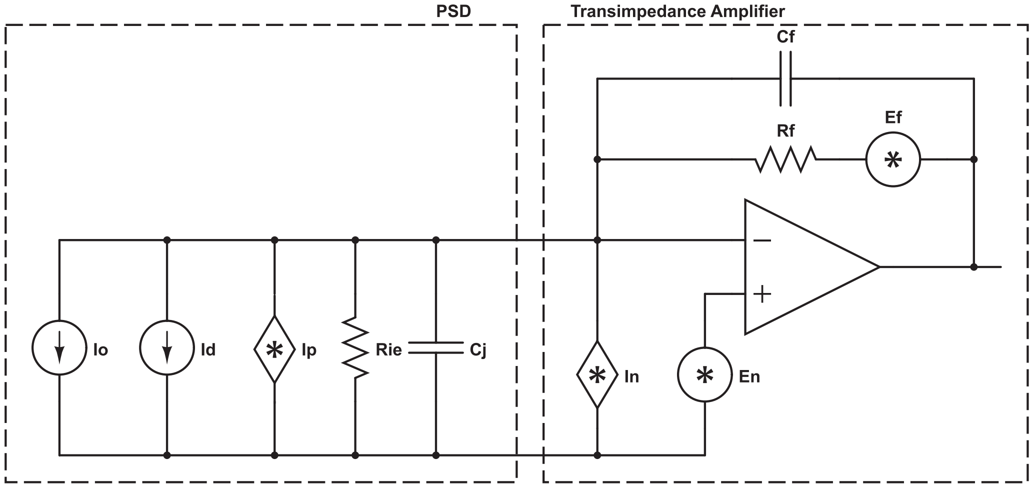


Figure 3.13: Noise equivalent circuit of a photodiode detector. The photodetector model is from Hamamatsu (2002) with added visual noise signals.

r_{ie} = Interelectrode resistance

R_F = feedback resistance

I_O = sensor dc photocurrent

I_D = sensor dc dark current

E_F = thermal noise of $R_F = \sqrt{4kTR_F\Delta f}$

I_P = shot, G-R, 1/f noise and thermal noise of $R_{ie} = (I_{sh}^2 + I_{G-R}^2 + I_{1/f}^2 + I_{Rie}^2)^{1/2}$

E_n = amplifier input voltage noise

I_n = amplifier input current noise

Shot noise is a function of photocurrent, dark current and input bias current. This means that this noise will increase with an increase in reverse bias or light intensity:

$$I_{sh} = \sqrt{2q(I_O + I_D + I_n)\Delta f} \tag{3.12}$$

The total PSD current noise can be expressed as:

$$I_{in} = \sqrt{\left(\frac{E_n}{R_{ie}}\sqrt{\Delta f}\right)^2 + I_{sh}^2 + I_{1/f}^2 + I_{G-R}^2 + I_{Rie}^2} \tag{3.13}$$

Building on this, the output voltage noise is expressed as:

$$E_{out} = \sqrt{E_{sh}^2 + E_{Rie}^2 + E_{1/f}^2 + E_{G-R}^2 + E_{no}^2 + E_F^2 + (I_n R_F \sqrt{\Delta f})^2} \tag{3.14}$$

(Hamamatsu, 2002)

Noise calculation example

Assuming a maximal photodiode current of 25 μA and dark current of 1 nA, room temperature (20° C or 293.1 K), interelectrode resistance (R_{ie}) of 10 kΩ and a noise bandwidth of 15 kHz.

Typical operating conditions for the OPA129 opamp is used, with an input bias current of ± 30 fA, input current noise density of 0.1 fA/$\sqrt{Hz}$ and a ± 0.5 mV input offset voltage. To simplify the calculations, the voltage noise density at 10 kHz, 15 nV/$\sqrt{Hz}$ is used for the calculations of its effect on the total noise. As the noise density in this frequency region will dominate the voltage noise, this is a valid approximation. The feedback resistor is 158 kΩ.

As the bias current is very low compared to the photodiode and dark current, it can be neglected for the total input current.

The noise signals amount to:

$$\begin{aligned}
E_{sh} &= \sqrt{2q(25\ \mu A + 1\ nA)\cdot 15\ kHz}\cdot 158\ k\Omega & \approx 44.69\ \mu V\\
E_{1/f} &= 280\ nV\sqrt{ln\left(\frac{15\ kHz}{1\ Hz}\right)} & = 0.85\ \mu V\\
E_{Rie} &= \sqrt{4k\cdot 293.1\ K\cdot 10\ k\Omega\cdot 15\ kHz} & = 1.27\ \mu V\\
E_{Rf} &= \sqrt{4k\cdot 293.1\ K\cdot 158\ k\Omega\cdot 15\ kHz} & = 5.06\ \mu V\\
E_{no} &= \left(1+\frac{R_f}{R_{ie}}\right)\cdot E_n\cdot\sqrt{15\ kHz} & = 30.86\ \mu V\\
I_nR_F\sqrt{\Delta f} &= 0.1\ fA/\sqrt{Hz}\cdot 158\ k\Omega\cdot\sqrt{15\ kHz} & = 1.58\ nV \qquad (3.15)
\end{aligned}$$

As a good formula for G-R noise wasn't found, this noise effect is negligated. The total output noise amounts to:

$$\begin{aligned}
E_{out} &= \sqrt{E_{sh}^2 + E_{1/f}^2 + E_{Rie}^2 + E_F^2 + E_{no}^2 + (I_nR_F\sqrt{\Delta f})^2}\\
&= \sqrt{(44.69\ \mu V)^2 + (0.85\ \mu V)^2 + (1.27\ \mu V)^2 + (5.06\ nV)^2 + (30.86\ \mu V)^2 + (1.58\ nV)^2}\\
&= 54.57\ \mu V \qquad (3.16)
\end{aligned}$$

It is clear from the calculations that the shot noise and the input voltage noise are the dominating contributions to the output noise. If a higher photodiode current of 50 μA was present and a lower feedback resistor of 70 kΩ employed, the distribution of the noise contributions would be different:

$$\begin{aligned}
E_{sh} &= \sqrt{2q(50\ \mu A + 1\ nA)\cdot 15\ kHz}\cdot 70\ k\Omega & \approx 28.00\ \mu V\\
E_{Rf} &= \sqrt{4k\cdot 293.1\ K\cdot 70\ k\Omega\cdot 15\ kHz} & = 3.37\ \mu V\\
E_{no} &= \left(1+\frac{70\ k\Omega}{10\ k\Omega}\right)\cdot E_n\cdot\sqrt{15\ kHz} & = 14.70\ \mu V\\
I_nR_F\sqrt{\Delta f} &= 0.1\ fA/\sqrt{Hz}\cdot 70\ k\Omega\cdot\sqrt{15\ kHz} & = 0.70\ nV\\
E_{out} &= 31.84\ \mu V & (3.17)
\end{aligned}$$

While shot noise is the dominating noise source, it is seen that the input voltage noise has an increasingly large contribution when a larger gain is applied. It is evident from the calculations that choosing an operational amplifier with a low input noise voltage is very important to the noise floor, as it is directly amplified on top of the signal. Even though the input bias and noise current is important to consider as well, their effect on the total output noise is negligable as they are much lower than the other current signals present. It can be seen that the effect of the input voltage noise is proportional to the feedback resistance by a factor of $1 + R_f/R_{ie}$.

The capacitance of the photodiode, C_d, the stray capacitance of the wire and the feedback capacitance, C_f, will reduce the max frequency of both the noise and the signal from the photodiode.

Matlab code for the noise calculations is included in Appendix C.1

Chapter 4

System Design

This chapter covers the design and development of the Digital Acquisiton Card (DAQ) and the Digital Sun Sensor (DSS2). Selection of the gyroscope and design of the external magnetometer instrument is also covered. An overview of special considerations made when designing the PCBs for the instruments is given as well.

4.1 System Overview

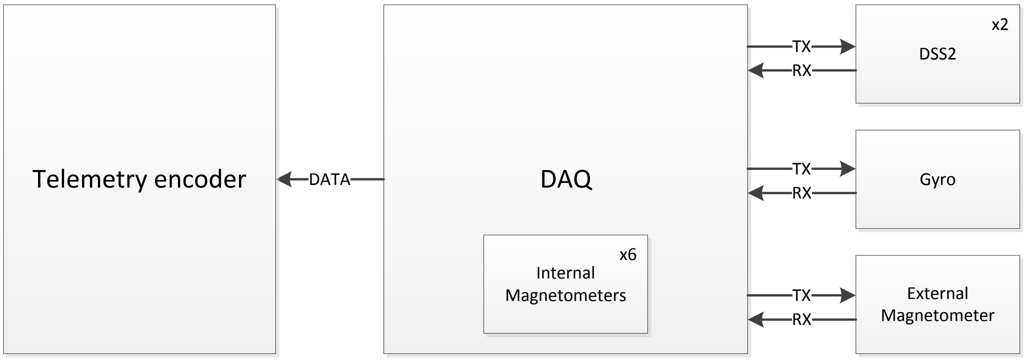


Figure 4.1: Top level block schematic of the measurement system.

As illustrated in Figure 4.1, the systems main unit is the DAQ, which communicates with the external sensors and carries the internal magnetometer. The external sensors include two sun sensors, a rate gyro and an external magnetometer. Communication with the rocket encoder, which transmits the data to the ground, is also handled by the DAQ.

Two sensors are used for redundancy, but also to give more frequent measurements to the attitude estimator. The electrical design faciliates the use of two sun sensors, but the ICI-4 will be launched with one unit of the sun sensor designed by Sollien (2006). Consequently, only one sun sensor has been implemented in the firmware at this point.

4.2 DAQ-board

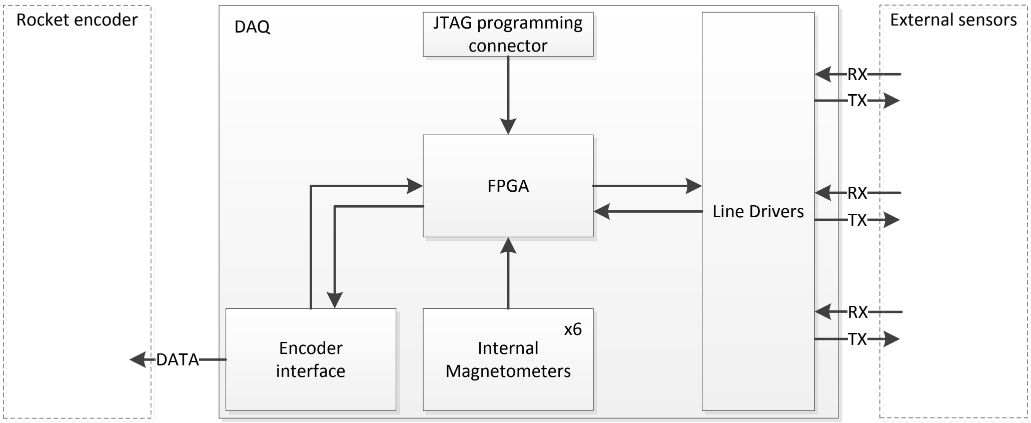


Figure 4.2: Top level block schematic of the DAQ.

A simple block schematic of the DAQ-board is presented in Figure 4.2. As mentioned in Section 4.1, the DAQ is the top unit of the measurement system.

The purpose of the DAQ unit is to collect data from the sensors, both the internal magnetometer and the external sensors, and relay the data to the rocket interface. To do this, an interface for communicating with the external units had to be developed. Realization of the sensor interface is covered in Section 4.2.4. Communication towards the encoder is outlined in Section 4.2.5.

The DAQ shall also include an internal magnetometer, capable of measuring at a sample rate of approximately 1 kHz. A brief coverage of the internal magnetometer realization is included in Section 4.2.3.

A processing device is needed to collect the data from all the sensors, internal and external, execute necessary calculations and relay the data to the encoder interface. Implementation of the digital processing and technology choice for the processing device is covered in Section 4.2.1. The algorithm for the sun sensors is implemented on the DAQs processing unit, but the other instruments transmit ready-to-send digital values.

Power supply for both internal components and external sensors are also provided by the DAQ, as described in Section 4.2.2.

4.2.1 Digital design

Programmable Logic

There are two general types of programmable circuits available, microprocessor based circuits and programmable logic, like FPGAs and CPLDs. For this thesis it was chosen to use programmable logic, as the timing of these devices can be determined to within one clock cycle. The timing of a microprocessor based system can not easily be determined to the same degree.

Unlike a microprocessor, an FPGA does not execute a compiled program, a list of instructions, sequentially. An FPGA consists of configurable blocks of logic, called logic elements, with configurable interconnects between them. Using a hardware description language (HDL) these logic elements can be programmed and interconnected to perform almost any logic function. It is even possible to implement a microprocessor on the FPGA, called a soft core. The HDLs are concurrent programming languages and the logic described will run concurrently, in parallel. This makes it possible to add functionality and other modules to the digital design without altering the existing functionality. In this thesis, the HDL chosen to describe the logic is VHDL (Very high speed integrated circuit Hardware Description Language).

Differences between FPGAs and CPLDs

Although FPGAs and CPLDs are very similar in usage, there are a few differences. FPGAs are classified as fine-grain devices, while CPLDs are coarse-grain. The most obvious difference is that FPGAs can contain many thousands of logic blocks, while CPLDs are usually restricted to a few thousand. The logic blocks of the FPGAs are smaller and capable of more complex operations, but the CPLDs have the advantage of a faster input-to-output time because of the simpler coarse grain architecture. This makes latency in a CPLD even easier to predict than in FPGAs and may also result in a higher maximal clock frequency.

In addition to having more logic elements, most FPGAs have embedded memory, multipliers and PLLs. Larger models often include more advanced features, e.g. embedded DSP blocks, microprocessor cores and trancievers. This enables very complex signal processing to be implemented on the FPGA, e.g. FFT and FIR filters. Soft processor cores can also be implemented, making the FPGA capable of running algorithms coded in C.

Most FPGAs, although exceptions are plentiful, are SRAM-based and must be configured at start-up by an external circuit, while are CPLDs usually flash or E^2PROM based and thus instantly on.

Logic design

The DAQ receives data from the gyro, the internal and external magnetometers and the sun sensors. Coordinates (X and Y) for the displacement on the sun sensors active area are calculated before the data is sent to the encoder. It is chosen to calculate the actual angle in post processing, as it can be done with higher precision and to reduce the complexity of the digital design. As the systems first flight will be conducted with only one unit of the new sun sensor, implementing a feature to discriminate the sun sensor measuring the highest intensity was less prioritized and is therefore delegated to future work. A diagram of the logic design and clock distribution network is shown in Figure 4.3.

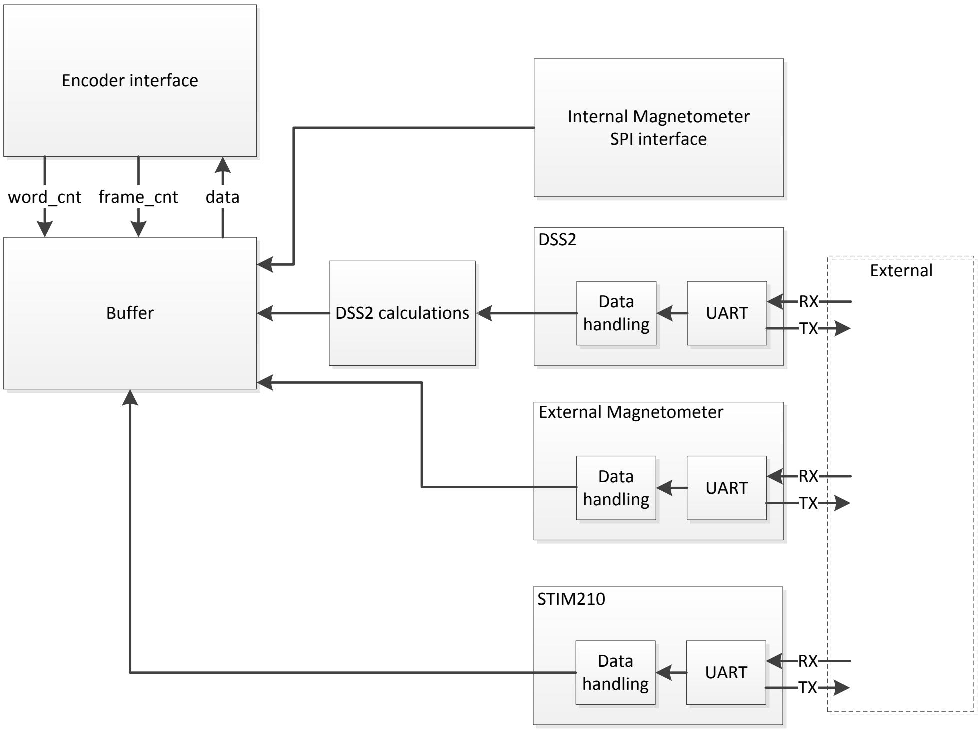
Data from the magnetometers and gyro are simply passed through to the encoder. The only processing done is shifting the data into the encoder interface at the right times. The rest is handled by the encoder interface by Bekkeng (2009), covered in Section 4.2.5.

The VHDL code and ASM-diagrams for the DAQ are included in Appendix B.2.1.

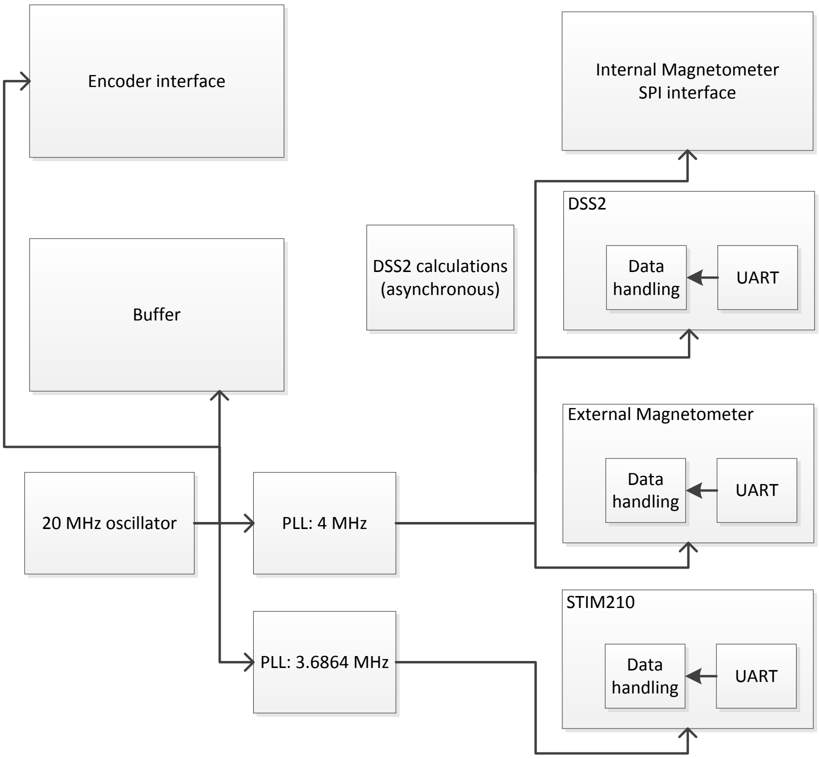
An FPGA was chosen for the design, as the CPLDs didn't have enough logic elements to implement the needed calculations. The FPGA chosen for the DAQ is an Altera Cyclone IV EP4CE15 with approximately 15.000 logic elements in an FBGA-256 package. It may be advisable to keep it at this size, even though it has much more logic resources than needed, as it is also used by the other instruments in the UiO-stack; the instrument stack consisting of all the instrument DAQs from UiO. The UiO-stack is covered in Section 5.2. Using the same unit could facilitate easier upgrades, if neccessary at a later time, and maintenance of the design.

It is also planned to use a common power supply for all the instruments in the UiO-stack in a later version.(Bekkeng, 2013) This will lead to a significant reduction of power supply components, weight and possibly lower power supply noise. Predicting the behaviour of the systems combined may also be simpler when they consist of several identical units, rather than if different models were to be used.

The interface for reading data from all the external sensors is based on a UART module developed as part of the thesis (see Appendix B.1) used in a general interface module (see Section 4.2.4). A simple protocol has been implemented, where a transmission must contain the following: ID byte, a specified number of data bytes, carriage return (x0D) and line feed (x0A). By doing it the same way for all the sensors, the possible error sources are reduced and the maintainability of the code is significantly increased. This UART-module is also utilized on the external magnetometer and the sun



(a) Logic schematic.



(b) Clock distribution network.

Figure 4.3: Logic schematic and clock distribution network of the DAQ.

sensor to reduce and DRY (Don't Repeat Yourself) the code base.

4.2.2 Power Supply

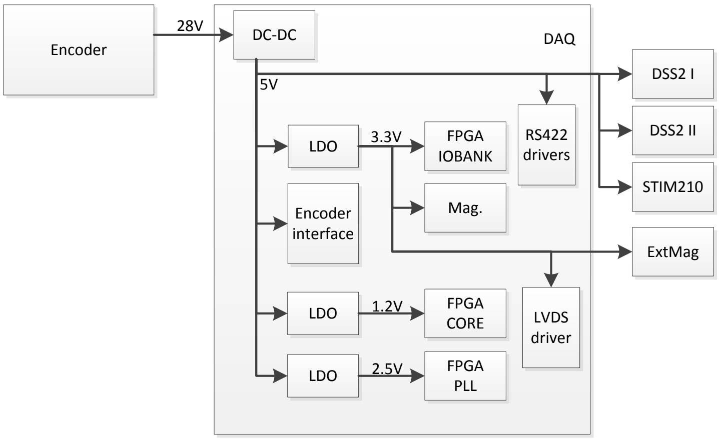


Figure 4.4: Power distribution network for the DAQ-board.

A Traco TEN5-2411 DC-DC converter is utilized to convert the unregulated 20-32 V supply from the rocket to the 5 V level used throughout the system It is capable of supplying a maximal output current of 1 A and accepts an 18-36 V input voltage. Output ripple is low, at maximally 50 mV_{pk-pk}. (Traco, 2012) As the unregulated supplys voltage may drift, a wide input voltage range was necessary.

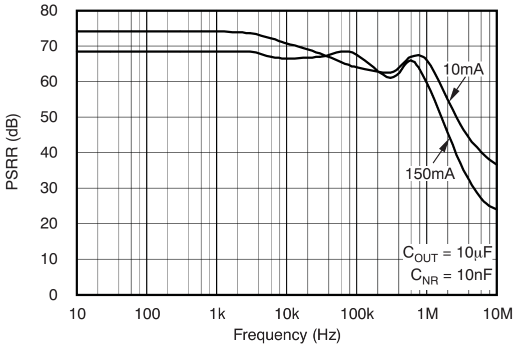


Figure 4.5: Typical PSSR characteristics of the TPS717 series LDO regulators in relation to frequency (assuming V_{in}-V_{out} = 1 V). (Texas Instruments, 2009)

Three TI TPS717 low dropout regulators were initially used to source the 1.8 V, 2.5 V and 3.3 V voltages. They are capable of supplying 150 mA output current with very low noise, typically 30 μV, and have a very high Power Supply Rejection Ratio (PSRR). When used with a sufficiently large difference in input and output voltage, the PSRR will typically be above 60 dB at frequencies up to 1MHz, as seen in Figure 4.5 (Texas Instruments, 2009). This effectively eliminates the output ripple of the DC-DC converter. Measurements of this are presented in Section 6.3.2.

As the load on the 3.3 V net proved to be too high for the TPS717, it was exchanged for the TI LM1117 in a second revision as this regulator could provide a higher output current.

Both the sun sensors and the gyro are powered by the 5 V supply and the external magnetometer is powered by the 3.3 V.

4.2.3 Internal Magnetometer

The Honeywell HMC5983 integrated three-axis magnetometer was chosen for this thesis. It requires only three capacitors to operate correctly and is able to communicate on both I^{2}C and SPI. To simplify the logic design, it was decided to use it in SPI mode, as this is a much simpler protocol to implement in VHDL.

The requirement of a $\sim$1000 Hz sampling rate for the magnetometer is accomplished by using six separate magnetometers. They are triggered sequentially by a 960 Hz 0 to 5 counter. Triggering the measurement mode of the magnetometers sequentially will make each magnetometer sample with a frequency of 160 Hz, skewed in the time domain by a clock cycle of the 960 Hz. When combined, this forms a 960 Hz magnetometer.

Because of time constraints the logic design for the magnetometers was not implemented as part of this thesis. However, as the same magnetometers are used on the external magnetometer card with the firmware implemented on a CPLD, it should be managable to port the code.

4.2.4 Interface to external units

To interface external units, a general interface module was created. It consists of the aforementioned UART-module and a finite state machine to find and read the right data. A simple protocol of sending a chosen ID-byte first, followed by the databytes, a Carriage Return (x0D) byte and a line feed (x0A) byte is implemented to make recognizing the right string of data a simple task. The VHDL code and ASM-diagram are found in Appendix B.2.2.

Line Drivers / Interfacing

To be able to send data over distance, without having to worry about noise, line drivers are employed. It is also a good way to make a system easier to interface and debug from a PC, as line drivers conform to relatively strict standards for interoperability.

For the sun sensors and the gyro, the TI SN65LBC179A RS-422 drivers are used. This choice was made because this specific model had a lot of flight heritage at the department, which meant it should be a safe choice. It is also to be used for the encoder interface and it made sense to use the same standard all around.

The line driver chosen for the external magnetometer is an LVDS type, TI SN65LVDS179, of which there have been few experiences with at the department before. In an effort to reduce power usage and noise in a later version, it was chosen to use the external magnetometer as a pilot project to test the viability of this communication standard.

RS-422

RS-422 is a technical standard for the characteristics of a differential signaling circuit. Protocols and pin assignments are defined in other standards. The maximum differential voltage swing is +6 to -6 V.

The standard covers transmitting data with as high rates as 10 Mbit/s at a distance of 12 meters, although an upper limit isn't actually specified. Typical data rates for RS-422 is 40 Mbps at 1-3 meters and 10 Mbps at 10 meters.

The maximal range of RS-422 is 1500 meters, well above the requirements being posed by this system.

A drawback of RS-422 is that it uses a lot of power, and this varies with frequency.

LVDS (Low Voltage Differential Signaling)

While they are quite similar in design, LVDS provides higher transmission speeds and lower power consumption than RS-422. Unlike the former, LVDS uses a constant-current driver which makes the power consumption much more independent of the transmission frequency. It's quiescent current is also significantly lower. The drivers have a typical voltage swing of 350 mV with a common mode voltage of 1.25 V.

By using resistor divider network, an LVDS receiver is actually capable of receiving data from an RS-422 driver.

4.2.5 Telemetry encoder interface

The instruments have been assigned a fixed space in the telemetry encoders PCM-matrix. The digital sunsensor (DSS2) has been assigned as Supercom-2, meaning it shall send data two times per telemetry frame. This means that it sends data with a frequency of 5787 Hz.

The gyro, external magnetometer and internal magnetometer has been assigned as SubCom-4, which means the data from these sensors are sent over the course of 4 frames, once per frame for each sensor. Data from these sensors are sent on the order of X, Y, Z and a last byte for optional housekeeping data.

The design of the telemetry encoder interface is a "tried and true" circuit and logic design by Bekkeng (2009) and his thesis covers the important aspects of this in great detail. The VHDL code for the interface is included in the top file in Appendix B.2.1 and a schematic of the encoder interface is included in Figure 4.6.

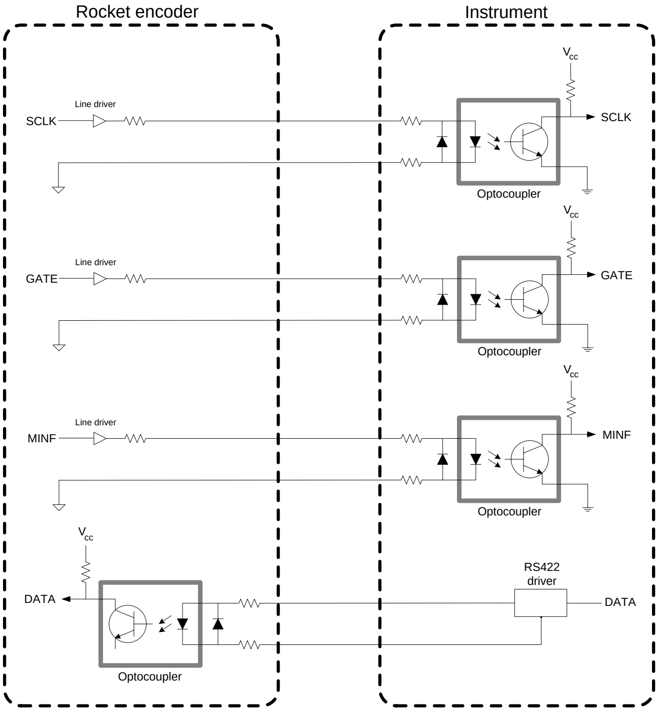


Figure 4.6: The telemetry encoder interface. The optocouplers provide a galvanic isolation from the encoder. Bekkeng (2009)

As this design is verified and has flight heritage from several sounding rocket launches there was no reason to do a new design. The code is modified slightly so that a 16-bit shiftregister could be used instead of concatenating the data words, which results in a cleaner code. While the main advantage is a more readable code, it also saves a few logic elements and increases the maximum frequency the circuit is able to operate at, giving it more headroom.

4.3 Digital Sun Sensor (DSS2)

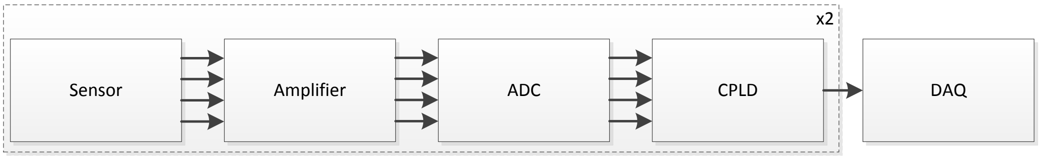


Figure 4.7: A block schematic of the DSS2 instrument.

Depicted in Figure 4.7 is a simple block schematic of the digital sun sensor instrument. The current output of the PSDs four cathodes is amplified through four transimpedance amplifiers to produce four voltage signals which are digitized by four ADCs. The digital data streams are received by the CPLD, which relays the serialized data to the DAQ-board (see Section 4.2).

This section will go through the requirements and considerations necessary when choosing a suitable detector for the sun sensor instrument and the reasoning behind the choices that have been made. A coverage of different sensor technologies is also included.

Initially, it was planned to calculate the X and Y coordinates for the light spot on the sun sensor instrument, but this proved to be too logic intensive and could not be fitted on the CPLD. Processing of the data is therefore done on the DAQ-board because of its inherent larger logic capacity.

4.3.1 Sensor Requirements

A two-dimensional sensor is beneficial since it permits easy calibration of offset, which can be caused by different factors, e.g. mechanical misalignment. With a two-dimensional sensor the process of calibration and compensating for a displacement in the measured center point can be done exclusively in post processing, or in firmware if so desired, with no need to adjust the mechanics. Seeing as the attitude is only determined and cannot be controlled, it is no actual need to compensate for offset and calibration in real-time.

A more accurate compensation can be applied in post processing (see Section 6.1.3).

Since the rocket spins at a maximal rate of 6 rps, the update rate has to be at least 2160 Hz to achieve a 1° resolution of available measurements. The sun sensor shall measure the X and Y coordinates for the sunlight at a rate of 5787 Hz (see Section 4.2.5), which fulfills this demand.

The target resolution of the sun sensor is $< 0.1°$ with an accuracy better than 1°. To attain the required accuracy the sensor needs an adequate resolution to take full advantage of the sample rate. At nominal spin of 4 RPS, the angular distance between measurements is 0.18° with a sample rate of 5787 Hz. The resolution required to detect the angular distance is found by dividing the field of view by the required angular resolution. Given a field of view of ± 60°, the result is $120°/0.18° = 666$ points, which should be achievable with most sensors.

The ability to measure the intensity of the light spot is desirable as it enables the use of tresholding. By detecting intensities over a set level, actual sun presence can be discriminated from less intense light sources.

4.3.2 Selection of light detector

Because of its ease of implementation, low cost and availabilty a pin-cushion type PSD (as described in Section 3.2) was chosen. An advantage compared to a CMOS or CCD device (as described in Section 3.2.2) is the reduced demands for complex algorithms as the X and Y coordinates for the light spot position can be calculated directly from the four current outputs. The sensor chosen for this instrument is the Hamamatsu S5991, shown in Figure 4.8. It is relatively cheap, has excellent linearity and the lead time was reasonable.

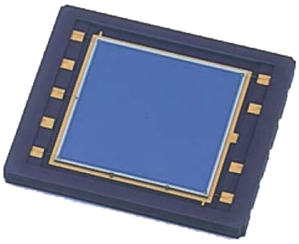


Figure 4.8: The Hamamatsu S5991 pin-cushion type PSD.

The specified position resolution for the S5991 is 1.5 μm (with a 0.2 mm spot size), which implies a theoretical resolution of 6000x6000 discriminable points. The resolution will most likely be significantly lower in practice, but it should be more than adequate. The optimal pin-hole size calculated in

Section 3.1.3 of 136 μm is exchanged for a 200 μm unit, to assure a large enough spot-size.

An important disadvantage of using a pin-cushion PSD is the low sensitivity of the device, which can be about 16 times lower than a duo-lateral device. (Wang and Busch-Vishniac, 1989) This was not considered to be an issue, as implementing a high-gain amplifier should not pose any problem.

The Hamamatsu S5991 has a typical rise time of 2 μs, when measuring 900 nm light, corresponding to a theoretical maximal readout rate of 500 kHz. Rise time is specified with a 5 V reverse voltage, but the reverse bias used in the system amounts to 1.4 V. A 1.4 V reverse voltage corresponds to a dark current of 500 pA, as can be seen in Figure 4.10(a), which is reasonably low.

The anode is connected to 5 V, as this enabled a simple routing of power for the circuits on the board. Biasing the PSD in photoconductive mode leads to a larger dark current and thereby a larger shot noise, as seen in Section 3.3.3. However, it offers the advantage of a decreased terminal capacitance (see Figure 4.10(b)). Less capacitance enables a faster response time and will contribute to the overall stability of the circuit. Even though the measurements are done at a relatively low frequency it is advantageous to increase the reponse time.

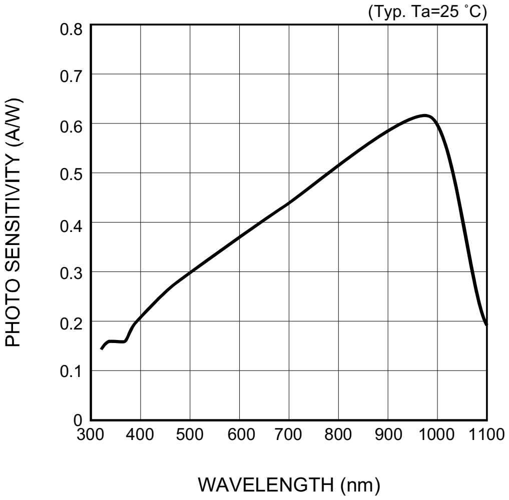
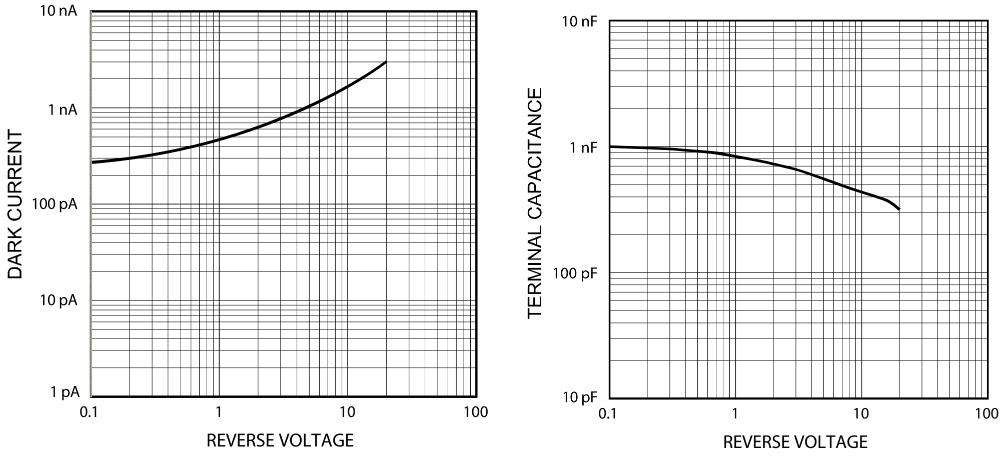


Figure 4.9: Spectral response of the Hamamatsu S5991 (Hamamatsu, 2007)

A filter can be added to the structure, as long as one takes it into account when the X and Y coordinates are used to find the sun angle. As the PSDs sensitivity is linearly decreasing below 900 nm it was deemed unnecessary, as it should be a sufficient filter in itself. Visible light, e.g. most of the reflections from the Earth, is thereby attenuated.

It is worth noting that the first flight will be done with one DSS2 and one DSS1. This will be a good test to see if the sensor works satisfactory without a filter while still using the old system as a backup.



(a) (b)

Figure 4.10: Reverse voltage vs (a) dark current, (b) terminal capacitance for the Hamamatsu S5991. (Hamamatsu, 2007)

As the sensor only receives a very small spot of light, a large dark current may be very deteriorative to the measurements, significantly the reducing dynamic range. This would have the most prominent effect on measurements with a large angle as the photon energy received by the PSD will be lower than when the light strikes it perpendicularly. Fortunately, the dark current is so low that it won't affect the measurements, as demonstrated in Section 3.3.3.

A two-dimensional CCD with a fast enough response seems to be non-existent. The CMOS sensors that were fast enough were on the other hand very expensive or difficult to implement (or both). Most of the products found were very speciallized for a specific purpose.

4.3.3 Transimpedance Amplifier

As described in Section 3.3, the transimpedance amplifier places a negligable load on the preceding circuit, which makes it a good choice for this application. When used to amplify the current signals from the PSD, it has to be stable when using a high gain and it should disturb the input signal as little as possible.

Choice of Operational Amplifier (Opamp)

When selecting an operational amplifier to use as a transimpedance amplifier for such low levels it was important to find a model with very low noise and bias. The OPA129 from Texas Instruments was chosen as this model provided an especially low input current bias and noise. Using a model with higher input current bias could potentially alter the measurements from the PSD in an unpredictable way, as the electrodes of the detector are dependent of each other.

As demonstrated in Section 3.3.3, the voltage noise will be a dominating contribution to the total output noise. While both voltage and current noise is important to consider, the current noise contributes less to the total output noise of the system when the bandwidth of the system is low. The noise calculations done in Section 3.3.3 are based on the actual values of the design and are therefore real world values. As seen, the effect of the input voltage noise is proportional to the feedback resistance by $1 + R_f/R_{ie}$. The opamp also has a substrate connection for making guard rings on the inputs, but this feature was not implemented as it was deemed an unnecessary increase in PCB layout complexity.

An important point is to amplify the signal as much as possible within the range of the ADC to maximize the dynamic range. A maximum output level of 4.096 V was chosen, since this was the highest output voltage reference that could be found with a 5 V supply (see Section 4.3.5). However, it was later discovered that a 5.5 V voltage is required for the amplifier to produce this level at the outputs, or exchanging the reference for a 3.3 V unit. This is discussed further in Section 6.1.2.

The voltage reference is connected to the positive input of the amplifier, which gives the transfer function in Equation (4.1). A disadvantage of doing it this way is that the previous 5.5 V reverse bias of the PSD will be reduced to 1.404 v, as the voltage on the input will carry directly over to negative input. This was deemed a small problem, since a 1.404 V reverse bias is still retained. If a higher reverse bias is desired, the circuit could be replaced by a transimpedance amplifier followed by a differential amplifier with a reference voltage.

$$V_{out} = 4.096\ V - I_{in} \cdot R_f \tag{4.1}$$

In the first revision of the design, a different ADC was used (see Section 4.3.4) and its maximum input signal was ± 2.5 V. This version was also supplied by a ± 12 V voltage, which increased the complexity of the power net routing. See Appendix A.3 for a detailed coverage of the different design revisions.

Determining gain

The pin-cushion type PSD's have low sensitivity and a high gain is necessary to amplify the signals to an appropriate level.

As the inter-electrode resistance of the PSD may vary, in the range of 5-15 kΩ (typically 7 kΩ), between units, it can be hard to determine a gain that will work with different units. The estimation of the correct gain thus becomes a part of the calibration routine for the unit.

The output current of the PSD can be calculated as in Equation (4.2a). Sunlight in space has an intensity of 1370 W/m^2, the PSD has a sensitivity of 0.6 A/W and it is covered by a 200 μm pinhole. The area of the pinhole amounts to $\pi \cdot r^2 = 31.4\ nm$. When the output current is estimated, the required gain is calculated in Equation (4.2b).

$$I_{PSD} = 0.6\ A/W \cdot 1370\ W/m^2 \cdot 31.4\ nm^2 = 25.81\ \mu A \tag{4.2a}$$

$$A = \frac{4.096\ V}{25.81\ \mu A} = 158.7\ kdB\Omega \tag{4.2b}$$

Typically, a gain value of 158.7 kdBΩ is required in sunlight. The nearest feedback resistor value, 150 kΩ, below this should be used.

If a 300 μm pin-hole were used, the estimation changes to:

$$I_{PSD} = 0.6\ A/W \cdot 1370\ W/m^2 \cdot 70.7\ nm^2 = 58.10\ \mu A \tag{4.3a}$$

$$A = \frac{4.096\ V}{58.10\ \mu A} = 70.5\ kdB\Omega \tag{4.3b}$$

As seen from the noise calculation in Section 3.3.3, this would improve the noise floor. It could however, become more susceptible to unwanted light sources and the resolution might be reduced.

It is advisable to keep the gain 10% lower than the calculated value to avoid saturating the voltage range.

Determining the feedback capacitor value

As the transimpedance amplifier is operated at a very high gain, it is prone to oscillation. To stabilize the circuit, a sufficiently large capacitor is needed in parallel with the feedback resistor.

A common approximation is:

$$C_F * R_f = C_{in} * R_{in} \tag{4.4}$$

meaning that when $R_{in} = R_f$, the feedback capacitor, C_F, should be of the same size as the input capacitance.

According to Westerman (2007), an appropriate value can be calculated more accurately like this:

$$C_F = \sqrt{\frac{C_{in}}{2\sqrt{2}\pi f_{GBW} R_f}} \tag{4.5}$$

where $C_{in} = C_J + C_{CCM}$, which are the capacitance of the diode and the common mode input capacitance of the opamp.

In the OPA129, the input capcitance is 2 pF. With 5 V reverse bias, the Hamamatsu S5991 PSD has a terminal capacitance of 500 pF. The Gain Bandwidth is 1 MHz.

$$C_F = \sqrt{\frac{502\ pF}{2\sqrt{2}\pi 1\ MHz \cdot 150\ k\Omega}} = 19.4\ pF \tag{4.6}$$

With a feedback capacitor, the output voltage V_o becomes:

$$V_o = -I_{in} * \frac{R_f}{1 + sC_F R_f} \tag{4.7}$$

And a pole is inserted at $f_p = \frac{1}{2\pi R_f C_F}$, which stabilizes the circuit.

For low frequencies the output voltage can be calculated without taking the feedback capacitor into account, as in Equation (3.8) ($V_O = -I_{in} * R_F$).

More in-depth information about this can be found in Westerman (2007).

4.3.4 Analog to Digital Conversion

An ADC (Analog to Digital Converter) is an electronic circuit that digitizes an analog signal (usually a voltage signal).

Assuming the maximum number of points the sensor would be able to discriminate is 6000, a minimal resolution of 13 bits ($2^{13} = 8192$) would be necessary. The reference biasing the transimpedance amplifier leads to an output of maximally 4.096 V.

In the first revision of the design, the AD7894-3 from Analog Devices was chosen. This is a 14-bit Successive Approximation Register ADC with an input voltage range of ± 2.5 V and a conversion time of 5 μs. It required an external 2.5 V reference to function correctly. Interfacing of the device is done with a simple serial protocol.

A successive approximation register (SAR) type ADC digitizes the analog signal by comparing the signal V_{in} to the output signal of an internal DAC. An internal sample-and-hold circuit acquires the magnitude to be digitzied and effectively freezes the signal while the conversion is executed. The output value of the internal DAC is altered by adding to or substracting from half of the remaining voltage range. In essence this means that first comparation is made with 0.5*max and then 0.25 is added or substracted to the DACs value, then 0.125 is added/substracted and so on. The digital value for the final output value of the DAC is returned as the result.

The Sigma-Delta ($\Sigma\Delta$) type converter could also be a good choice, but it is not worth the extra expense for this design. In comparison to a SAR ADC, which takes a snapshot of the signal, the resulting digital value from a $\Sigma\Delta$ device is based on averaging of the input signal over several clock cycles and is therefore less time deterministic. Averaging the signal does give some advantages with regards to noise, as the averaging will eliminate most of the noise. The intrinsic anti-aliasing properties of the $\Sigma\Delta$ architecture would be the most desirable functionality in this case, but as seen in Section 4.3.4 it isn't needed. These considerations are, for the most part, also valid for oversampling SAR ADC's. As the averaging done by these types of ADCs will reduce the time determinism, it was chosen to use the regular SAR type.

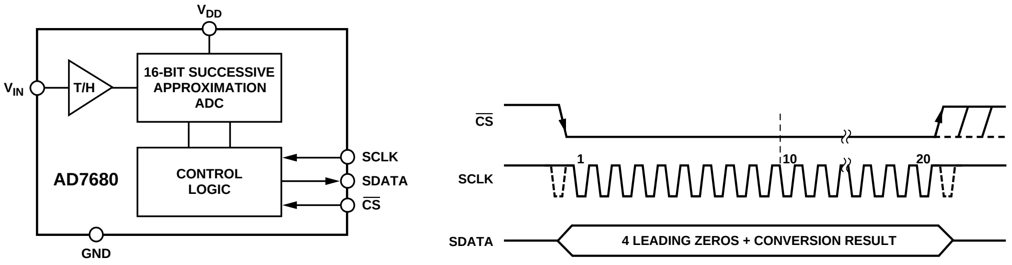
Flash ADC was not considered as this architecture is only used for very high performance ADC's and they are very expensive compared to SAR and $\Sigma\Delta$. Counting ADC was never considered either as they are only suited for low speed applications. The excellent high resolution properties of this architecture is not needed for this design either.

For the second revision, the AD7680 was implemented. It is about half the size, it's about 40% cheaper, the voltage range spans from 0 to V_{DD}, 5 V, with V_{DD} as reference. The conversion time is controlled by SCLK, the serial clock from the CPLD, with a maximal frequency of 2.5 Mhz. In the normal mode of operation it takes 20 clock cycles to complete a conversion (see Figure 4.11(b)), but the result can be read from the register after 4 clock cycles. Interfacing the device is resultingly very simple leading to a less complex logic design. This is very advantageous with regards to future maintainability and further development.

The AD7680 is a 16-bit device with a maximal sampling frequency of 100 kSps, which is much higher specifications than what is actually needed.

As this is a 16-bit device, the voltage corresponding to an LSB is 76.29 μV ($5\ V/2^{16}$), or 83.92 μV with 5.5 V supply. This is above the highest noise floor, 54.57 μV, calculated for the system, as seen in Section 3.3.3. As only 13 bits of resolution is needed, and the top bit is halved by selecting a voltage level lower than the maximum input, it would requrie more than two LSBs (168 μV) to influence the measurement. The noise is therefore negligable.

When communicating with the ADC it was necessary to use the 3.3 V LVCMOS output standard on the Max V. As the minimum high-level output voltage (V_{OH}) is $V_{CCIO} - 0.2\ V$ for this standard it is barely inside the


V_DD
V_IN
T/H
16-BIT SUCCESSIVE APPROXIMATION ADC
AD7680
CONTROL LOGIC
SCLK
SDATA
CS
GND
CS
SCLK
1
10
20
SDATA
4 LEADING ZEROS + CONVERSION RESULT

(a) Functional block diagram

(b) Operation

Figure 4.11: The AD7680. (Analog Devices, 2011)

specifications of the ADC. The minimum high-level input voltage of AD7680 with $V_{DD} = 5\ V$ is 2.8 V. A potential problem with this ADC is the fact that the logic high output voltage also follows the supply voltage, meaning it can deliver a 5 V output at logic high. Current reducing resistors are placed between the ADC outputs and the input of the CPLD and the internal clamp diodes of the CPLD are activated. The source and sink current of the device are not especially high, at 200 μA, compared to the CPLD, at 8 mA. Therefore it should not be any risk of overloading the input of the CPLD, but the resistors are still added for safety measures.

Anti-alias and HF noise filtering

The nature of the measurements done by the sun sensor renders an anti-alias filter unnecessary. As the sun angle is calculated from snapshots of the sensor data, it is irrelevant if the waveform can be represented at a later time. Any filtering for alias problems has therefore not been implemented and this is one of the reasons that the Sigma-Delta type ADC's oversampling operation wasn't considered an advantage.

There is still a possibility of higher frequency noise entering the system. An optional passive RC-filter has been implemented, as recommended by Analog Devices (2011), to reduce the likelyhood of this disturbing the measurements. By choosing a cut-off (-3 dB) frequency of 15 kHz, the attenuation at 5787 Hz is negligable.

The combination of this passive filter and the active filter formed by the feedback capacitor, C_F, in the amplifier effectively realizes a two poled low pass filter. Adding to this, capacitance in the PSD's terminals, the inputs of the ADC and the wiring will all attenuate higher frequencies to a degree, but are in practice negligable in comparison to the two-pole filter.

4.3.5 Voltage reference

A voltage reference is used to bias the transimpedance amplifiers, to get a positive signal on the outputs. In the first revision of the design, two ADR3425 2.5 V references was used as the ADC of the first revision required an external reference. When the voltage range used in the system was altered, from 0-2.5 V to 0-4.096 V, it was necessary to choose a different reference. The ADR3440 4.096 V voltage reference from Analog Devices was selected, as it is a direct replacement for the ADR3425, except for the output voltage level. This voltage level, 4.096 V, was chosen because it was the highest found on a reference circuit supporting a 5 V supply voltage.

4.3.6 Logic design

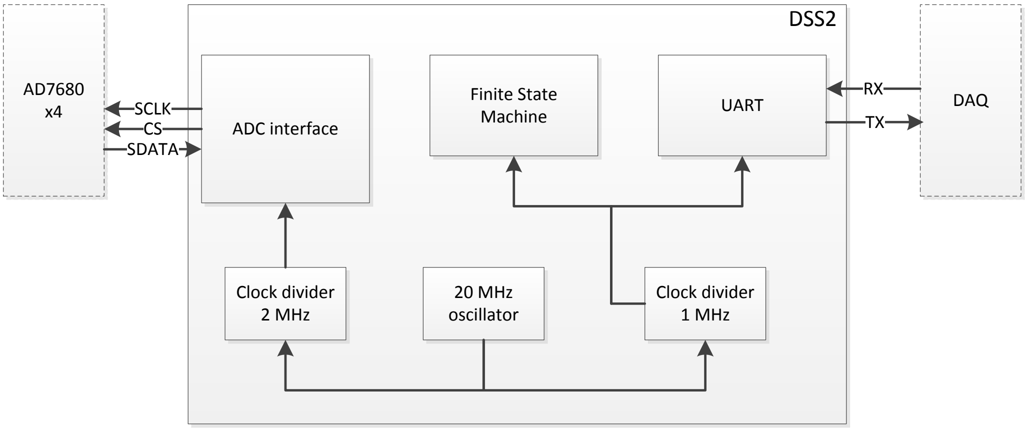


Figure 4.12: Logic schematic of DSS2.

The logic implemented in the sun sensor has a very simple job. It collects the data from the four ADC's, inverts the results and sends them to the DAQ via UART (RS-422). The rate of which the measurements are conducted is controlled by a control line from the DAQ, which is set high when new measurements are required.

The CPLD chosen for the task is an Altera Max V with 1270 logic elements in a TQFP-144 package. While this is a large package, it was chosen for the ease of debugging and mounting. The initial plan was also to fit the calculations of X and Y in the CPLD. It was not possible to fit the calculations on the CPLD if adequate accuracy were to be achieved, as the dividers require copious amount of logic elements. An even larger CPLD would be needed in that case. Because of this, the calculations are fitted on the FPGA of the DAQ.

To implement the algorithm, fixed point arithmetic has been chosen. Fixed point representation works by assigning a fixed number of bits before and after the decimal point. It is crucial to know this ratio when decoding it.

VHDL code and ASM-diagrams are found in Appendix B.3.2.

4.3.7 Power Supply

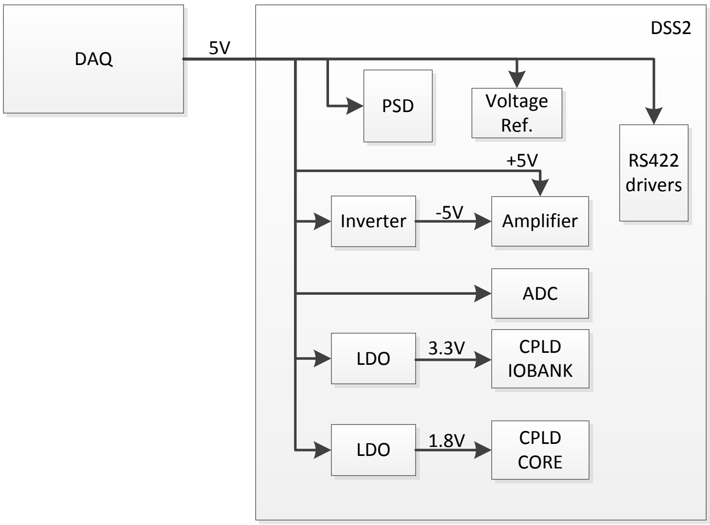


Figure 4.13: Power distribution network for the DSS2-board.

Two TI TPS717 low dropout regulatorors are used to source the 1.8 V and 3.3 V voltages. They are described in Section 4.2.2. All 5 V components are powered directly by the DAQ-board, filtered at the input by a single 10 μF capacitor. The power supply network is illustrated in Figure 4.13.

To supply the negative voltage (-5 V) for the amplifiers, a TI TPS60403 voltage inverter is employed. Since the output voltage of the amplifier never passes below 0 V, the current load is very low and an inverter was therefore an acceptable choice. With a output filter capacitor of 1 μF, the ouput ripple is specified to 15 mV_{pk-pk} at 5 mA output current. The high switching frequency of the TPS60403 (typ. 250 kHz) makes it easy to filter and it should have a negligable effect on the total noise. Practical measurements of the noise are presented in Section 6.3.2, which reveal that this assumption was incorrect.

4.4 Rate gyro

The Sensonor STIM210 gyroscope module is chosen for this system. It is temperature compensated and calibrated from the manufacturer, and has a very low drift. This should ensure correct measurements, even when experiencing fluctuating temperatures. The measurement principle is described in Section 2.1.1 and a block schematic of the module is presented in Figure 4.14.

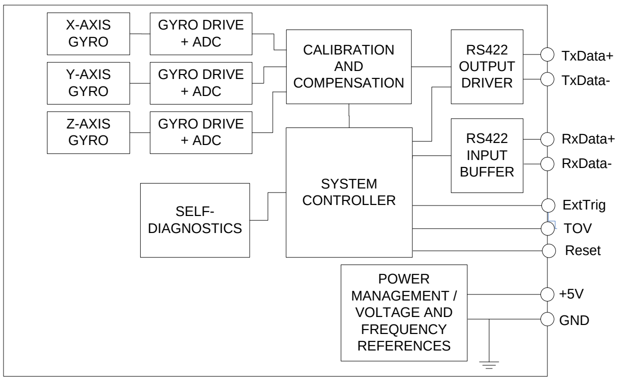


Figure 4.14: A block schematic of the STIM210. (Sensonor, 2013)

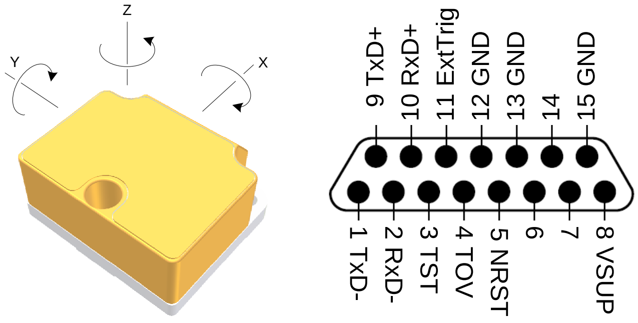
Compared to the old gyroscopes, it is much smaller and lighter. The STIM210 measures 39x45x22 mm and weighs 52 grams.

Being a readily manufactured module saves a lot of assembly and calibration time. Having all 3 axes made on a single chip will also give much better control of misalignment errors, compared to individual gyro chips mounted on different PCBs. In the ICI-4 rocket the gyro will be placed on top of the UiO stack (see Section 5.2) in the middle of the rocket.

The STIM210 unit has a built-in RS-422 interface, which is used to transmit the rate and temperature measured. A status byte is transmitted with each dataset, which contain the results of the continuous self-diagnostic of the unit. In this design the status byte is transmitted to the encoder as housekeeping data, but the implementation of logic to check if the gyro is operational should be effortless to implement. As the gyro has a separate reset input, the DAQ could be programmed to trigger a reset of the module if required.

The maximal sample rate of the STIM210 is 2000 SPS, and it can be set to sample and transmit data continuously or to rely on an external trigger signal from the DAQ. A continuous sampling of 2000 SPS is used in this thesis.

The only processing that needs to be done before sending the gyro data to the telemetry encoder is to convert the 24-bit 2's complement value to a 16-bit value. As the 16 MSBs actually have a 1's complement representation when separated from the LSBs, this is as simple as adding 1 to the result for the negative numbers.



(a) (b)

Figure 4.15: Pinout of the connector on the STIM210 gyro and relation of the gyro axes. (Sensonor, 2013)

On the rocket the maximal rotational speed is usually up to about 2160°/s. As the regular version of the STIM210 has a full-scale resolution of ±400°/s, a specially modified version was ordered. This version has a full-scale resolution of ±2000°/s on the Z-axis, linearly to ±1800°/s. The X and Y axes are unmodified. As the rocket rarely reaches more than 4.5 RPS it should not be a problem that the maximal resolution is ±2000°/s.

4.5 External Magnetometer

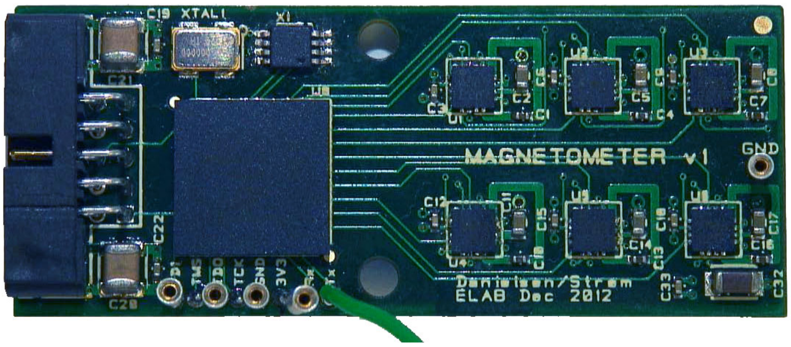


Figure 4.16: The External Magnetometer.

An external PCB with 6 magnetometers is placed on a boom outside the rocket, communicating with the DAQ by LVDS at a rate of 115200 bps. It is designed by Halvor Strøm and Roar Danielsen at the ELAB, Dept. of Physics, University of Oslo.

The magnetometer used on the DAQ-board, the Honeywell HMC5983, is also employed in this instrument. Equipped with a small CPLD, it is capable of delivering a serialized stream to the DAQ.

4.6 PCB design

Two prototype PCBs has been produced as part of this thesis, one for the sun sensor and one for the DAQ. This section covers some of the most important principles of PCB design. Design choices made when designing the two PCBs are explained and the signal and data flow of the boards are illustrated. The layout of the DAQ PCB is pictured in Figure 4.18 and Figure 4.19, and the DSS2 PCB is pictured in Figure 4.20 and Figure 4.21.

Complete PCB layouts are found in Appendix A.1 and Appendix A.2 with a summary of the design revisions included in Appendix A.3.

4.6.1 Component placement

To reduce noise emission and interference, careful consideration should be given to component placement. All wires should be as short as possible, especially those connecting drivers with short rise times and high current capabilites.

Oscillators are large emittors of noise and should be placed as far as possible from analog components, connectors and other sensitive circuits. It is crucial to reduce the length of the clock wiring, as it may be the most instense source of emission on the board, if not accounted for.

Grødal (1997) recommends designing the layout with the analog and digital components separated to minimize interference. If this is done properly, emitted noise from the digital components will have little effect on the analog components.

4.6.2 Ground planes

A ground plane guarantees a relatively clean ground signal (0 V) for all components in the circuit. According to Grødal (1997), a noise reduction of 10-20 dB is achieved solely by the addition of a ground plane in four layer PCBs compared to two layer cards.

Another immediate effect of adding a ground plane is a slight increase in capacitance between signal wires and ground. While this may have the disadvantage of limiting very high frequency signals, it is usually an advantage. The added capacitance to ground will reduce the parasitic capacitance between conductors, leading to a reduction in crosstalk. (Grødal, 1997)

The ground plane is actually acting like a shield, which can be utilized for several purposes, e.g. shielding of sensitive components on one side from noisy components on the other like on the DSS2 (see Section 4.6.5). By placing an additional ground plane on the top or bottom layer, shielding around the signal wires is formed, reducing cross-talk and possibly adding some protection from external noise as well.

As digital devices in general produce more noise than analog components, an often used approach is to split the ground plane in digital and analog ground. This is usually done by splitting the ground plane in two separate planes, with only a narrow connection between the parts. While this may help reduce noise in the analog domain in some cases, the actual benefits of this technique are much debated. One of the main problems is to predict where the return currents will go. If a signal wire crosses the split, the return current might take "the long route home", which may lead to increased interference.

Because of the unpredictable nature of a split ground plane, it has been chosen to not do this in these layouts. It's a better choice to reduce noise by separating digital and analog components. (Grødal, 1997)

4.6.3 Bypass capacitors

Bypass capacitors are used for two reasons: to keep the circuit functioning correctly and to increase the noise immunity of the system.

Bypass capacitors are needed for digital circuits to function correctly because of the way a digital circuit operates. When the output stage of a digital circuit switches value it results in a short circuit current, giving a dip or peak in the current draw of the circuit. This leads to transients in the supply voltage, which, if not attended to, can lead to a voltage deviation in the output of the circuit. A large enough voltage deviation in an output stage might mean that the receiving circuit will receive something different than what was sent. In the worst case it might also lead to a circuit freezing up and simply not operating.

It's obvious that this can be a significant problem. The solution is, however, quite simple. By adding a capacitor between the leads to ground and the power supply, most of the transient can be eliminated by the capacior discharging (filling a dip) and charging (smoothing a peak). As the frequencies best coupled by the capacitor is inversely proportional to the capacitance it is often neccessary to use different values of capacitors in parallel.

As there are no ideal capacitors in the real world, it may be necessary to place several capacitor values in parallel to filter a broader span of noise frequencies. The packaging of the capacitors has some stray inductance, which will cause a rise in the impedance of the bypass connection at higher frequencies. This means that it will only conduct the noisy current surges at a certain frequency span, above which the bypass capacitor no longer provides a low impedance. If different sizes of capacitors are connected in parallell their combined attentuations will "add up" and create a wider frequency span of safely decoupled dips and peaks. It may also be necessary to choose different capacitor packages to achieve this, like 0603 and 0402, which have different inductive properties. (Bogatin, 2004)

The choice of capacitor material is also an important consideration, as different types have differing characteristics that make them suitable for different uses. A tantalum capacitor is often used due to its high capacitance per volume, low ESR and leakage. They are also rated for higher operating temperatures than other electrolytic capacitors. For smaller circuits with more limited current impulse demands, ceramic capacitors are normally used. A usual default value when there aren't any specific demands is $\sim$100 nF.

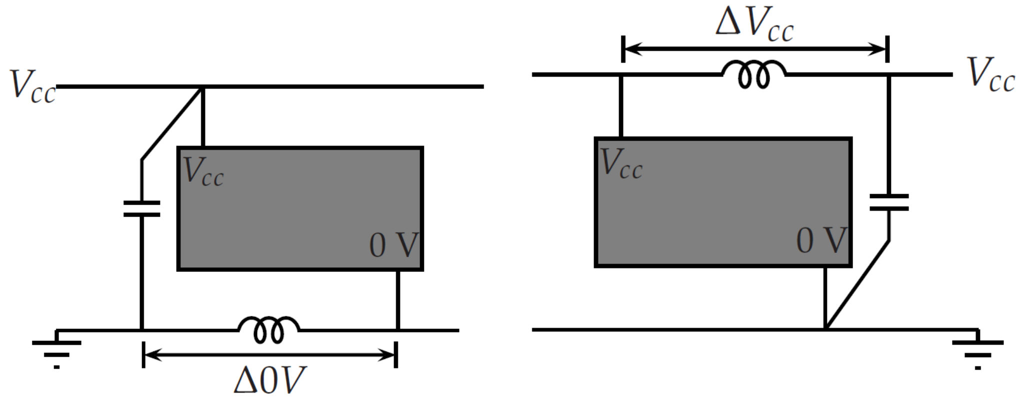
While there are differing opinions on how the bypass capacitors should be connected, it has been chosen to use the approach of Grødal (1997). This approach is basically to connect both the power and ground signals through the terminals of the bypass capacitor, to generate the correct current loop as seen in Figure 4.17. Care must be taken so that the vias don't connect to the power and ground planes on both sides, if the capacitor is connected through vias.

Further reading on the subject can be found in Grødal (1997) and Bogatin (2004).

4.6.4 Layout of the DAQ PCB

The PCB layout of the DAQ is pictured in (Figure 4.18(a)). It is designed with modularity in mind, as can be seen in the pictures, where related components are grouped together in modules. Placing the components in this manner reduces interference between different parts of the circuit, but it will also result in a simplified debugging process. As the signal flow can be followed by visual inspection, the circuit schematics won't have to be consulted for every part of the design. All the power and ground routes are placed on the bottom layer of the PCB and most of the signals are routed on the top layer.

The modularity of the design should help facilitate future updates to the



(a) Wrong connection

(b) Right connection

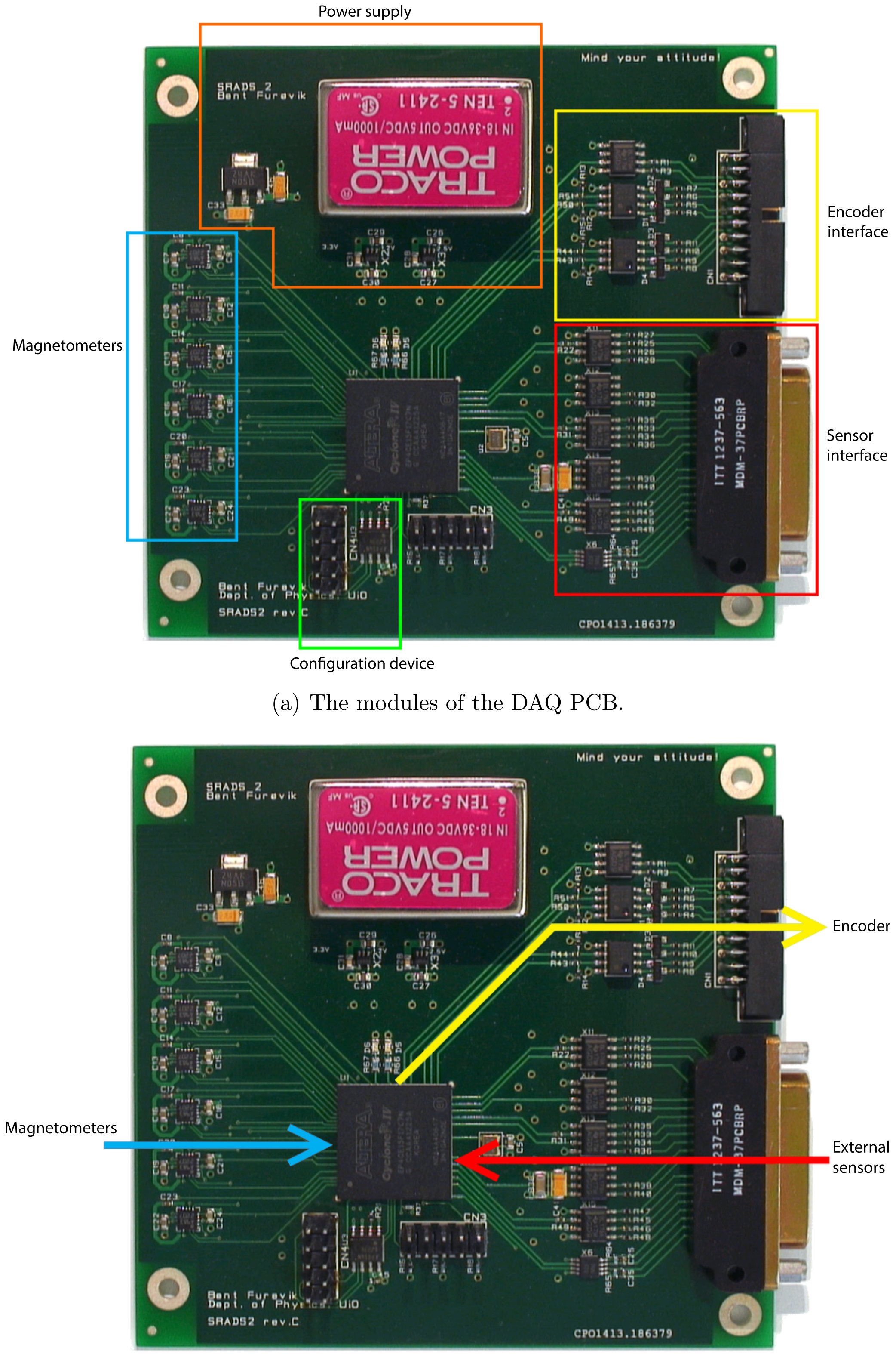
Figure 4.17: Placement of bypass capacitors, as decribed in Grødal (1997). Figure taken from Bekkeng (2009).

design, e.g. a common power supply for the UiO-stack (Section 5.2). The common power supply will most likely be connected where the magnetometers are placed in this version, but they are easily moved and rotated 90° in the CADStar software. It is then only a matter of "connecting the dots", i.e. routing the signals ends to the FPGA once more.

The control signals for the Sensonor STIM210 (Reset, Time of Validity and External Trigger) are routed in the bottom layer. Using the bottom layer was deemed the best option, as placing the signals in the top layer would make the routing very tight. It can be seen in the route of the bottom layer in Figure 4.19 that the 3.3 V power plane (which covers the bottom layer) is fractioned because of this. As the only components located in and drawing current from the lower fraction is an LVDS driver and the external magnetometer, the effects of this should be minimal. These components have very low power consumption, and the currents going through the plane should not be large enough to cause any real problems. Routing the signals in the power or ground planes could make much bigger problems, as the more power hungry RS-422 drivers might create more noise or potentially large ground loops.

To increase the shielding capabilities of the ground planes, the power planes are recessed from the edges of the ground plane by a distance equal to 20 times the spacing between the planes. (Ott, 2011)

The complete schematics and PCB layout for the DAQ are found in Appendix A.1.



(a) The modules of the DAQ PCB.

(b) The data flow on the DAQ PCB.

Figure 4.18: The modules and data flow of the DAQ PCB.

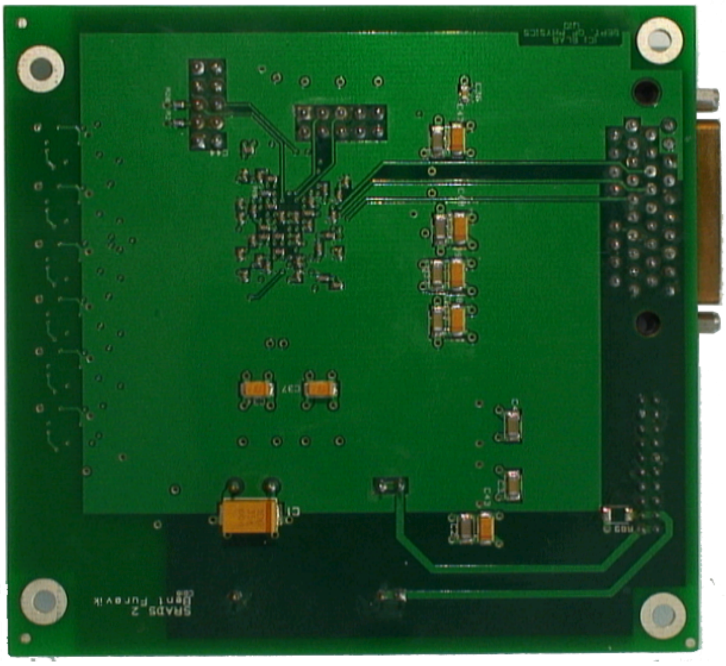


Figure 4.19: The bottom side of the DAQ PCB.

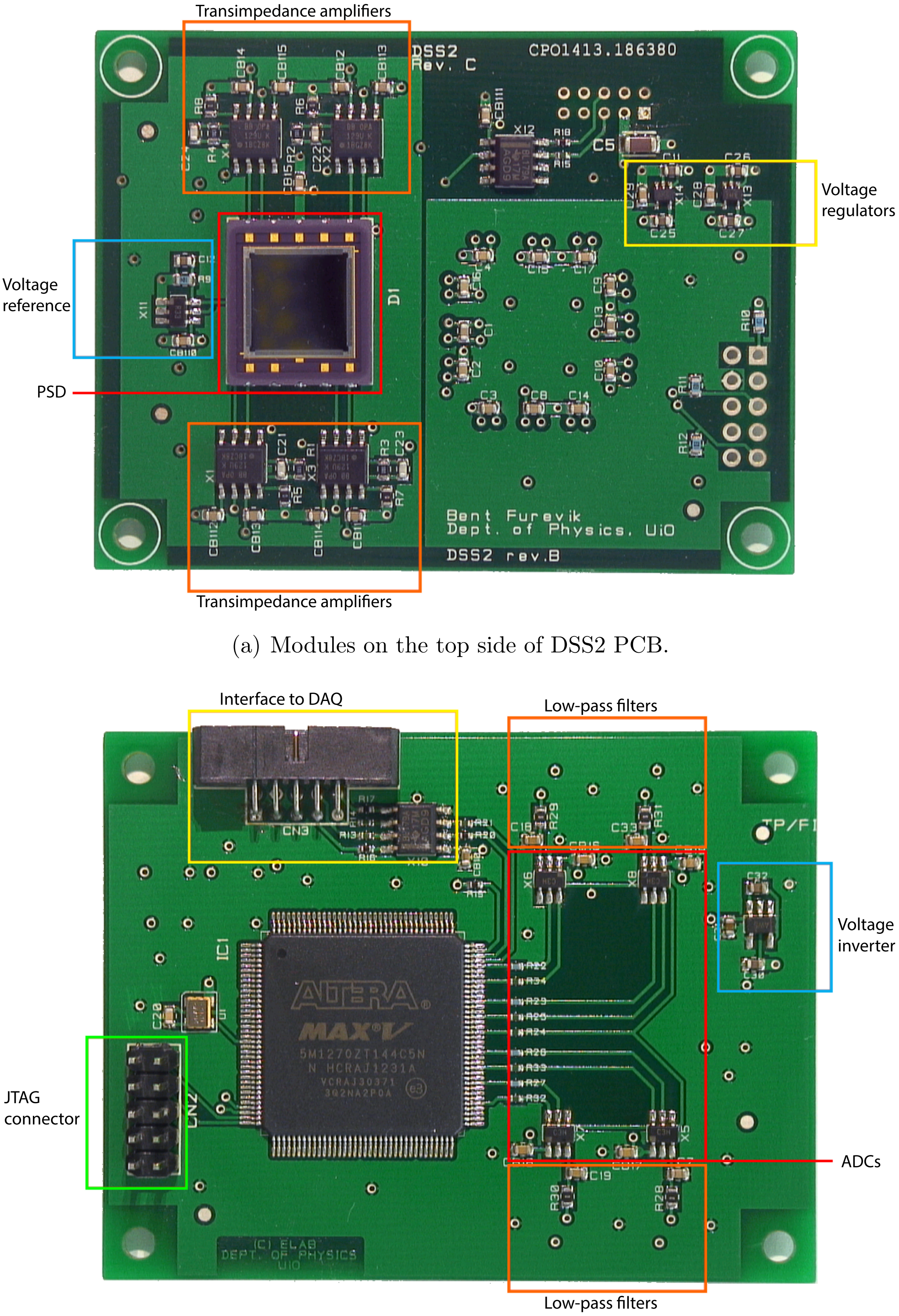
4.6.5 Layout of the DSS2 PCB

The routing on the DSS2 PCB is done in a similar manner. To reduce interference, the sensor and amplifiers are placed on the top layer and the ADCs and CPLD are placed on the bottom. As only one of the IO-banks of the CPLD has clamp diodes, both of the RS-422 drivers had to be connected to the same bank. To work around this limitation, the drivers were placed on opposing layers of the board. This makes a short route to the drivers possible.

To reduce the analog signal conductor length, the output signal from the amplifier is wired to a via very close to the output. The ADC is connected close to the via on the other side of the board, only separated by a passive RC-filter. This very short track should reduce noise pickup well.

The power planes for the 3.3 V and -5 V supplies of the DSS2 are routed very visibly on the top layer of the PCB. Power planes for 1.8 V and 5 V is routed in the same pattern in the internal power layer. If signals are routed across the splits between the power planes, an interrupted return current path will be created. This would increase the noise emission of the board.

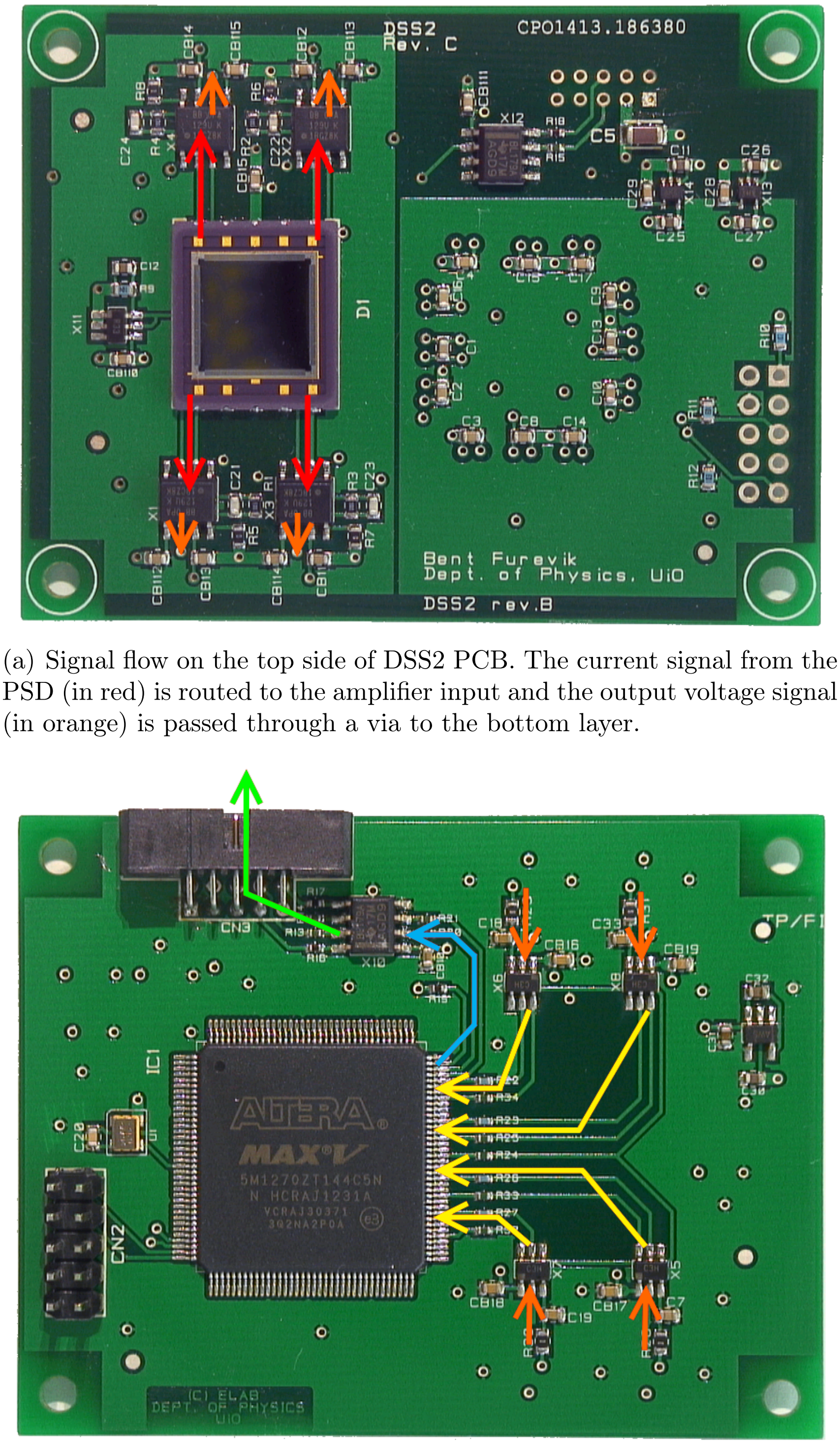
To reduce emitted noise, the ground planes covers the board almost all the way to the edge, while the power planes are constrained further in. This improves the shielding. (Ott, 2011)



(a) Modules on the top side of DSS2 PCB.

(b) Modules on the bottom side of DSS2 PCB.

Figure 4.20: The modules of the DSS2 PCB.



(a) Signal flow on the top side of DSS2 PCB. The current signal from the PSD (in red) is routed to the amplifier input and the output voltage signal (in orange) is passed through a via to the bottom layer.

(b) Signal flow on the bottom side of DSS2 PCB. The voltage signal passes through a LF-filter and is digitized by the ADC, which sends the digital signal (in yellow) to the CPLD. After minor processing, the CPLD sends the data stream (in blue) to the RS-422 driver, which transmits the stream differentially (in green) to the DAQ.

Figure 4.21: The signal flow of the DSS2 PCB.

While there are no signals crossing the split on the analog side of the board, all the terminals of the CPLD are located directly above it and may thus be affected. Although the current levels are small, the bottom side of the board is covered by a ground plane to add decoupling of this noise source. This will also decouple noise from the voltage inverter, as it is located in the middle of the ground plane.

The PCB will be oriented with the PSD down and the PSD is placed in such a way that the X-axis of the PSD is the vertical axis of the sun sensor.

The complete schematics and PCB layout for the DSS2 are found in Appendix A.2.

Chapter 5

Mechanical Design

This chapter details the mechanical design for the intrument boxes and calibration fixtures. All the enclosures are made in 7075 space-grade aluminium.

5.1 DSS2

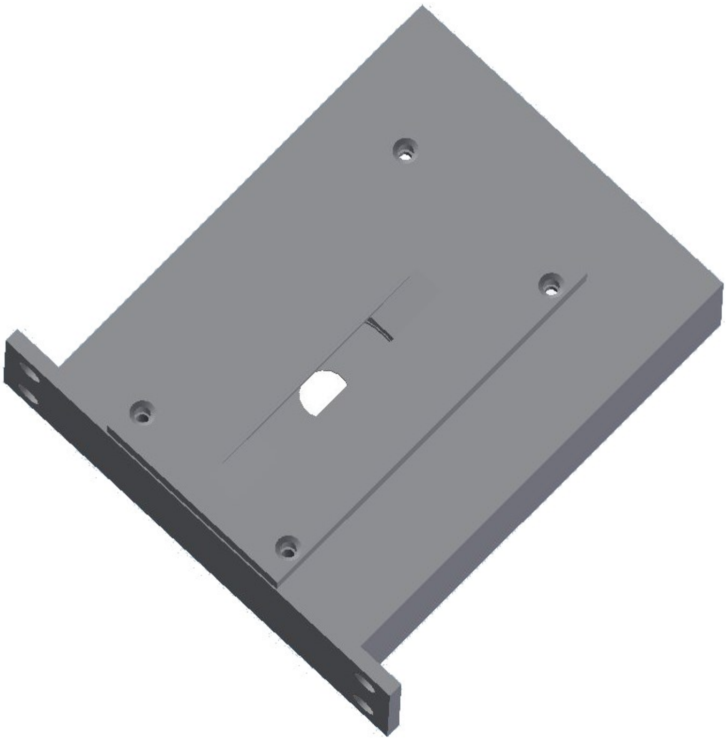


Figure 5.1: Front view of the DSS2 enclosure.

The sun sensor box, seen in Figure 5.1 and Figure 5.2, was designed by staff engineer Sverre Andre Hegg at the Instrumental Workshop, Dept. of Physics, UiO.

Distance from the PCB to the box bottom is 3.4 mm, because of a miscalculation in the early development of the instrument. This will be decreased in a later unit to the requried 2.6 mm.

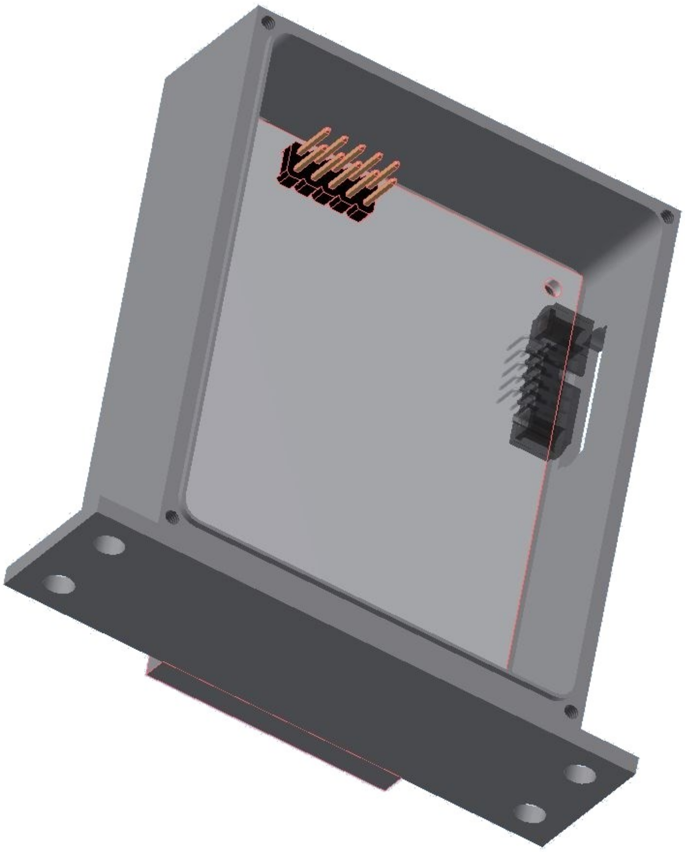


Figure 5.2: Rear view of the DSS2 enclosure.

5.2 DAQ and the complete UiO-stack

The enclosure for the DAQ, as seen in Figure 5.3 were designed by staff engineer Sverre Andre Hegg at the Instrumental Workshop, Dept. of Physics, UiO. This version has special connectors that are used by the DAQ in this thesis. Figure 5.4 depicts the complete UiO-stack, which will contain all the instruments and the gyro mounted on top.

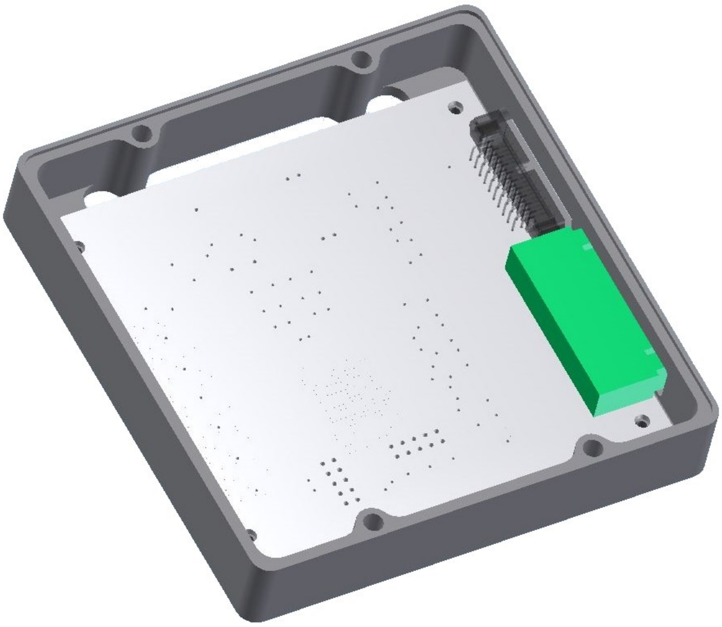


Figure 5.3: The DAQ enclosure, depicted with a simplified board model.

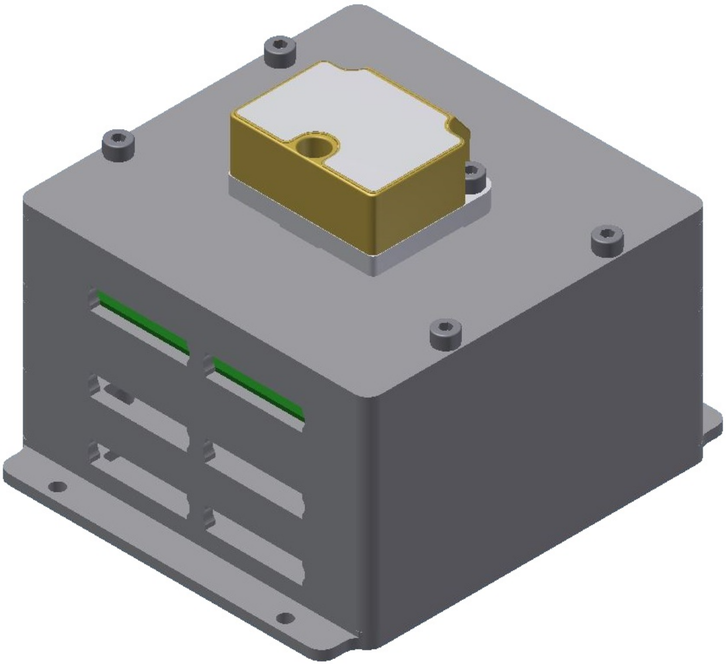


Figure 5.4: The complete UiO instrument stack.

5.3 Gyro mounting

Figure 5.5 shows a mechanical model of the Sensonor STIM210 gyro module.

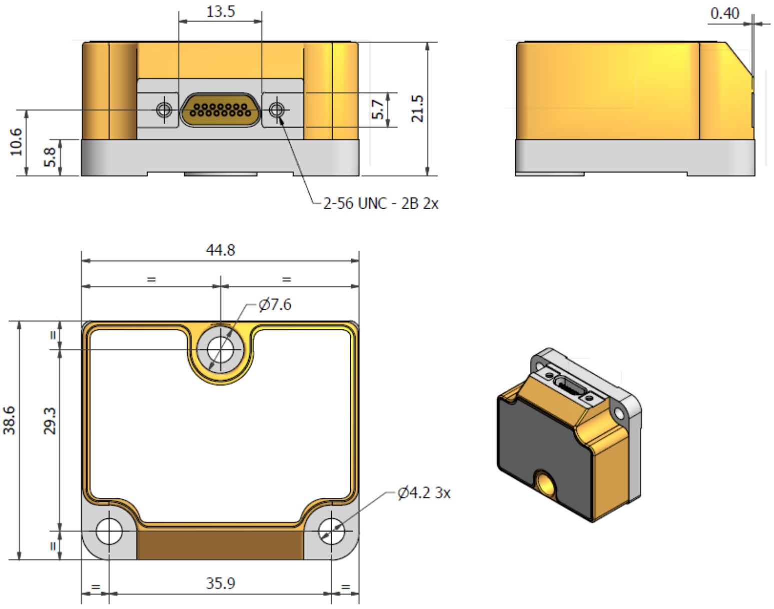


Figure 5.5: A 3D model of the Sensonor STIM210. (Sensonor, 2013)

Chapter 6

Performance analysis and calibration

Before the measurements can be used, the accuracy must be verified. This section covers the calibration and verification process, and the results of this.

As the system has not been launched yet, there are no flight results to present.

Measurements of the noise level in the system is also included, as it is important to find weak points in the design. If a part of the circuit is noisy, it may degrade maximum performance and reliability. It is important to find any such issues, so they can be fixed in the future.

6.1 DSS2 Verification

This section covers verification and calibration of the sun sensor. Results of the tests and a description of the measurement setup is also provided.

6.1.1 Calibration setup

A prototype calibration fixture was constructed by head engineer Thor Arne Agnalt at the Instrumental Workshop, Dept. of Physics, UiO. The requirement for the fixture is that the pinhole stays centered when the box is rotated and tilted. Done this way, the measurements can be used to predict real world behaviour.

A SmartMotor, from Animatronics, is used to control the tilt of the sun sensor. The horizontal rotation is controlled by a rate table from Ideal Aerosmith, which the structure is mounted on top of.

The light source is a 55 W car headlight. Since the sun sensor utilizes the pinhole camera principle, the homogenity the light source is a bit less important as it has to pass through the small pinhole. However, a better light source should be used in the future to get more accurate measurements, e.g. a laser with a beam expander or an integrating sphere. Ideally, a point-source at infinite distance with an angular radius of 0.53° should be used, since this is the size of the sun seen from the measurement point in space. However, this is difficult to achieve. The initial tests were made with a laser, but it proved difficult to hit the pinhole correctly as the spot was very small. Because of this, the laser was substituted for the headlight.

Together with misalignment of the test setup, the light source is the largest error source as its light is largely uncontrolled. Inaccuracies in the servo and the rate table are also contributing to the error.

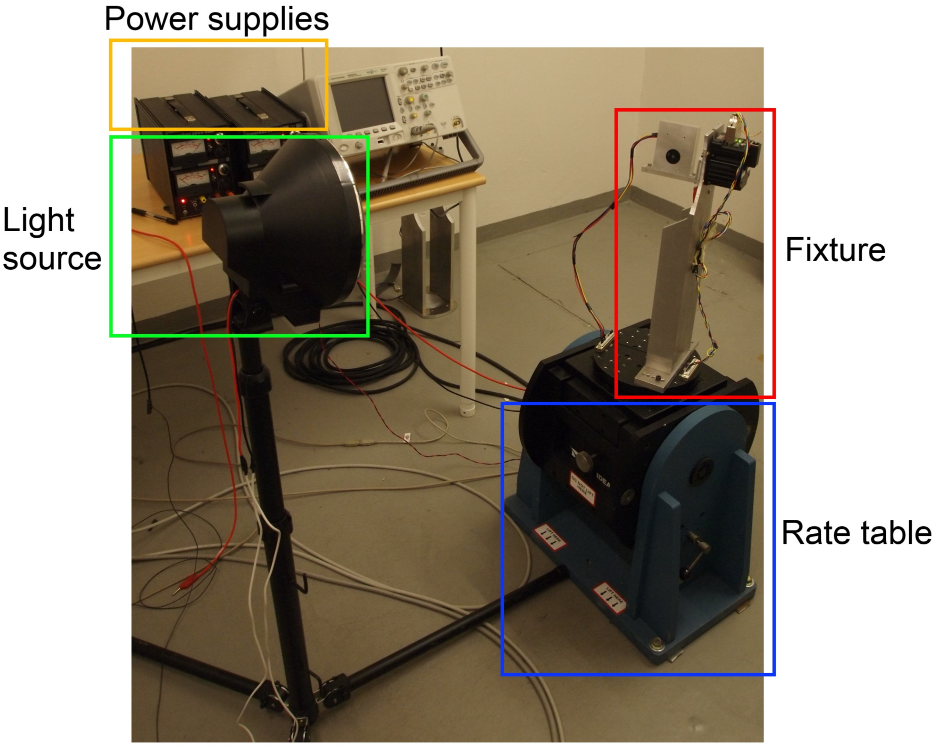


Figure 6.1: The complete test and calibration setup. The power supplies power the sun sensor and the servo, while the rate table is controlled by a separate servo controller.

A LabView Virtual Instrument (VI) controls the testing process. A screenshot of the LabView Virtual Instrument (VI) is included in Appendix D.2 This is a modified and stripped version of the routine used by Bang (2013).

The VI makes the servo tilt the sun sensor from −60° to 60° in 0.4° increments. For each increment in the tilt angle of the servo, the rate table rotates from −60° to 60° with 1° increments.

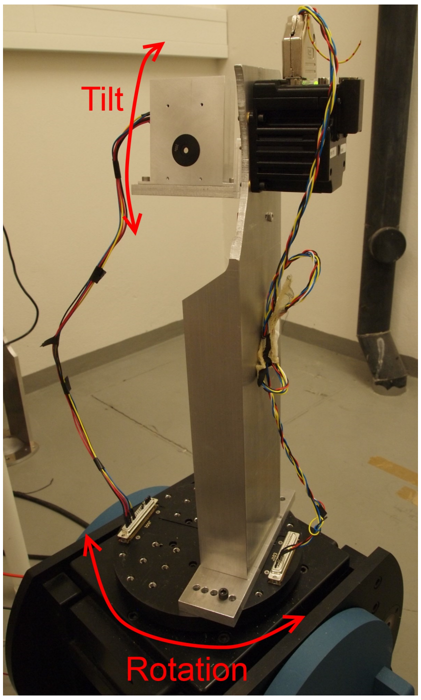


Figure 6.2: The test and calibration fixture mounted on the rate table.

Calculations and visualization of the characteristic curves are done with a Matlab script (made with assistance from Bang (2013)), which produces the plots in Section 6.1.3. The displacement and distortion of the measurements can be read directly from the results, making compensation in post processing realizable with little required effort.

The Matlab code used is found in Appendix C.2.

6.1.2 Characterization of the input stage

When the first measurements with the car headlight were made, it quickly became evident that something was not working as intended. After debugging the system, it was found that the ± 5 V supply for the opamp was 0.5 V too low to reproduce the 4.096 V output. The opamp was operating in saturation until the input signal was large enough, as can be seen in Figure 6.5. The OPA129 used in this design is not a rail-to-rail model, and will therefore require a higher supply voltage than the output voltage. A log of the debugging is included in Appendix D.1.

When the voltage was increased to ± 5.5 V, the problem was eliminated. There are, fortunately, other alternatives to fix this as well: the reference could be swapped for a 3.3 V unit, and the gain changed accordingly or the opamps could be replaced, which would require a redesign. A detailed discussion of the options are found in Future Work in Section 8.3.2.

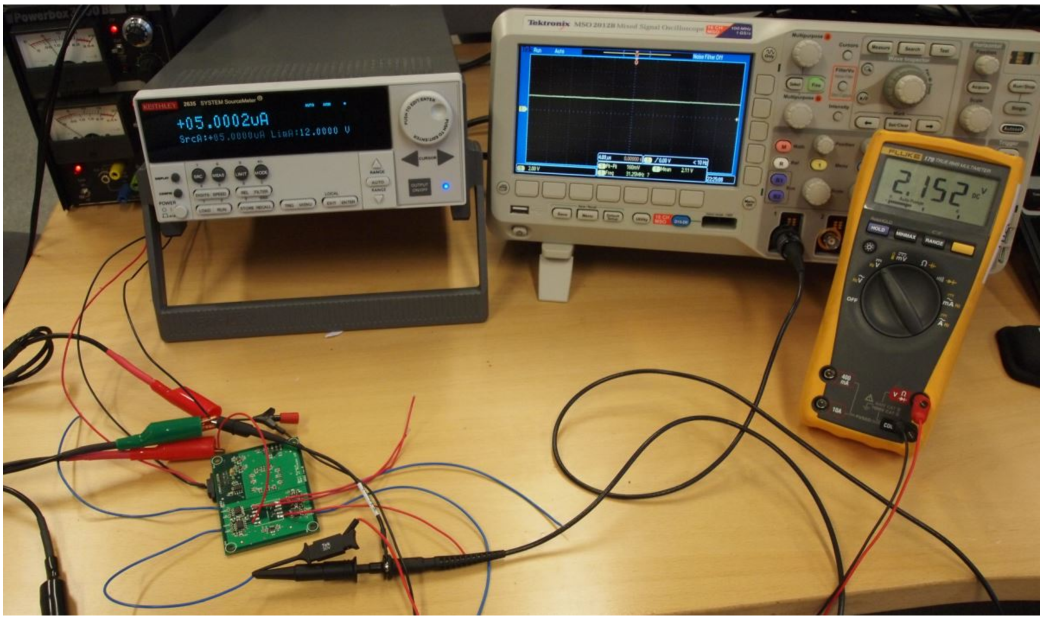
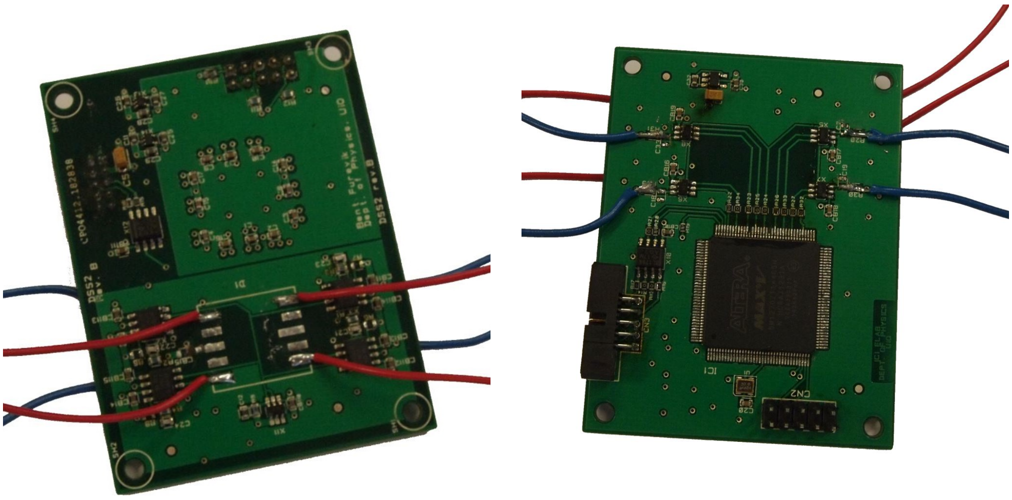

KEITHLEY 2635 SYSTEM SourceMeter
+05.0002uA
SrcA:+05.0000uA LimA:12.0000 V
Tektronix MSO2012B Mixed Signal Oscilloscope
FLUKE 179
2.152 DC V

Figure 6.3: The measurement setup for the input characterization.

To perform the measurements, a Keithley 2635 Source Meter was utilized as a very accurate current source. While the output current of the Keithley was increased in 1 μA increments, the output of the amplifier was observed with a Textronix MSO2012B oscilloscope and a Fluke 179 DMM. The results are depicted in Figure 6.5. It was necessary to remove the PSD from the PCB and solder wires on the inputs and outputs of the amplifier, as seen in Figure 6.4. The measurement setup is pictured in Figure 6.3.



(a) Top side of the modified PCB.

(b) Bottom side of the modified PCB.

Figure 6.4: The modified PCB used for the input characterization measurements. The red wires are soldered to the inputs of the amplifiers and the blue wires to the outputs.

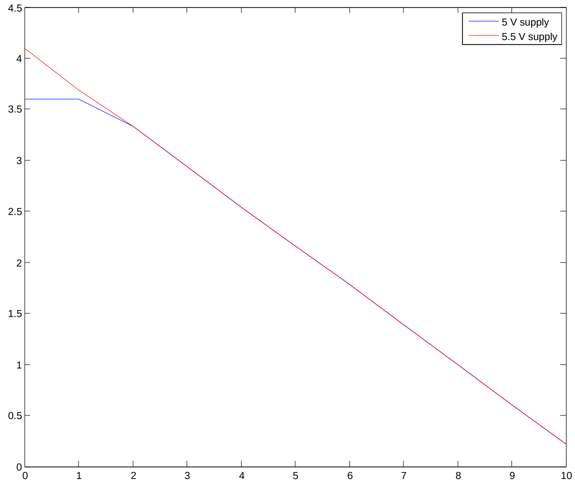


Figure 6.5: Characteristics of the transimpedance amplifier. As seen in the plot, the linear response of the amplifier with 5 V supply is compromised until the input current exceeds 2 μA. $R^2 = 1$, meaning that the amplifiers response is completely linear in the measured range.

6.1.3 Measured performance

These measurements are done to estimate the accuracy of the sun sensor. They are performed with the calibration setup described in Section 6.1.1.

As seen in Figure 6.8, the measurements are quite linear, although a wave-like pattern is seen in the measured plane. The field of view is ± 45°, revealing an earlier specification error of the distance between PSD and pinhole. This means that the pinhole is too far from the PSD, which can be fixed by shortening the distance between the PCB and the enclosure. Making a new enclosure with a shorter distance from the PCB to the front of the box should not be necessary. This is commented in Section 5.1.

The wave-like pattern appearing in the measurements may be caused by several factors. Some diffraction will be present, but this would be a minor problem when the pinhole size is larger than the wavelength of the incident light.

As discussed in Section 3.2.1, the PSD is an analog sensor measuring the location of the light spot. Through the test and calibration sequence, the light spot is moved across the entire active area of the PSD. In the larger angles, the light spot will take on a more elliptic shape. This will skew the measured position, since the sensor measures the center point of the light spot.

Depicted in Figure 6.6(a) is a sketch showing how the light will reflect in the enclosure. A closer look is given in Figure 6.6(b), where variables for the pythagorean theorem are included. As the diameter of the pinhole is very small, it is neglected. The reflection angles are the same, which means that the a distances in Figure 6.6(b) will have the same length. This makes it simpler to calculate where reflections will appear.

It is assumed that both the sensors active area and the enclosure have smooth surfaces with uniform reflection patterns. This is a very simplified expression, as the pinhole is mounted in a slot in the front of the enclosure, which means that there is a gap between the reflection plane of the enclosure and the pinhole. The gap will have a certain height and could be a source of reflections at certain angles as well. Even the pinhole itself has three different planes in its structure, as seen in Figure 6.7.

Expressed mathematically, the distance between the center point and each reflected light spot can be expressed as in Equation (6.1).

$$X_n = X_d + 2nX_d \tag{6.1}$$

where X_0 is the actual light spot.

A reduction in intensity will occur, depending on the reflection coefficients of the materials, which can be expressed as Equation (6.2). The reflection coefficient of the aluminium is assumed to be constant through the whole of the the detector spectral response. As the coefficient for the PSD sensor is unknown, it is expressed as an integral over the whole spectrum.

$$p_n = \left(\int_{\lambda_1}^{\lambda_2} \eta_s(\lambda) d\lambda \cdot \eta_A \right)^n \tag{6.2}$$

To express a total measured displacement, it is assumed that the sensor would sense two light spots of identical intensity as an ellipse with the center point between the two light spots. If the two light spots have different intensity, it is also assumed that the measured center point would be skewed towards the spot with the highest intensity, proportionally to the intensities. To find the measured displacement, the contributions from the light spot and each reflected spot must be summed. This can be expressed as in Equation (6.3).

$$X_M = \sum X_n p_n \tag{6.3a}$$

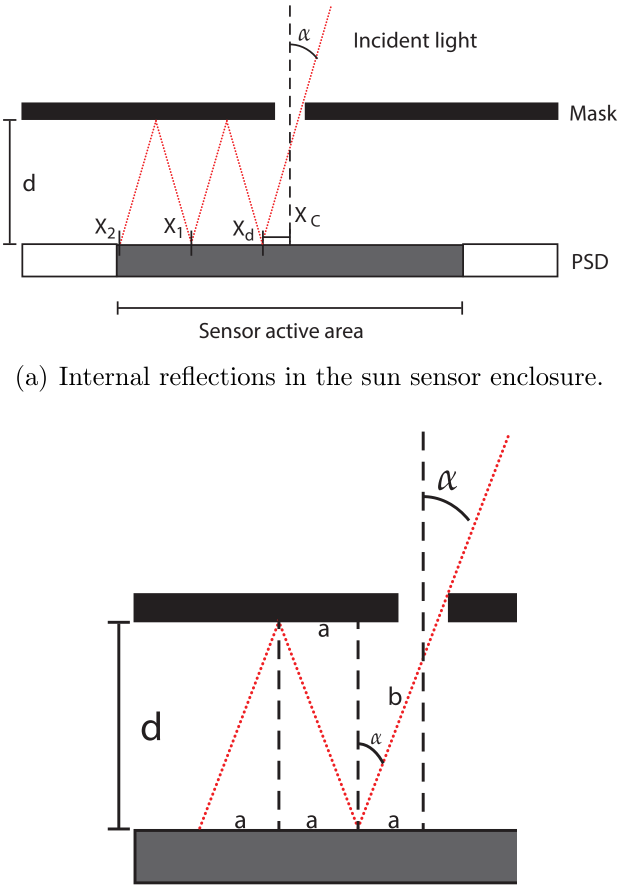
Although this is a simplified expression, the principles can be used to calculate reflections in the gap between the enclosure and the pinhole. A real world calculation would have to be calculated manually or through a model of the enclosures insides, as the different planes would lead to different reflections. The reflection coefficients may vary between the three materials in the pinhole structure, and the reflection coefficient of the PSD sensor is likely to vary with angle. As the surface of the milled aluminium is not completely even, the reflections inside the enclosure will not behave like assumed.

It is assumed that the pinhole mount and the uneven surface of the actual milled aluminium is the reason for the wavelike pattern that appears when the light spot is displaced from the center. As there are four different surfaces when combined with the mount, there are several planes that can reflect light in each direction.

Further plans for the enclosures are to paint them internally with non-reflective paint. This should reduce the reflections significantly, although infrared light may act the same as it is difficult to damp its reflection.

The noise from the voltage inverter, seen in Section 6.3.2, may also be a contributing factor, but this would lead to general noise in the measurements. A wave-like pattern like the one seen in the measurements would not be created by switching noise, as it would be more random. Almost identical results were had when the measurements were retaken, which eliminate this possibility.

The curvefitting is done with a ± 40° field of view. As the X and Y axes carry almost identical distortions, it was chosen to only curvefit the detectors



(a) Internal reflections in the sun sensor enclosure.

(b) A closer look at the internal reflections.

Figure 6.6: Internal reflections in the sun sensor enclosure, assuming a simplified enclosure model where the pinhole is a hole in the metal with no internal reflections. Assuming homogenous surfaces of the aluminium inside the enclosure and the active area of the detector. As the diameter of the pinhole is very small, it is neglected.

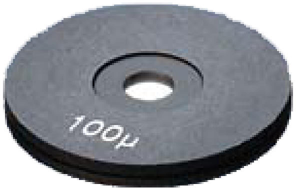


Figure 6.7: The pinhole used in this thesis. Three planes are present in the structure, the metal circle with the actual pinhole, the inner plastic circle and the outer plastic circle. (Edmund Optics, 2013)

X-axis, which is the sensors Y-axis.

Figure 6.10 depicts the curvefit with a 1 degree linear polynomial and its residuals. It is evident that this is not a good fit. The reflected pattern does not make for particulary linear measurements. The achieved accuracy with a 1 degree linear polynomial fit is approximately ± 8°, with most of the range measuring much better. In Figure 6.11, a 5-5 degree linear polynomial has been used to make a fitting. The achieved accuracy with a 5-5 degree linear polynomial fit is approximately ± 2° through the majority of the fitted range.

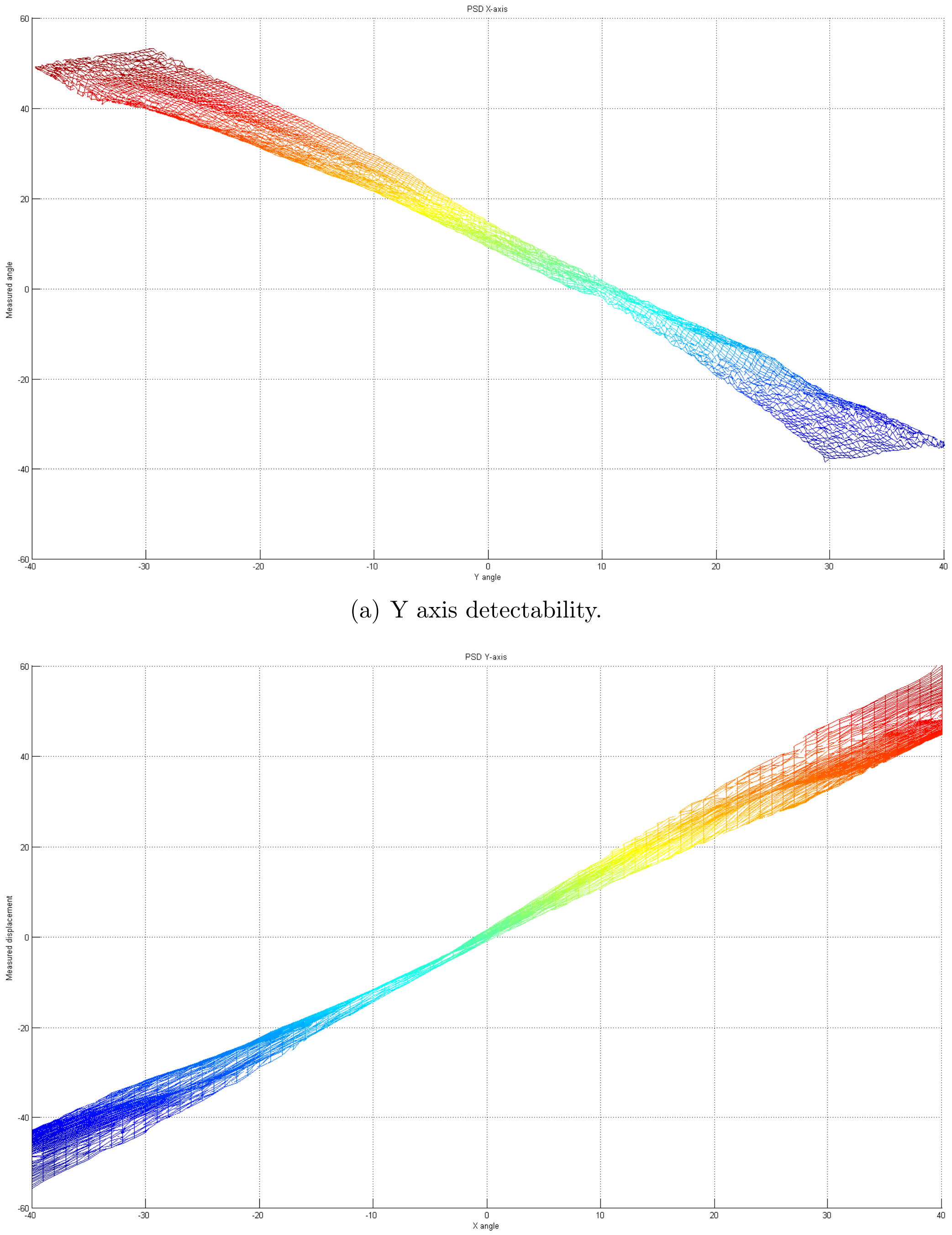
While these initial results are less than optimal, they can be expected to be greatly improved with non-reflective paint on the inside of the enclosure. Because of time constraints, this were not done in this work.

Selecting the measurements done with a 0° Y-angle results in a much higher accuracy, as seen in Figure 6.12(a). With a 9-degree polynomial curvefittig, the resulting accuracy is approximately ± 1° (depicted in Figure 6.12(b). The coefficients for the polynomial are listed in the matlab code in Appendix C.3.1.

The matlab code used to generate the curvefitting in Figure 6.10 and Figure 6.11 is found in Appendix C.3.2 and Appendix C.3.3.

Both a 200 μm and 300 μm pinhole was tried, which made a negligable effect on the measurements. As long as the spot size is large enough, increasing it is unnecessary, unless it is desirable to reduce gain and thereby reducing noise. This is further discussed in Section 4.3.3, Section 3.3.3 and Section 4.3.4, concluding that the noise is negligable. It is chosen to keep using the 200 μm pinhole.

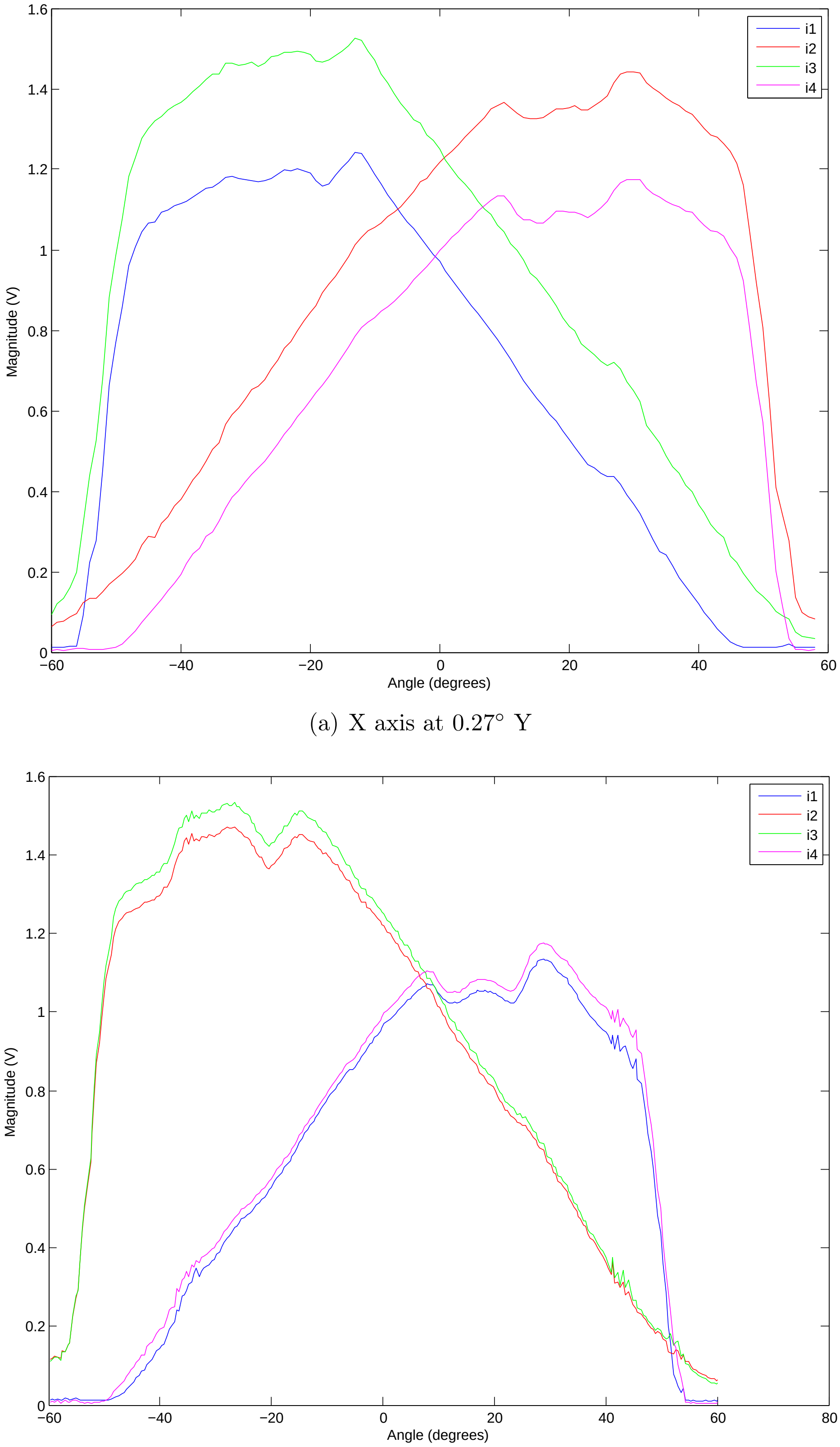
The plots in Figure 6.8 and Figure 6.9 are generated by the matlab code in Appendix C.2. Figure 6.12 was generated by the matlab code in Appendix C.3.1.



(a) Y axis detectability.

(b) X axis detectability.

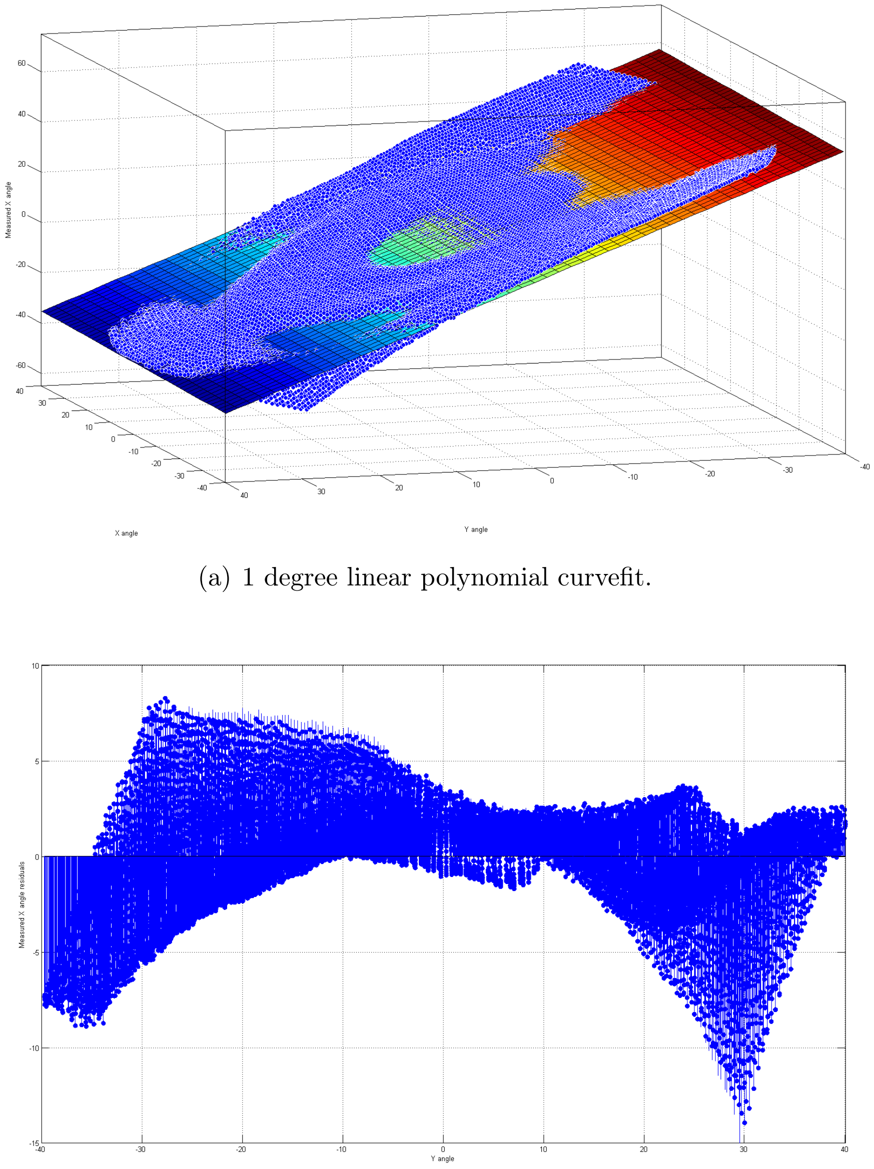
Figure 6.8: Calculation of X and Y axis detectability. The X and Y axes are exchanged, as the PSD is oriented with a 90° rotation on the PCB.



(a) X axis at 0.27° Y

(b) Y axis at 0° X

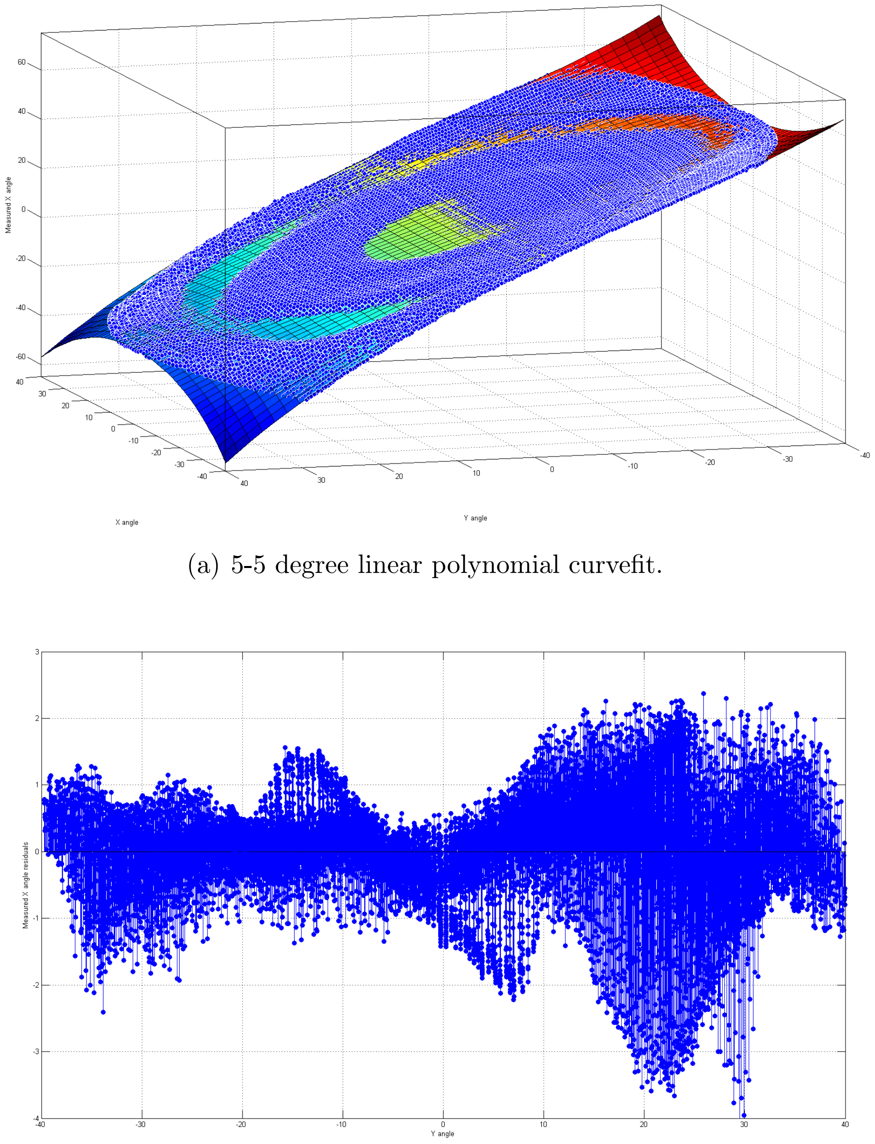
Figure 6.9: Relation of the current output signals from the PSD. The output signals from the transimpedance amplifier has been inverted to give the same curve as the actual current signals.



(a) 1 degree linear polynomial curvefit.

(b) Residuals of the 1 degree linear polynomial curvefit. The scale is -15 to 8 degrees.

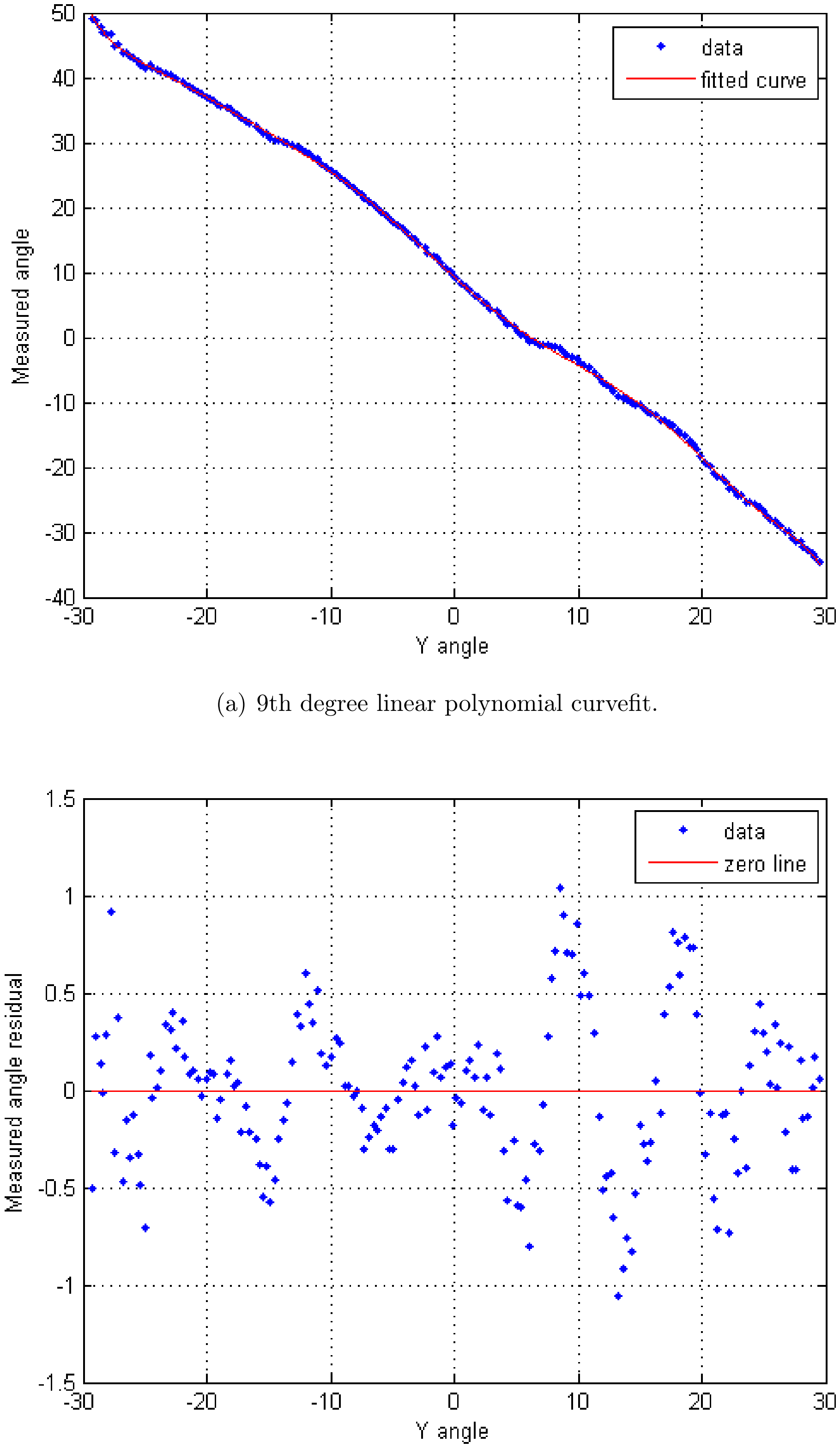
Figure 6.10: Curvefitting results for the X axis of the PSD using a 1 degree polynomial (a plane). A ± 8° accuracy is achieved through the majority of the fitted range with this fit.



(a) 5-5 degree linear polynomial curvefit.

(b) Residuals of the 5-5 degree linear polynomial curvefit. The scale is -4 to 3 degrees.

Figure 6.11: Curvefitting results for the X axis of the PSD using a 5-5 degree polynomial (a plane). A ± 2° accuracy is achieved through the majority of the fitted range with this fit.



(a) 9th degree linear polynomial curvefit.

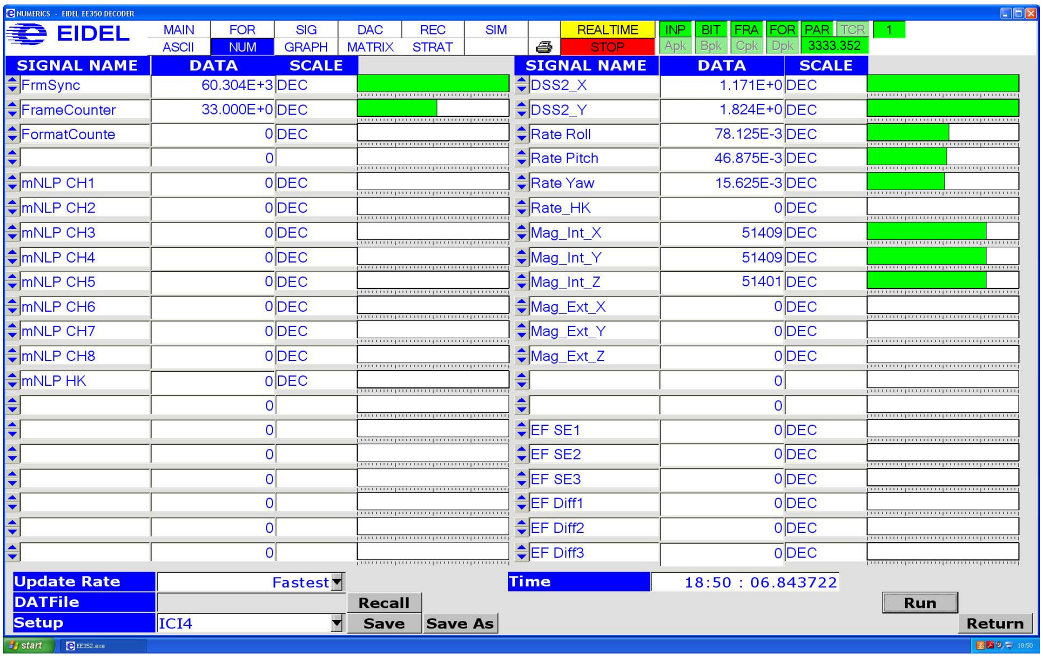
(b) Residuals of the 9th degree linear polynomial curvefit.

Figure 6.12: Curvefitting and residual plot of the PSD X-axis, measuring only the Y-axis of the sun sensor.

6.2 System integration

6.2.1 Functional system test

To ensure that the sensor data is correctly relayed to the rocket encoder, the system was tested against an encoder from FFI. The rocket encoder and the encoder interface is dicussed in Section 4.2.5. Eidel EE350 decoder software is used to display the recieved data. This verified that the data is sent correctly.



SIGNAL NAME	DATA	SCALE
FrmSync	60.304E+3	DEC
FrameCounter	33.000E+0	DEC
FormatCounte	0	DEC
	0	
mNLP CH1	0	DEC
mNLP CH2	0	DEC
mNLP CH3	0	DEC
mNLP CH4	0	DEC
mNLP CH5	0	DEC
mNLP CH6	0	DEC
mNLP CH7	0	DEC
mNLP CH8	0	DEC
mNLP HK	0	DEC
	0	
	0	
	0	
	0	
	0	
	0	
	0	

SIGNAL NAME	DATA	SCALE
DSS2_X	1.171E+0	DEC
DSS2_Y	1.824E+0	DEC
Rate Roll	78.125E-3	DEC
Rate Pitch	46.875E-3	DEC
Rate Yaw	15.625E-3	DEC
Rate_HK	0	DEC
Mag_Int_X	51409	DEC
Mag_Int_Y	51409	DEC
Mag_Int_Z	51401	DEC
Mag_Ext_X	0	DEC
Mag_Ext_Y	0	DEC
Mag_Ext_Z	0	DEC
	0	
	0	
EF SE1	0	DEC
EF SE2	0	DEC
EF SE3	0	DEC
EF Diff1	0	DEC
EF Diff2	0	DEC
EF Diff3	0	DEC

Update Rate: Fastest
DATFile:
Setup: ICI4
Time: 18:50 : 06.843722

Figure 6.13: Screenshot of the Eidel EE350 decoder software operating with the sun sensor and gyro. The internal magnetometer is not implemented and is replaced by a dummy counter for testing purposes.

The Internal and External Magnetometer data is a simple counter, just to confirm that the encoder interface is operating correctly. These sensors were not fully operational at this point.

6.2.2 ICI-4 first integration at Andøya Rocket Range

The first integration of the ICI-4 sounding rocket was done at Andøya Rocket Range (ARR) on the 23.04.2013.

This was an electrical compliance and physical placement test, meaning that the instruments were mounted in their respective placements and the interfaces were tested against the rocket encoder. As the interface to the encoder worked correctly on the first attempt, it was not much that had to be done.

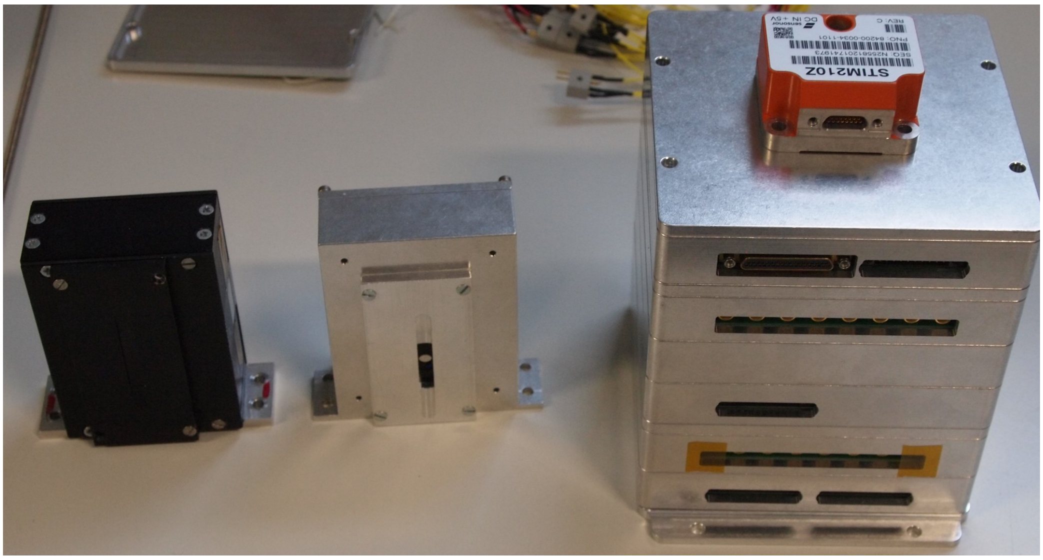


Figure 6.14: The complete UiO-stack with the STIM210 on top and the sun sensors, both the DSS and the DSS2.

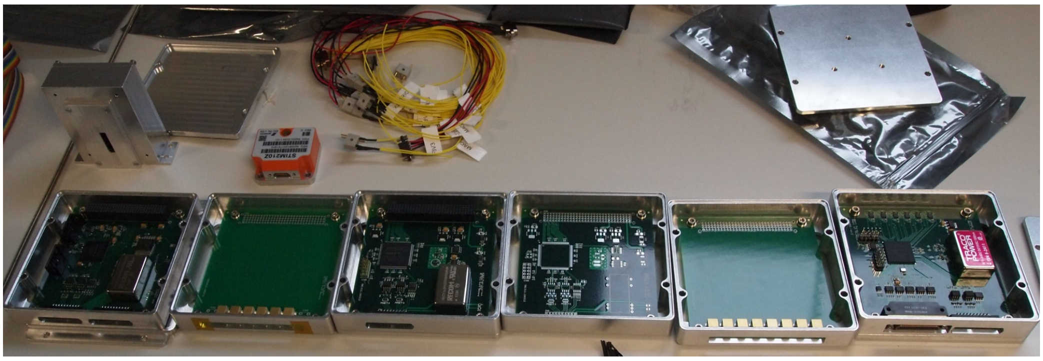


Figure 6.15: All the DAQ-boards for the UiO instruments, the DSS2 and STIM210 units.

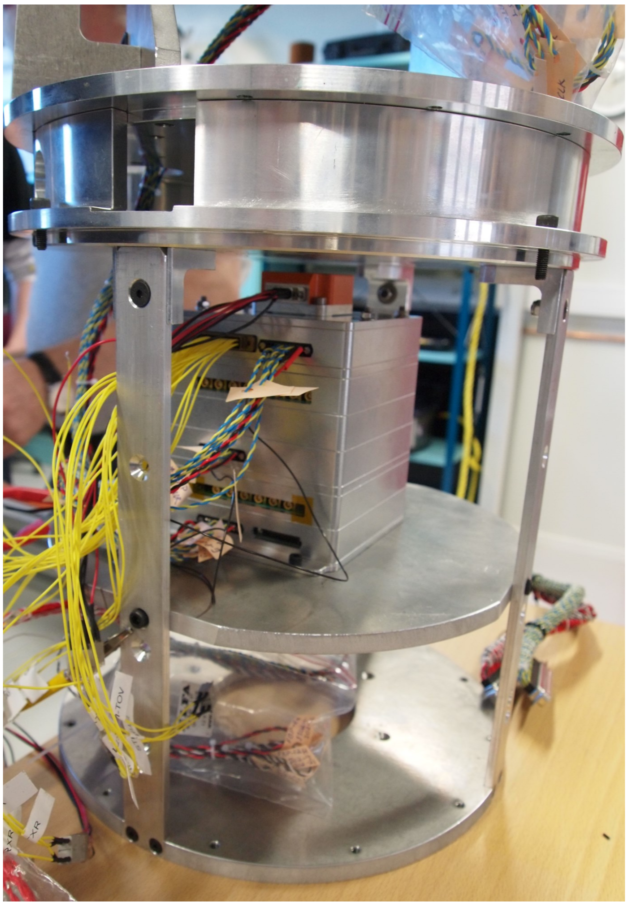


Figure 6.16: The UiO-stack mounted inside the payload "hotel" of the rocket. Notice the yellow harness coming from the DAQ-board of this thesis, used for testing.

6.3 Noise performance testing

Noise testing is performed with the whole system connected and sending data to an EIDEL rocket encoder which also provides the 28 V supply. All the measurements are taken with a Tektronix MSO4034B oscilloscope, using local ground as the reference level.

6.3.1 DC-DC noise

The Traco TEN5 DC-DC converter is specified with a maximal voltage ripple of 50 mV_{pk-pk}. It is seen in Figure 6.17 that the noise is well within specification, at a level of 15-20 mV when the system is fully operational.

Current drawn with one sun sensor and the STIM210 connected is 130 mA@28 V, which equates to ∼730 mA@5V. Taking into account the typical efficiency of the converter, 83%, this means that ∼600 mA is actually drawn from the converter.

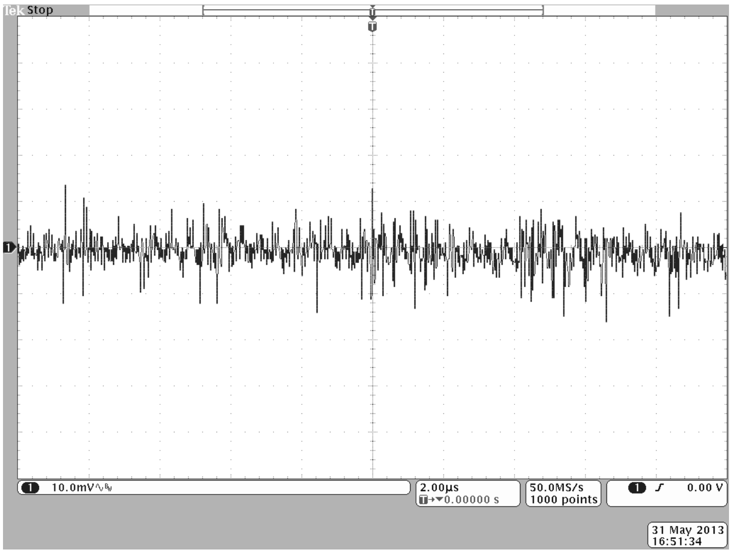


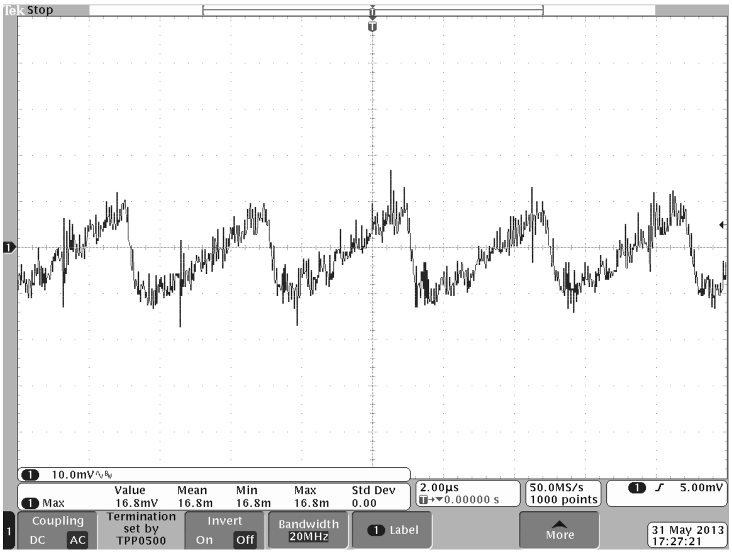
Figure 6.17: Output noise of the Traco TEN5 DC-DC converter. The ripple voltage has an amplitude of approximately 20 mV_{pk-pk}.

6.3.2 Inverter

The TI TPS60403 voltage inverter used to supply the negative rail, -5 V, for the amplifiers is the only other switching converter in the system. It is a source of switching noise and it may be advisable to implement a more aggressive filter for it in a later revision. The best alternative may be to find another solution.

As seen in Figure 6.18, the amplitude of the ripple is reasonably low, at 20 mV. With an ideal voltage inverter, the output would be -5.5 V, but the -5.4 V produced by this circuit is acceptable.

As the noise from the inverter is the dominating noise source on the DSS, it would be worthwile to apply more filtering to the output and input. During the measurements the 1 μF output and input capacitors was replaced by different sizes of tantalum capacitors (0.47 - 10 μF), which did not provide any substantial dampening of the ripple. A 4.7 μF ceramic capacitor was also tried, but it did not provide any noteworthy reduction of the ripple either. A more aggressive approach may need to be taken to eliminate the noise, e.g. an LC-filter on the output.



1 10.0mV BW

	Value	Mean	Min	Max	Std Dev
1 Max	16.8mV	16.8m	16.8m	16.8m	0.00

2.00µs
T→▼0.00000 s

50.0MS/s
1000 points

1 ⨍ 5.00mV

Coupling DC [✓ AC]

Termination set by TPP0500

Invert On [✓ Off]

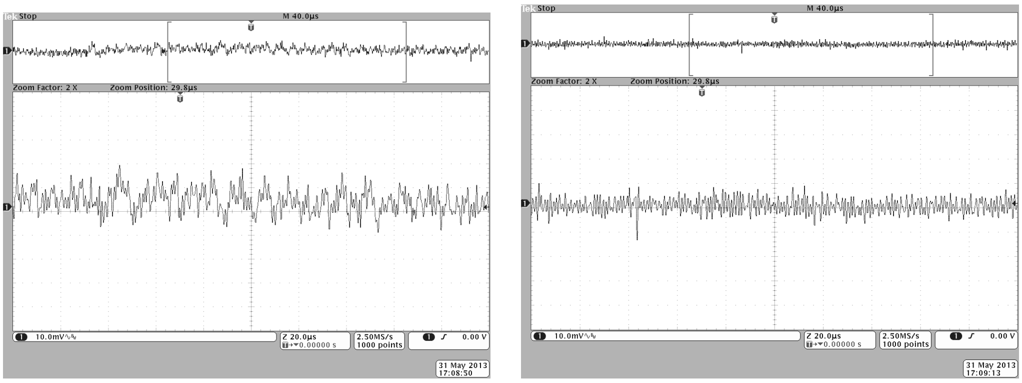
Bandwidth 20MHz

31 May 2013
17:27:21

Figure 6.18: Voltage ripple of the TI TPS60403 voltage inverter. The amplitude is ca 20 mV_{pk-pk} with a frequency of approx. 250 kHz.

6.3.3 LDO noise suppression

As suspected, the TI TPS717xx LDO regulators attenuate the noise from the DC-DC (see Figure 6.19). The ripple voltage level on the input is low, but the attenuation is obvious when observing the figure. Since the ripple voltage levels are so low, the measurement equipment is an error source when measuring the ripple voltage.



(a) 5V input

(b) LDO 3.3V output

Figure 6.19: Input voltage and output of LDO on the DSS2. It is seen that the ∼20 mV_{pk-pk} noise on the 5 V rail is attenuated to about 8 mV_{pk-pk}.

All the TPS717xx LDOs have an almost identical, very low, noise level of ∼10 mV.

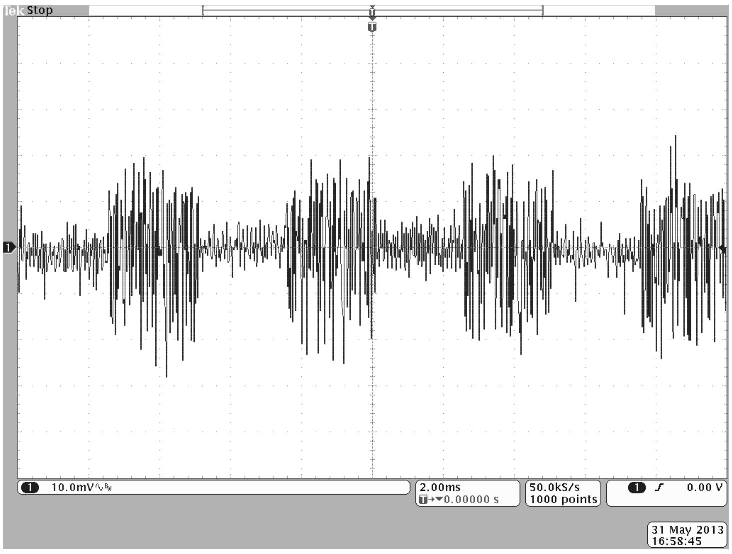


Figure 6.20: The ripple voltage in the LM1117 regulator. The ripple of the "pulse trains" are approximately 40 mV_{pk-pk}. It is observed that the noise is active with a frequency of 1 kHz.

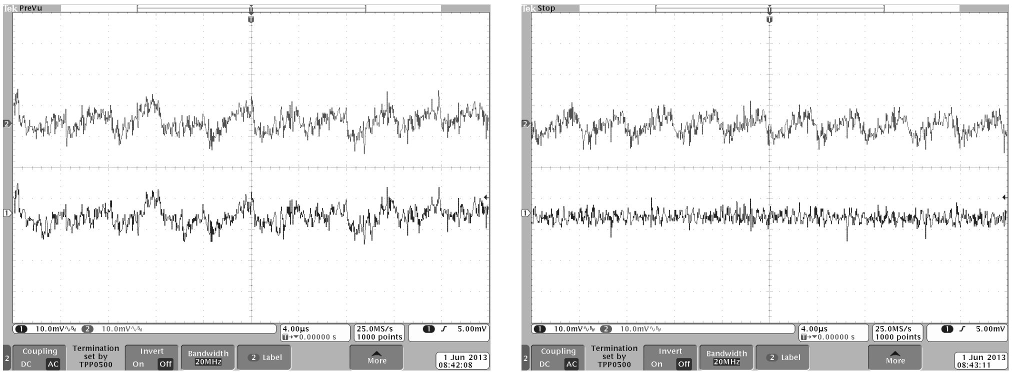
The LM1117 regulators are more noisy, with a ripple voltage of 40 mV as seen in Figure 6.20. As there is no specified ripple voltage for this device, this is probably to be expected. More aggressive noise filtering should be implemented in a later version or a suitable alternative could be found, if the common power supply for the UiO-stack is not yet implemented. If the magnetometers on the DAQ-board and the external board prove to be sensitive to supply noise, it may be necessary to use a more bypass rigorous strategy for this voltage net.

6.3.4 Noise in the photodetector.

Background noise

The measurements in Figure 6.21 are taken with the PSD covered up, so that it measures no light. From the figure, it is evident that the power supply noise carries directly over to the signal noise due to common ground. The dominating noise signal has a frequency of 250 kHz, which is the nominal switching frequency of the voltage inverter.

With the RC-filter inserted in the signal chain, the ripple voltage is attenuated from approximately 12 mV_{pk-pk} to 5 mV_{pk-pk}. When the system is taking real world measurements, the effect of this noise dampening might have a bigger impact on the output signal as other noise sources may also be present. It is probably wise to keep the filter to eliminate all the noise outside of the measurement bandwith.



(a) Without RC-filter

(b) With RC-filter

Figure 6.21: Quiescent noise on the ADC input pin, with and without RC-filter, related to the 5 V supply voltage. It is observed that the power supply noise has a direct carry over to the output signal. The ripple voltage is reduced by approximately 60%.

Consideration regarding the noise in the DSS

It is clear that the inverter noise is much more dominant than the noise sources of the photodetector circuit discussed in Section 3.3.3.

Assuming the noise were to affect the power supply voltages for both the amplifier and the ADC in the same way, the noise signal could be (at least partly) cancelled out. The fact that the ADC uses the V_{DD} as a reference could have been an advantage, as the fluctuations of the power supply voltage will affect the reference in the same way as the amplifier and could thereby have cancelled out the noise. This assumption might be correct if a Flash ADC were used, as this type would almost instantly convert the signal. With an SAR ADC, the conversion does not happen instantly as the converter will use some time for the internal comparing process. The reference would then still be oscillating, while the input signal is frozen in the SHA (Sample and Hold Amplifier). This could then actually lead to even more noise, with the possibility of the noise being cancelled out or being amplified. If, however, the SHA is affected by the noise in the same way as the internal reference, the noise seen at the input could remain unchanged through the circuit. As the actual implementation of the ADC is not known, this behaviour is impossible to predict with any precision.

On the other hand, if all four measured signals are affected in the same magnitude, then the algorithm could probably be modified to cancel out the noise. It may be more expedient to leave this as it is, and try to eliminate the noise from the inverter, since it is most likely the culprit.

As the background noise in the photodetector looks to be manageable with extra power supply filtering, it should not be necessary with more signal filtering than what is implemented. Tuning of the feedback cap in the transimpedance amplifier may improve cutoff attenuation if so desired. Even if the noise should get high enough to create a whole LSB of error, it would still leave 15 usable bits and the sensors resolution should be fully utilized.

The OPA129 has a very good PSRR at low frequencies, but the 250 kHz inverter noise will have a large effect as the PSRR for the negative rail is relatively weak. At 250 kHz, the +PSRR is approximately 50 dB and the -PSRR is 10 dB. Switching noise on the power supply nets, especially the -5 V rail, will have a prominent effect on the output of the opamp because of this. The results found in Figure 6.21 suggest that this assumption is correct.

It is evident that the inverter is causing the larges noise source and should be filtered better. Having tried different capacitors with no luck, it seems that it may be necessary to make an LC-filter or find another alternative to the inverter for a later version of the instrument.

Chapter 7

Comparing the new vs old systems

Comparing the newly developed measurement system to the old is done to give a look at the improvements in the new design. An obvious difference between the two systems is the size and weight reduction. The connection to the rocket encoder is also simplified greatly, with all the sensors transmitting their data through the DAQ-board instead of having separate connections.

7.1 Sun Sensor

Although the physical size of the sun sensor is not much reduced, the PCB layout is easily restructured for future reductions. Design changes are suggested in Section 8.3.1 (Future Work), which will make the instrument smaller and better. Suggestions are also given for changing the shape of the sensor, if e.g. a narrower, but higher, sensor is desired.

The results in Section 6.1 indicate an accuracy of ± 1° when only using the one axis. However, limitations in the measurement setup, particularly the light source, may limit the maximal accuracy of these measurements. With a better setup the results will most likely be better. Internal reflections, as discussed in Section 6.1.3 in the sun sensor enclosure may have contributed to this measurement error. The old DSS had an estimated accuracy in the order of a few degrees. Limitations in the test setup contributed on these measurements as well.

The DSS weighs in at 190 grams while the DSS2 weighs 168 grams. Size reduction of the DSS2 PCB, and then also the enclosure, is planned and will increase this ratio.

Assembling the new sensor is simple, as none of the parts are particularly hard to mount. Mounting the PCB in the enclosure is straight forward and fastening the pinhole is done via four screws and a metal plate on the front. When assembling the DSS1, there are many parts to put together for the enclosure as well. Many parts means that many small misalignments are possible, and this affects the calibration of the unit. It also increases the time required for to assemble the instrument. Combining that with the assembly of three PCBs makes the difference very notable.

The physical measurements of the boxes are very similar, as the focus of this prototype was to accomplish correct function before reducing size as much as possible.

The cost of the electric components of the old DSS was ca 2300 NOK and on the new DSS2 this has been reduced to about 1300 NOK. Making the enclosure is probably a bit cheaper as well, because of the reduction of parts.

The DSS1 consumes 65 mA@28 V, while the DSS2 is considerably lower at approx. 10 mA@28 V (through the SRADS2).

7.2 Gyro Module

The gyro is an extreme example of the space savings with the new system, where the new module from Sensonor weighs in at 52 grams and measures 39x45x22 mm. Comparatively, the IRU developed by Bekkeng (2007) weighs in at 600 grams and measures 110x110x90 mm. At approx. 3.5% of the volume and 8.7% of the weight, the Sensonor module has a clear advantage.

Mechanical misalignment errors are much better controlled as all the 3 axes of the STIM210 are made on a single chip, compared to the 3 separate PCBs of the old IRU.

Calibration of the Sensonor module is not necessary, as they are calibrated from the factory. Temperature compensation is also integrated, so any compensation for this is not required in post processing. In comparison, the IRU is temperature compensated in post processing, thus requiring three more data words to be sent to the encoder.

The power consumption of the old IRU was 85 mA@28 V, while the Sensonor unit has been measured to 50 mA. It is specified to a max consumption of 1.5 W, meaning that it could reach 66 mA@28 V when connected through the Traco DC-DC converter.

The cost of the IRU from the original SRADS measurement system has the advantage of lower component cost. Counting in the time used to manufacture the instrument and its enclosure and calibration may even out the difference between the two, since the Sensonor unit is ready to play from the box.

7.3 Magnetometer

The external magnetometer developed by Roar Danielsen and Halvor Strøm is mounted on a boom outside the rocket. Eddy currents in the payload skin can be induced by the charge difference in the ram and wake regions around the payload.

Another possible explanation is that contamination of the payload skin surface could lead to a difference in the work function on different sides of the rocket, which may affect excitation of electrons from the rocket skin. This could in turn lead to a charge difference between the sides of the rocket when different areas enter and exit the ram and wake regions.

Charges in the payload skin can create an internal magnetic field, meaning that the measurements of the internal magnetometer are basically useless for attitude determination.

As there is a magnetometer implemented on the DAQ-board as well, both internal and external magnetic fields are measured. This makes it possible to study the effect of the eddy currents in the payload skin, as the external magnetometer could be used as a reference.

Chapter 8

Conclusion

8.1 Summary of the present work

Through this thesis, a new digital sun sensor has been developed. With this, an updated measurement system for usage with the SRADS software (Bekkeng, 2007) has been put together. It comprises the new sun sensor, a gyroscope module from Sensonor and an external magnetometer card made by the ELAB at UiO. The performance of the sun sensor has been verified and a system integration was performed to make sure that all the external sensors are interfacing correctly with the DAQ.

As the system is expected to recieve design updates in the future, the firmware and hardware has been designed with maintenability in mind. Changes and addititons to the design should hopefully require little effort because of this.

Some unforseen challenges were met during the development process. When the sun sensor didn't work correctly in the final characterization tests, it was necessary to debug the system in the last period of the thesis. A misunderstanding of the documentation for the FPGA made the first DAQ revision inoperable, but this was easily fixed. This is covered in Appendix A.3.1.

8.2 Conclusions

The resulting system is a working prototype and the communication between the sensors and the DAQ board has been verified. As all of the sensors use the same interface, integration with the external magnetometer should be relatively effortless.

The initial tested accuracy of the sun sensor is approximately ± 1°, but several factors have been found that may limit this measured performance.

Internal reflections in the enclosure may contribute to a diplacement of the measured light spot position. This may be improved by painting the insides of the enclosure with non-reflective paint. The measurement setup itself is likely the largest limiting factor, and improving this may well improve the measured performance greatly.

Further integration of power supply and size reductions of the sun sensor is easily implemented in future revisions of the system.

Two small design updates to the digital sun sensor are required before launch, discussed in Section 8.3.1.

8.3 Future work and design changes

The sun sensor model and the corresponding mathematics in the Kalman filter estimator used in the attitude determination estimator by Bekkeng (2007) must be modified.

As this is a prototype design, the focus has been to engineer a well functioning system, rather than reducing the size and power consumption. Nevertheless, several suggestions for design changes are made in this section on how to improve these aspects at a later time, as a smaller and less power hungry system should always be strived for. Recommendations for functional improvements are also given.

8.3.1 DSS2

Two design changes are needed before flight:

To obtain a ± 60° field of view, the distance from the PCB to the pinhole must be decreased. As of now, the distance between the PCB and the pin-hole is 3.4 mm, instead of the required 2.6 mm.

The insides of the enclosure should be painted with a non-reflective paint. This is expected to improve the accuracy of the measurements, which are likely plagued by internal reflections.

The operational amplifier operates in saturation, as discussed in Section 6.1. For the first flight it is recommended to exchange the voltage reference for a 3.3 V version, because of time constraints. A redesign should be done before the next rocket campaign to exchange the opamp for a rail-to-rail single 5 V supply model, to eliminate the inverter and thereby the most dominating noise source. This would also simplify the routing by eliminating one of the power planes. The OPA140 would be a good choice. It has better voltage noise and similar current noise characteristics, but the bias and offset currents are higher, although acceptably low. The benefit of removing the inverter noise far outweighs the larger bias and offset.

For the first flight, it is sufficient to exchange the 4.096 V voltage reference, the ADR3440, for the 3.3 V variant, the ADR3433. These models are directly exchangable, but gain will also have to be reduced by 20% ($3.3\ V/4.096\ V = 0.806$). This means that a feedback resistor, R_f, of about 127 kΩ should be used. A feedback capacitor of 22 pF should be sufficient to stabilize the circuit.

Changing the voltage reference will make the system operationable, but it will also effectively lower the signal to noise ratio, making the system more susceptible to noise. Exchanging the opamp for the OPA140 in single 5 V supply mode means that the inverter can be removed. This will lower the noise significantly, as the noise from this circuit dominates the total noise. The pinout is almost identical, so it should be an easy modification.

Functional improvements

- The "shelf" in the enclosure for the PCB should be raised by 1.8 mm, as discussed in Section 6.1.3. In the current position, the PSD is a bit too far from the pin-hole. This reduces the field of view to ∼± 45°.

- As discussed in Section 6.1.3, a wave-like pattern is formed in the measurement plane of the PSD. The low reverse bias may contribute to this and should be further researched to eliminate it as a source of error.

- The biased transimpedance amplifier stage should be replaced by a transimpedance amplifier with a differential amplifier, if a faster response time from the PSD is desirable. This will improve two things: The output signal will be rising with intensity instead of falling and the reverse bias of the PSD will be increased. As the bias voltage on the amplifiers positive terminal will effectively forward bias the PSD, a very low reverse bias of 1.4 V is left.

- A more aggressive filtering should be adopted for the inverter, unless it is removed in a later version. As seen in Section 6.3.2, its switching noise is present on the power planes and should be reduced.

- The power supply for the ADC should be changed to 4.096 V to increase the resolution in the voltage range actually used (0-4.096 V), unless the opamp has been exchange for a rail-to-rail variant. As the voltage value corresponding to one LSB with 4.096 V supply would be 62.5 μV and only the 13 MSBs are used, noise is considered no problem in this configuration. To realize this, it may be necessary to increase the PCB stackup to six layers.

- The accuracy of the PSD could be further increased by using a more sophisticated algorithm. By applying the principles from the work of Cui and Soh (2010) it may be possible to decrease distortion substantially. It may be possible to reduce the effects of the reflected light from the pin-hole shape as well, but this must be researched further before a conclusion can be drawn.
- While a lower-voltage ADC may not be a better solution, as some of the possible dynamic range would be lost, it could make for an easier design. Driving the ADC with a 3.3 V supply would make the measuring range 0-3.3 V. Substituting the voltage reference and modifying gain would make it easier to integrate with a smaller CPLD, as they lack the clamping diodes.
- For a possible reduction in line capacitance between the sensor and amplifiers, if it is desirable, implementing guard rings on the inputs of the amplifier should be considered. It was opted out in this version, as it was deemed an unneccessary complication of the layout.
- A six layer PCB stackup should be considered with the stack up being: Analog side, ground, 5 V and 3.3 V power plane, -5 V and 1.8 V power plane, ground, digital side. This will reduce the risk of interrupted current paths caused by signal traces crossing the splits in the power planes. It will also shield the power planes from the electronic components, possibly reducing noise further. If the inverter is still present in the design, it should be placed in a "dummy" ground plane, tightly coupled to the main ground planes to reduce noise.
- The serial clock (SCLK) for the ADCs on the DSS2 PCB should be routed as two separate wires, one for each pair, to reduce the possible emitted noise from the long clock route. Care must be taken to ensure that the outputs are synchronous, but this should not be a problem on the CPLD. This could also make routing easier in a future redesign. It could also reduce capacitance in the wire.

Size and power consumption reduction

- The 20 MHz oscillator used on the DSS2 could be exchanged for a 10 MHz unit, which may reduce noise and power consumption a little bit. It should be thoroughly tested if the firmware design still functions correctly with this change. As the clock divider would be reduced, a small reduction in logic cost would be expected as well.

- All of the 0-ohm resistors connecting to the ADCs can be removed. If another ADC is substituted, this may be irrelevant.

- As the final logic design ended up using 382 logic elements, the CPLD with 1270 available elements is not neccessary. This size was chosen at the beginning of the project as more of the functionality was planned to be implemented on it. The 570 element variant in MBGA100-package is suggested for a later version, as it measures 36 mm^2 compared to the TQFP144 with 484 mm^2. It will also draw less current.

  Another option might be to use the FBGA256 1270-element version as this still saves a bit of space, measuring 289 mm^2.

  However, care must be taken if the ADCs are to be interfaced to the 570 element Max V, as it doesn't have a built in IO clamp diode. Adding external clamp diodes shouldn't likely cause any problems though and they are only required for the input lines to the CPLD.

- If the same CPLD is to be used regardless, the footprint has to be changed. The footprint used turned out to have a skew in the position of the pins on one side. This made for a very unforgiving assembly of the PCB, with many solder bridges that had to be reworked, but it was not discovered until the last boards were mounted with the Onyx 32 robot at the electronics laboratory.

- Another space-saving change could be to move the transimpedance amplifier circuits to the bottom side of the card, placing them close to the ADC's. If the CPLD has been changed it should be feasible, although tight. As the digital and analog components would not be shielded by a ground plane any more, it may not be wise to do this with regards to noise. The output currents of the PSD would also have to pass through the vias before being amplified. This might add some noise and possible parasitic loading of the sensor, but this has not been estimated.

  Depending on the CPLD chosen, and the layout of bypass capacitors underneath it, it could also be possible to move the amplifiers so that they are located beside the sensor. Visual estimation suggest that this could save up to 30% in itself, but most of the other recommended changes must be implemented to make room for this. Increasing the length of the board would obviously be an easy way to enable these changes, if that is acceptable.

- The voltage reference, used for biasing the opamps, and the voltage inverter, used for powering them, could be moved inwards on the board, which might leave up to potential 10 mm free space if the previous recommendation has been followed.

- The RS-422 drivers should also be exchanged for LVDS-drivers, if these are found reliable, as they are smaller of size and less power-hungry. They are faster as well, but the capabilites of either standard is much higher than the requirements of this system as of now. The second RS-422 driver for recieving an extra control signal can most likely be removed. A control signal is needed to signal when the sensor should collect data, but this might not need differential driving as it is used as a simple flag.

- Most of the resistors and capacitors on the board are of the 0603 size to allow easy debugging/modification, but in a finished instrument this is not necessary. If they were to be changed to 0402 types this could help facilitate the other proposed changes.

- A smaller JTAG-connector could also be used, as the currently used pinrow connector is larger than neccessary. The only actual advantage with the used connector is that it is the same size as the header on the JTAG programmer from Altera.

- The proposed changes may be able to reduce size by about $30 - 50\%$, as the space savings that can be had by replacing the components is quite significant. There is also a lot of free space on the board on both sides.

8.3.2 DAQ

- The STIM210 gyro module used was preprogrammed to expect a trigger input, but the voltage level for logic high is minimum 3.7 V. Seeing as the FPGA is capable of delivering maximum 3.3 V and that the clamp diodes will eliminate any pullup voltage it is not possible to drive this input. A simple level converter should eliminate this issue and it shouldn't take much effort to implement. The gyro module has been reprogrammed to not expect a trigger input and will need to be reprogrammed again. This is, fortunately, very easy to do.

- If the RS-422 drivers are replaced by LVDS on the sun sensor it goes without saying that they should also be replaced accordingly, and the driver for the extra control signal removed if done so on the sun sensor.

Bibliography

Analog Devices (2011). AD7680 datasheet.

Bak, T. (1999). *Spacecraft Attitude Determination: A Magnetometer Approach.* Ph. D. thesis, Aalborg University.

Bang, D. M. (2013). A sun sensor for the cubeSTAR nano sattelite.

Bekkeng, J. K. (2007). *Prototype Development of a Low-Cost Sounding Rocket Attitude Determination System and an Electric field Instrument.* University of Oslo, Faculty of Mathematics and Natural Sciences.

Bekkeng, T. A. (2009). Prototype development of a multi-needle langmuir probe system.

Bekkeng, T. A. (2013). personal communication.

Bogatin, E. (2004). *Signal integrity: simplified.* Prentice Hall Professional.

Coriolis, G. G. (1835). *Mémoire sur les équations du mouvement relatif des systèmes de corps.* Bachelier.

Cui, S. and Y. C. Soh (2010). Linearity indices and linearity improvement of 2-D tetralateral position-sensitive detector. *Electron Devices, IEEE Transactions on 57*(9), 2310–2316.

Edmund Optics (2013). Optics and optical instruments catalogue.

Finlay, C., S. Maus, C. Beggan, T. Bondar, A. Chambodut, T. Chernova, A. Chulliat, V. Golovkov, B. Hamilton, M. Hamoudi, et al. (2010). International geomagnetic reference field: the eleventh generation. *Geophysical Journal International 183*(3), 1216–1230.

Fraden, J. (2004). *Handbook of Modern Sensors: Physics, Designs, and Applications* (third ed.). Springer.

Grødal, A. (1997). *Elektromagnetisk kompatibilitet for konstruktører*. Tapir forlag.

Hamamatsu, P. (2002). PSD (position sensitive detector). *Hamamatsu, Japan*.

Hamamatsu, P. (2005). Photodiode technical information. *Hamamatsu, Japan*.

Hamamatsu, P. (2007). Two-dimensional PSD S5991.

Honeywell. AN211 Applications of magnetic position sensors. Last checked 07.05.2013.

Johnson, J. B. (1928). Thermal agitation of electricity in conductors. *Physical Review 32*(1), 97.

Keithley, J. (2004). *Low Level Measurements Handbook: Precision DC Current, Voltage, and Resistance Measurements*. Keithley Instruments.

Kvisterøy, T., N. Hedenstierna, S. Habibi, B. Nyrud, and E. Wolla (2004). Realization of the sar 10 rate gyro based on a proven manufacturing platform.

Liebe, C. C., K. Gromov, and D. M. Meller (2004). Toward a stellar gyroscope for spacecraft attitude determination. *Journal of Guidance, Control, and Dynamics 27*(1), 91–99.

Mancini, R. (2002). *Op amps for everyone*. Texas Instruments.

McGuire, T. and R. Potter (1975). Anisotropic magnetoresistance in ferromagnetic 3d alloys. *Magnetics, IEEE Transactions on 11*(4), 1018–1038.

Motchenbacher, C. and J. Connelly (1993). *Low noise electronic system design*. J. Wiley & Sons.

Nyquist, H. (1928). Thermal agitation of electric charge in conductors. *Physical review 32*(1), 110–113.

Ott, H. W. (2011). *Electromagnetic compatibility engineering*. Wiley.

Psiaki, M. L. (2006). Batch algorithm for global-positioning-system attitude determination and integer ambiguity resolution. *Journal of guidance, control, and dynamics 29*(5), 1070–1079.

Schottky, W. (1918). Über spontane stromschwankungen in verschiedenen elektrizitätsleitern. *Annalen der Physik 362*(23), 541–567.

Sensonor (2010). SAR150 rate gyro datasheet.

Sensonor (2013). STIM210 multi-axis gyro module datasheet.

Sollien, M. (2006). Prototyputvikling av digital solsensor for sonderaketter.

Texas Instruments (2009). TPS717xx datasheet.

Traco (2012). TEN5 datasheet.

Wang, W. and I. J. Busch-Vishniac (1989). The linearity and sensitivity of lateral effect position sensitive devices-an improved geometry. *Electron Devices, IEEE Transactions on 36*(11), 2475–2480.

Wertz, J. R. (1978). *Spacecraft attitude determination and control*, Volume 73. Kluwer Academic Pub.

Westerman, D. (2007). Understand and apply the transimpedance amplifier. *National Semiconductor Corp.*

Zumbahlen, H. (2007). *Basic Linear Design.* Analog Devices.

Appendix A

Schematics and PCB Layout

A.1 DAQ

Listing: Parts list for the DAQ board

```
______________________________________________________________________________
                                 Parts List

                     CADSTAR Design Editor Version 13.0.0.3
______________________________________________________________________________
Part Name                  Description                        Qty.    Comps.
_________                  ___________                        ____    ______
ALTERA/EP4CE15/FBGA-256    Cyclone IV FPGA Family             1       U1
ALTERA/EPCS4/SOIC-08                                          1       U3
CAP/100NF/0402R            10% 16V 0402 X7R 7"REEL            5       C6
                                                                      C25
                                                                      C35
                                                                      C44-45
CAP/100NF/0603R            10% 16V 0603 X7R                   8       C5
                                                                      C9
                                                                      C12
                                                                      C15
                                                                      C18
                                                                      C21
                                                                      C24
                                                                      C36
CAP/100NF/1206R            10% 50V 1206 X7R                   3       C2-4
CAP/10NF/0603R             10% 50V 0603 X7R                   2       C28
                                                                      C31
CAP/1U0F/0603R             10% 25V 0603 X5R                   4       C26-27
                                                                      C29-30
CAP/220N/0402R             10% 16V 0402 X5R                   6       C8
                                                                      C11
                                                                      C14
                                                                      C17
                                                                      C20
                                                                      C23
CAP/4U7F/0603R            AVX 10% 10V 0603 X5R                6       C7
                                                                      C10
                                                                      C13
                                                                      C16
                                                                      C19
```

			C22
CAP/BYPASS/0402R	10% 16V 0603 X7R	32	CB1−28
			CB30−33
CAP/BYPASS/1206R	10% 50V 1206 X7R	5	CB35−39
CON/Datamate M80 2x10 R	Datamate M80 2x10 pins Right A	1	CN1
CON/MDM−37PCBRP/HOR	MICRO−D Connector 37pos Right	1	CN2
CON/PR5X2	5X2 SCOTT ELEC. PINROW	2	CN3−4
DIO/BAS16/SMD	SWITCH DIODE 75V/0.1A. SOT23	4	D1−4
LED/19−21SDRC/SMD	SMD LED RED	1	D5
LED/19−21SYGC/SMD	SMD LED GREEN	1	D6
LOGIC/SN65LVDS179/SMD	LVDS tranciever, single	1	X6
MAG/HMC5983/QFN16	3−axis magnetometer I2C/SPI	6	U4−9
MAXIM/MAX490/SMD	FULL DUPLEX TRANCIEVER	6	X1
			X11−15
OPTO/HCPL0631/SMD	HIGH SPEED OPTOCOUPLER	2	OC1−2
POW/LM1117I	3.3V POS. VOLTAGE REG.	1	X4
POW/TEN5−2411/TEN	DC/DC CONVERTER	1	X5
POW/TPS717XX/SMD	LDO REGULATOR	2	X2−3
RES/0R00/0402R	RES KOA 0402 1% 63mW MINIREEL	16	R1
** 0402 RES **			R22
			R25−26
			R30−31
			R33−34
			R38
			R43−46
			R49−51
RES/0R00/1206R	RESISTOR KOA 1206 1% 0.25W	1	R99
RES/100R/0402R	RES KOA 0402 1% 63mW MINIREEL	2	R64−65
** 0402 RES **			
RES/10K0/0402R	RES KOA 0402 1% 63mW MINIREEL	6	R2
** 0402 RES **			R17−20
			R37
RES/120R/0402R	RES KOA 0402 1% 63mW MINIREEL	9	R3
** 0402 RES **			R27−28
			R32
			R35−36
			R40
			R47−48
RES/1K00/0402R	RES KOA 0402 1% 63mW MINIREEL	5	R12−16
** 0402 RES **			
RES/220R/0402R	RES KOA 0402 1% 63mW MINIREEL	8	R4−11
** 0402 RES **			
RES/24R0/0402R	RES KOA 0402 1% 63mW MINIREEL	1	R21
RES/470R/0603R	RESISTOR KOA 0603 1% 0.1W	2	R66−67
TANT/10UF/16V/SMD−A	KEMET T491 SERIES 20%	4	C33−34
			C37
			C46
TANT/10UF/35V/SMD	TANTAL ELECTROLYTIC CAP	1	C1
TANT/1UF/20V/SMD	KEMET T491 TANTAL ELECTROLYTIC	6	C38−43
XTAL/ASEMB20MHZ	ABRACON MEMS CLOCK OSCILLATOR	1	U2

End of report

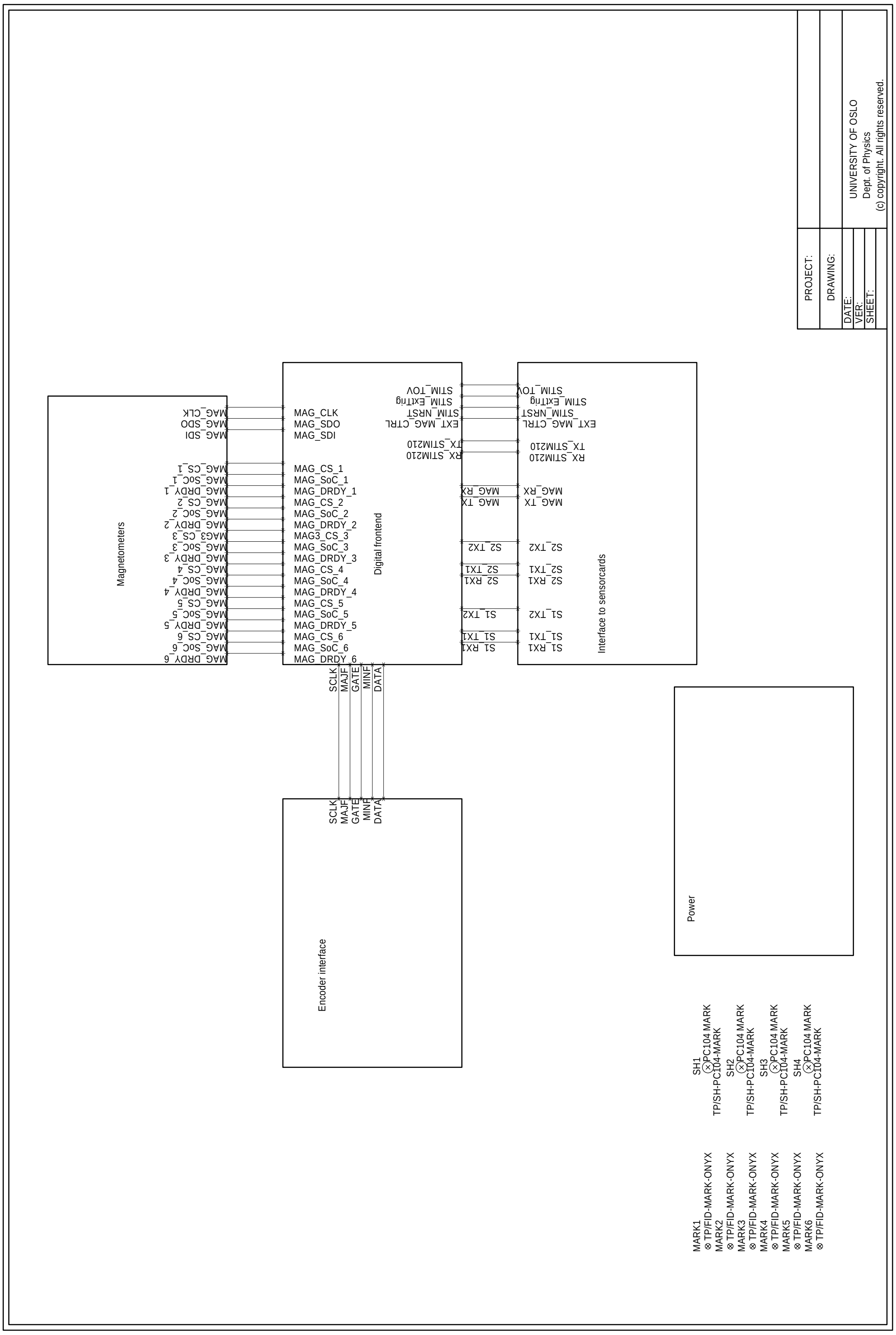


Figure A.1: DAQ top level schematic.

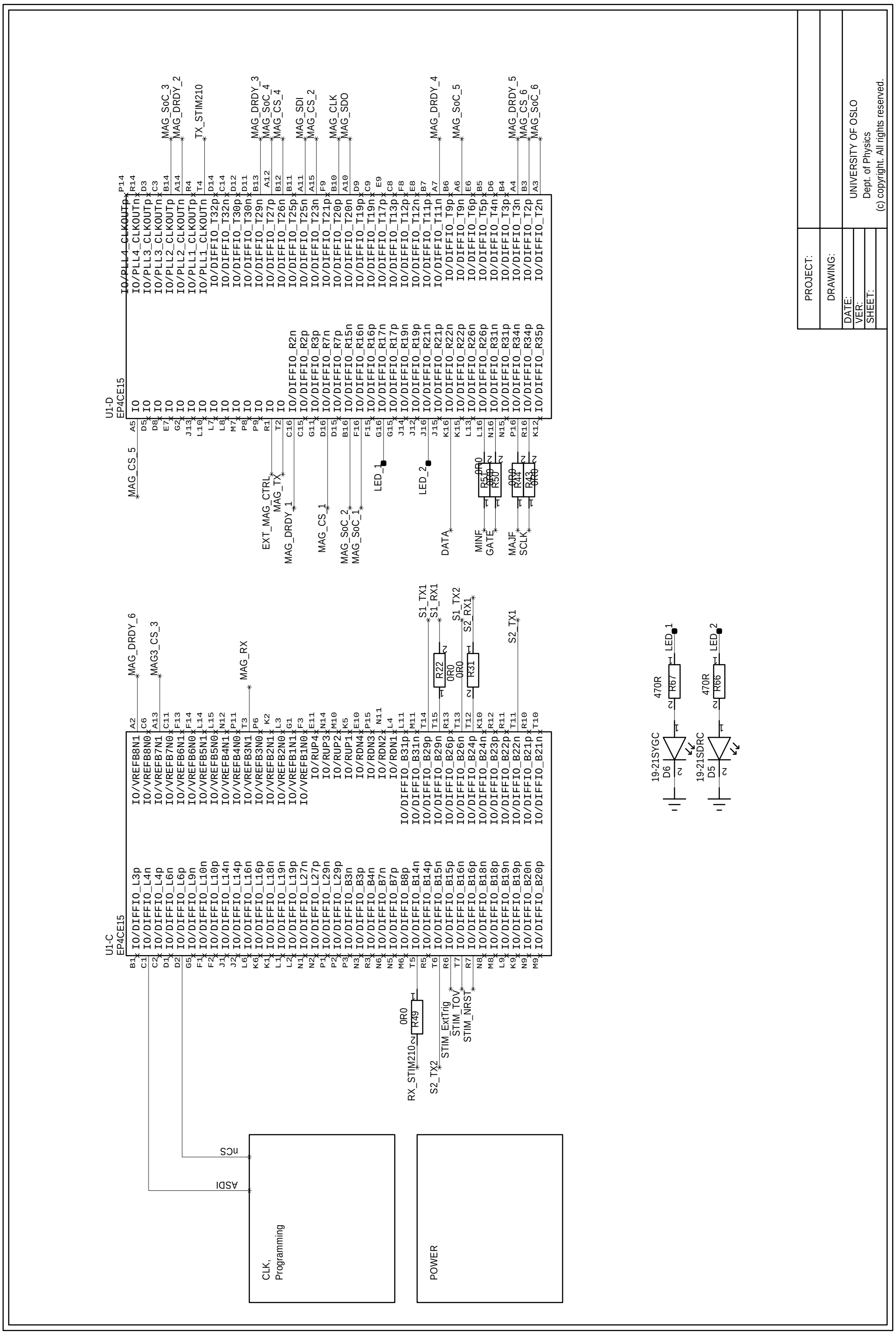


Figure A.2: DAQ digital front-end.

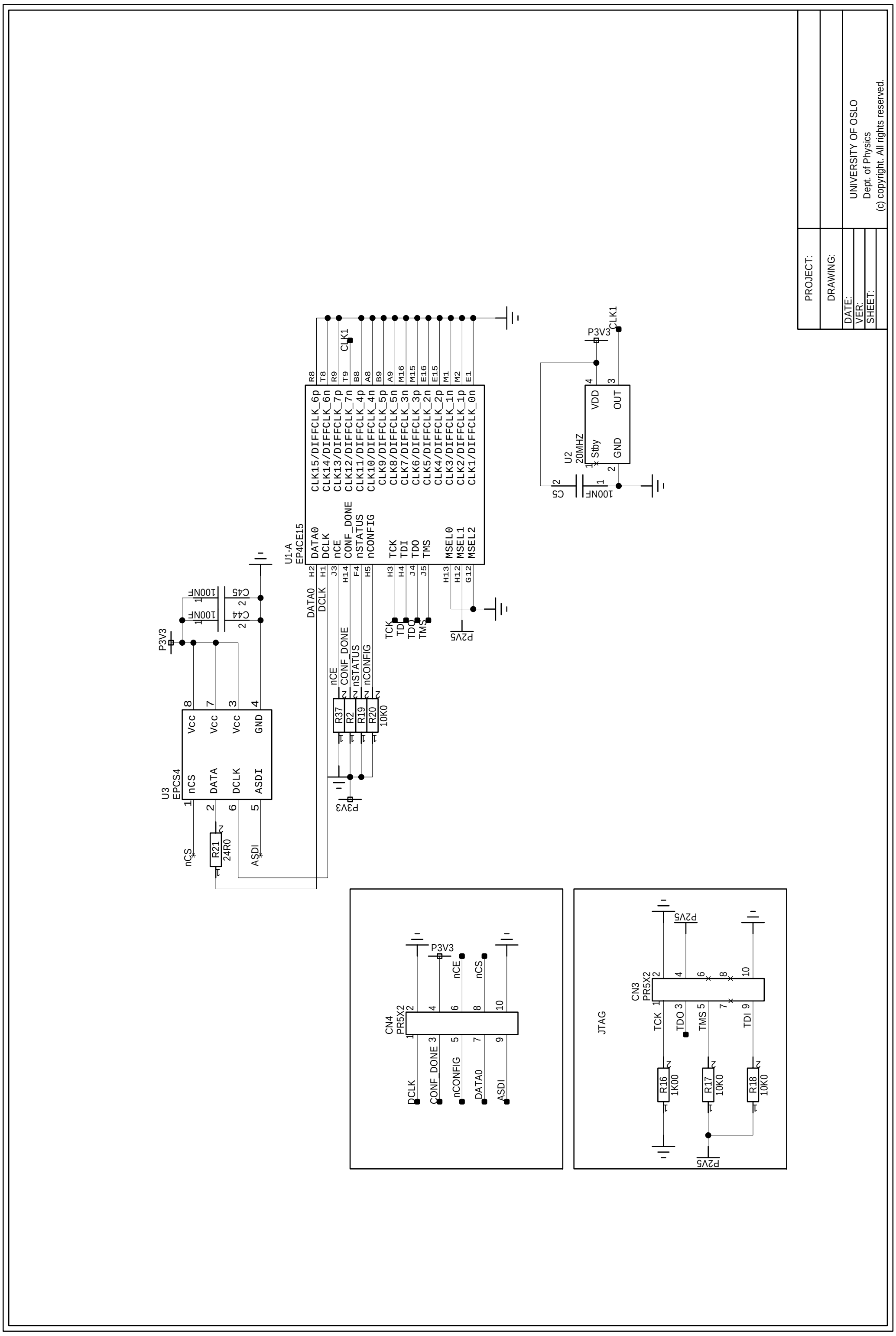


Figure A.3: DAQ digital front-end, FPGA clock and programming interface.

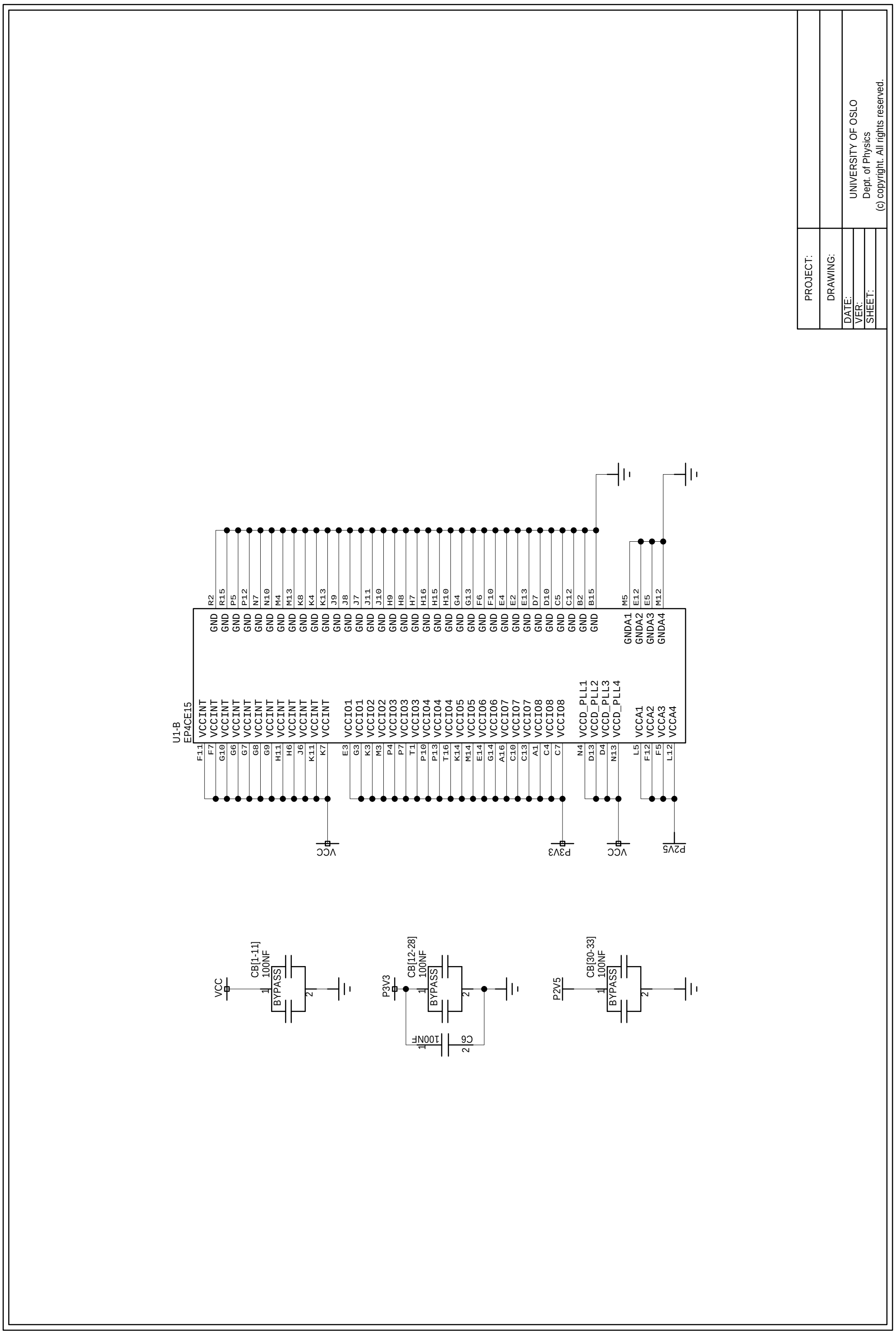


Figure A.4: DAQ digital front end, FPGA power.
)

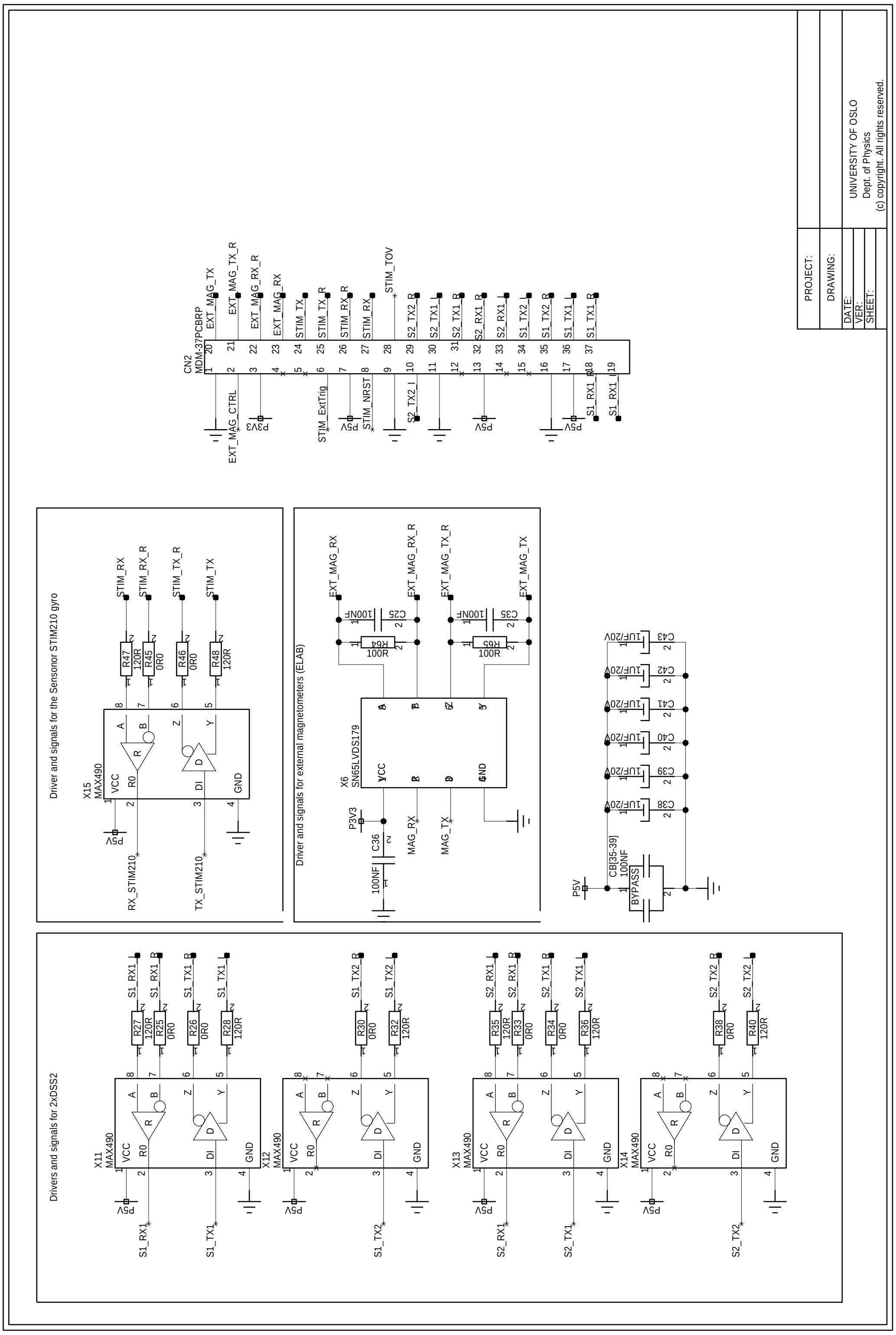


Figure A.5: DAQ external sensor interfaces.

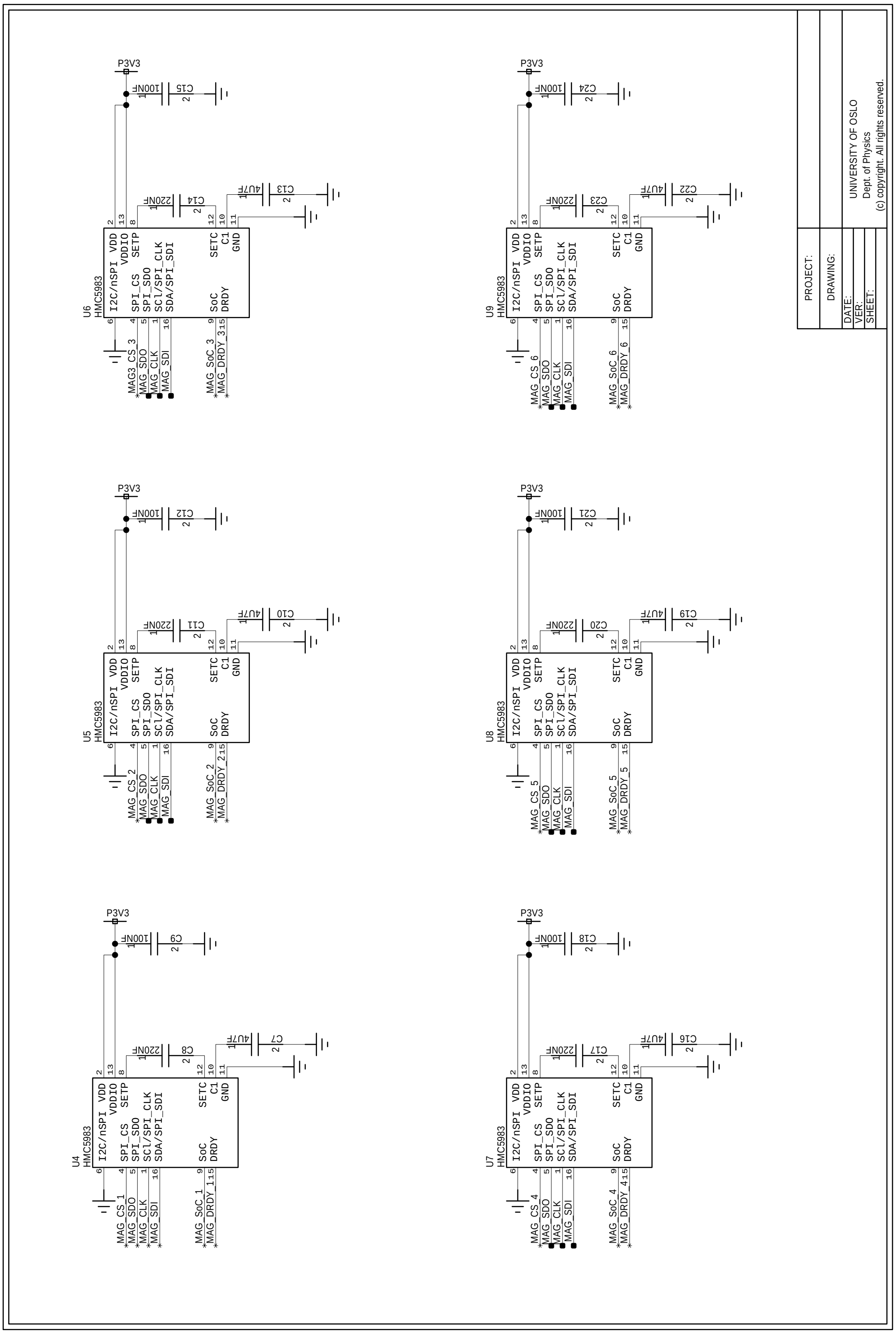


Figure A.6: DAQ internal magnetometers.

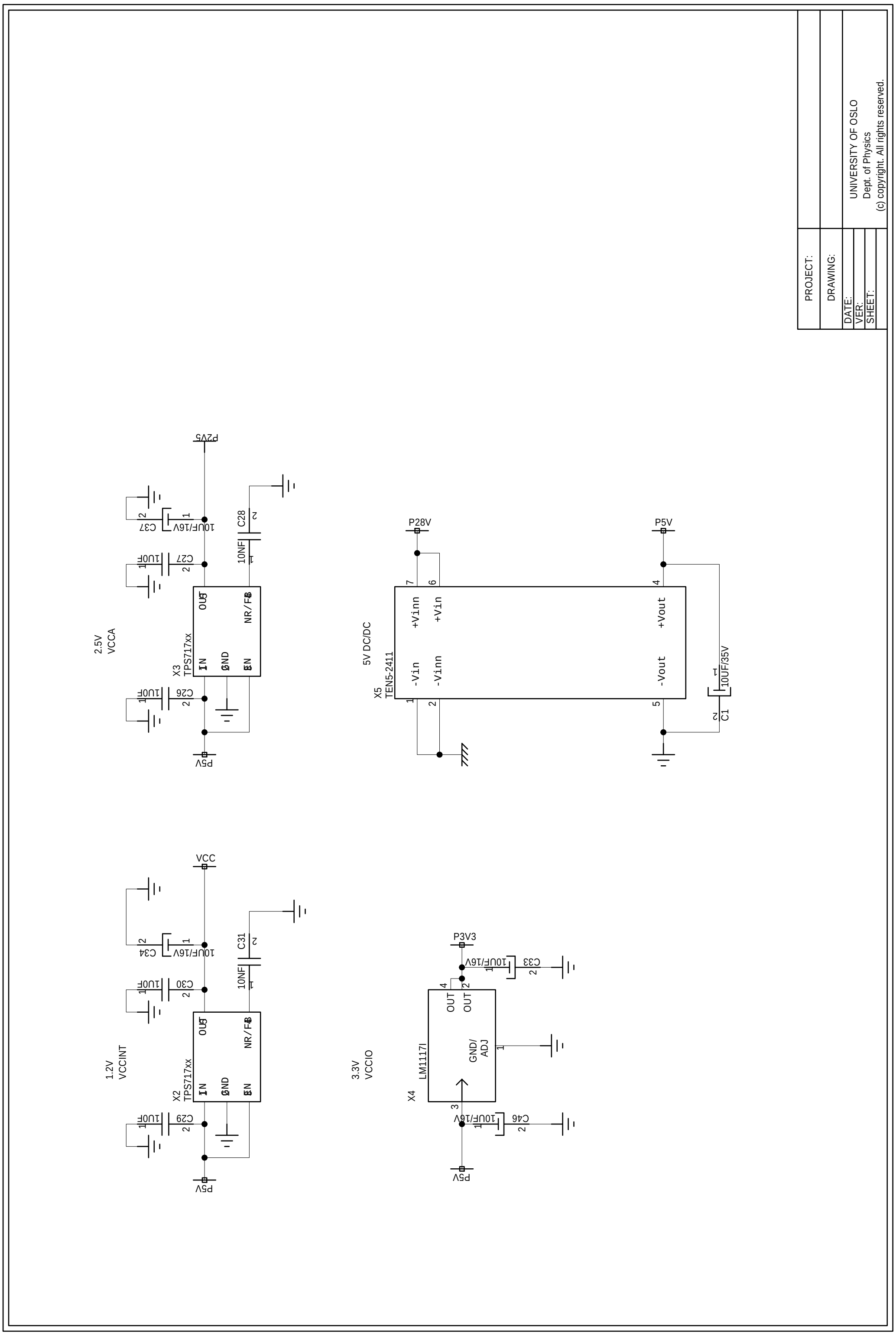


Figure A.7: DAQ power supply.

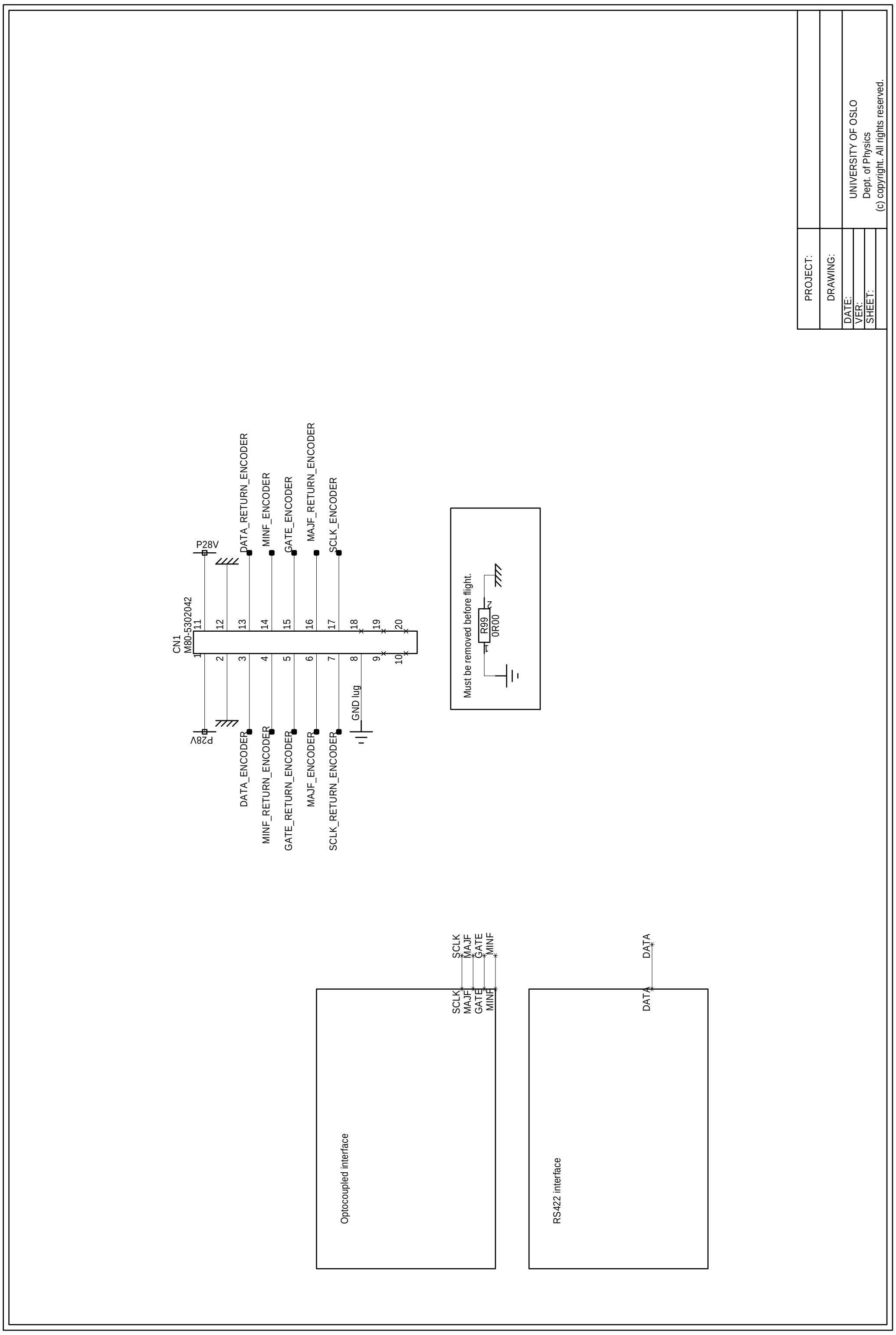


Figure A.8: DAQ encoder interface top level. (Bekkeng, 2009)

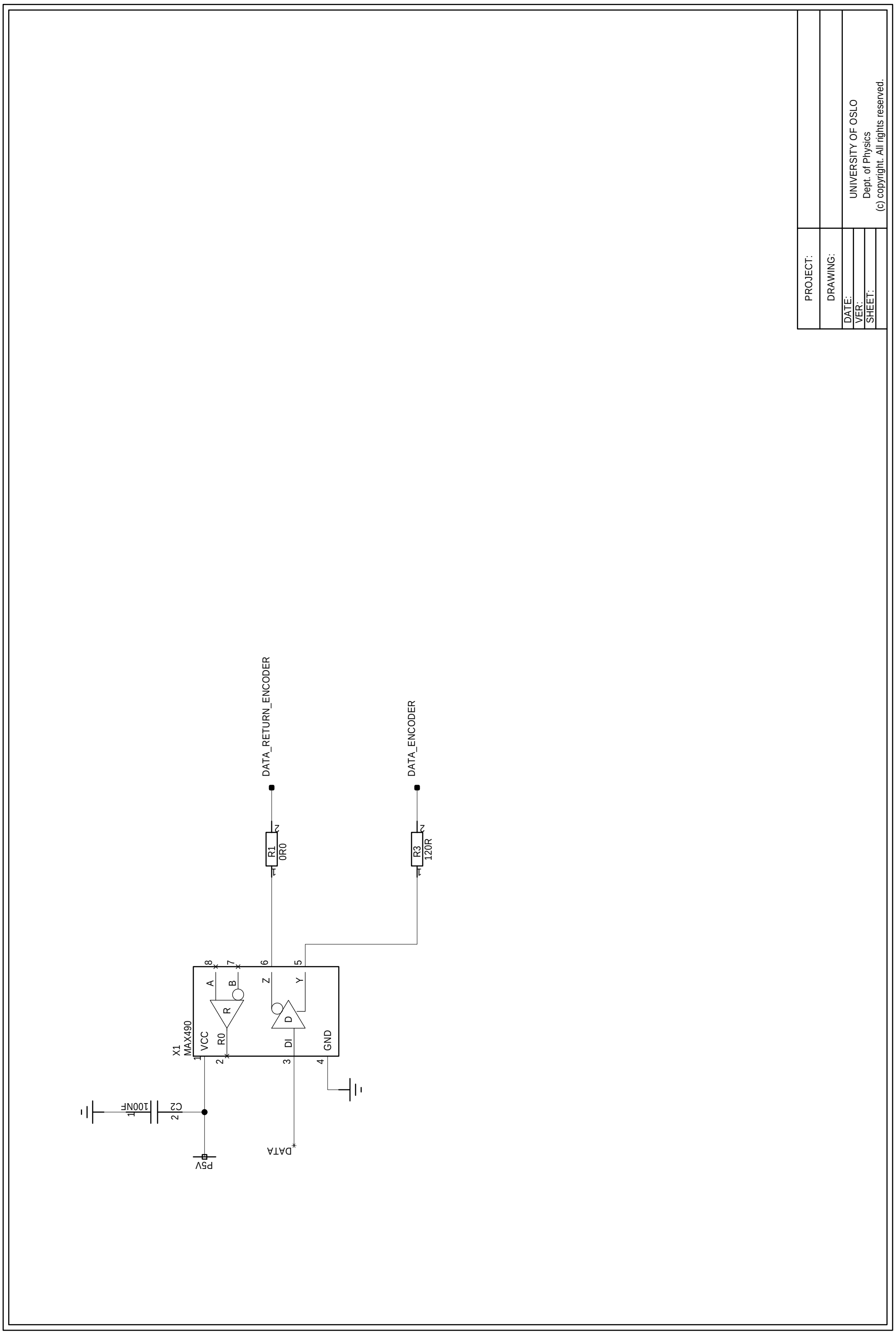


Figure A.9: DAQ encoder interface RS-422. (Bekkeng, 2009)

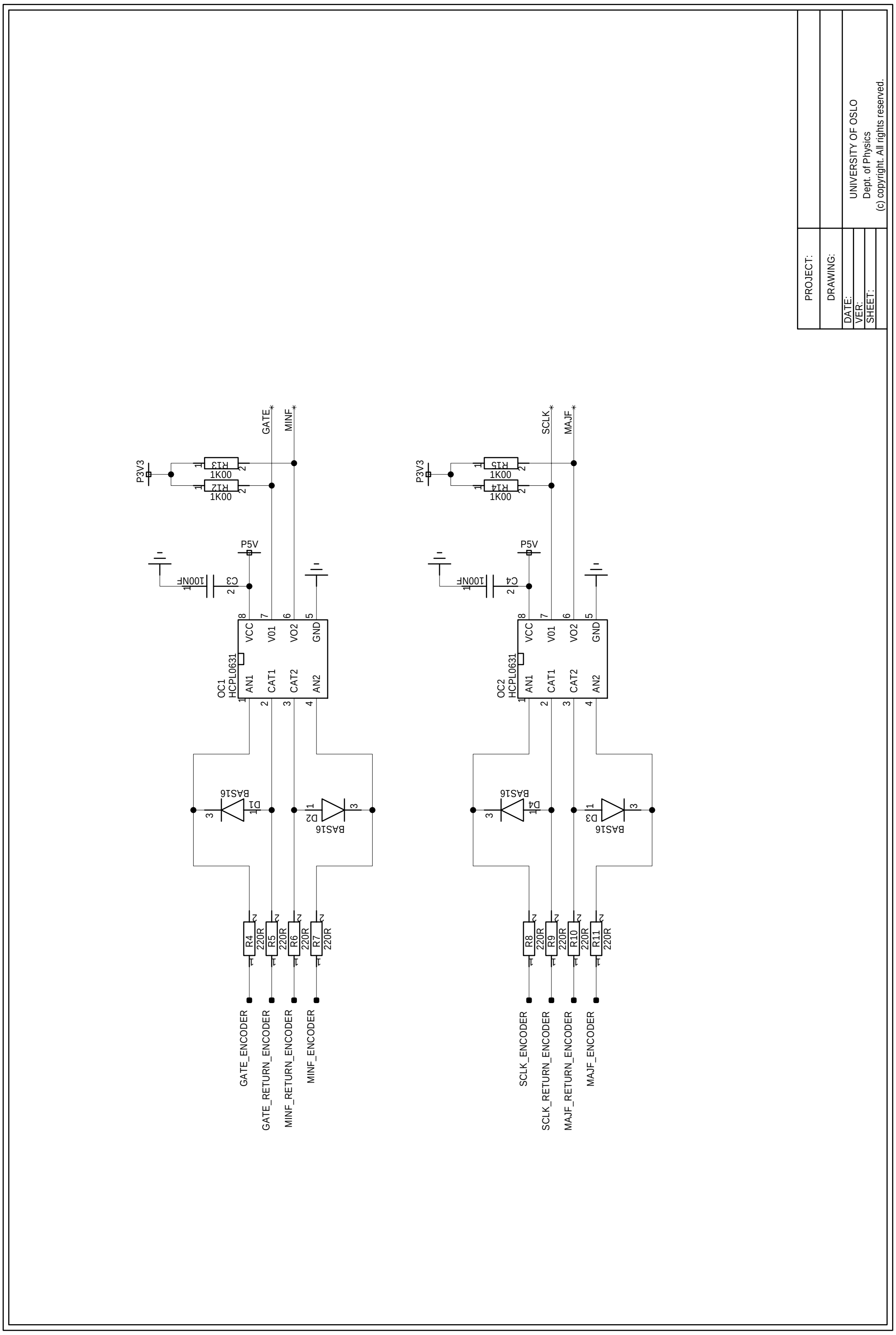


Figure A.10: DAQ encoder interface optocouplers. (Bekkeng, 2009)

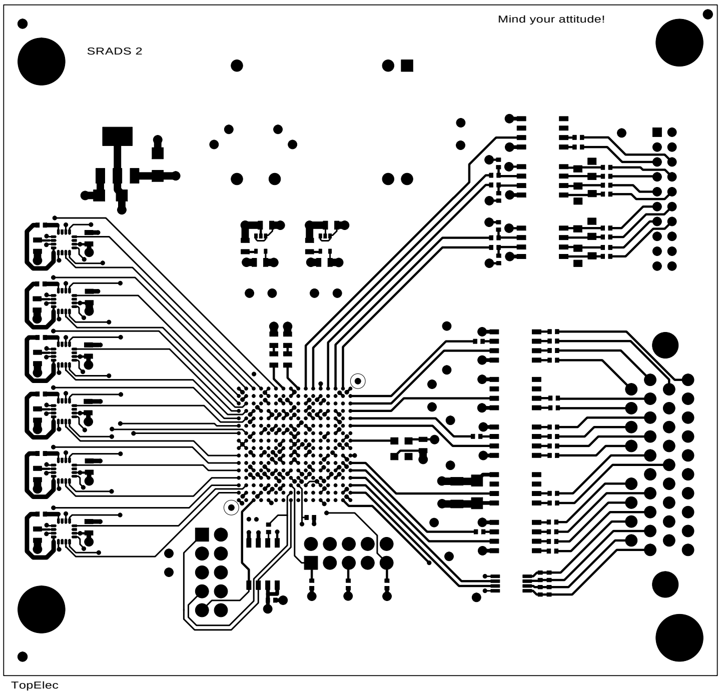


Figure A.11: Top electric layer

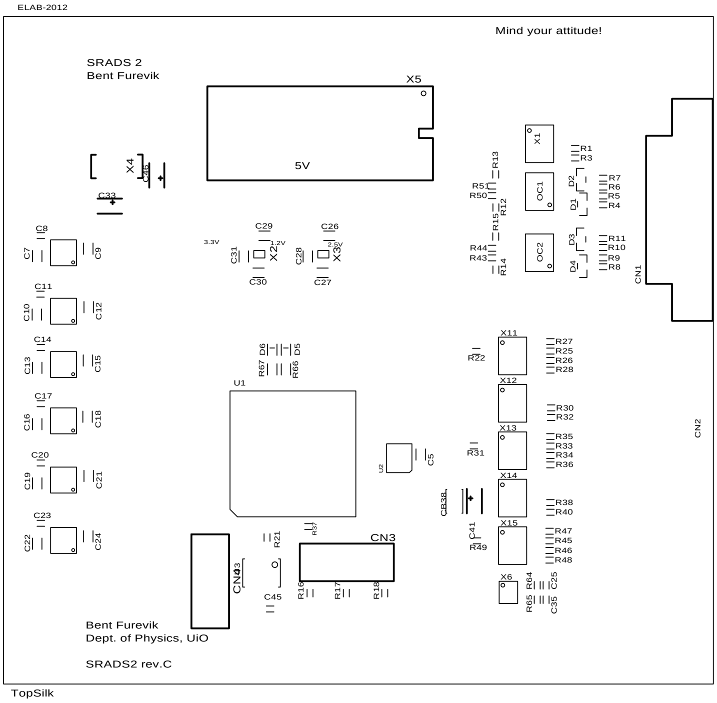


Figure A.12: Top layer silk print

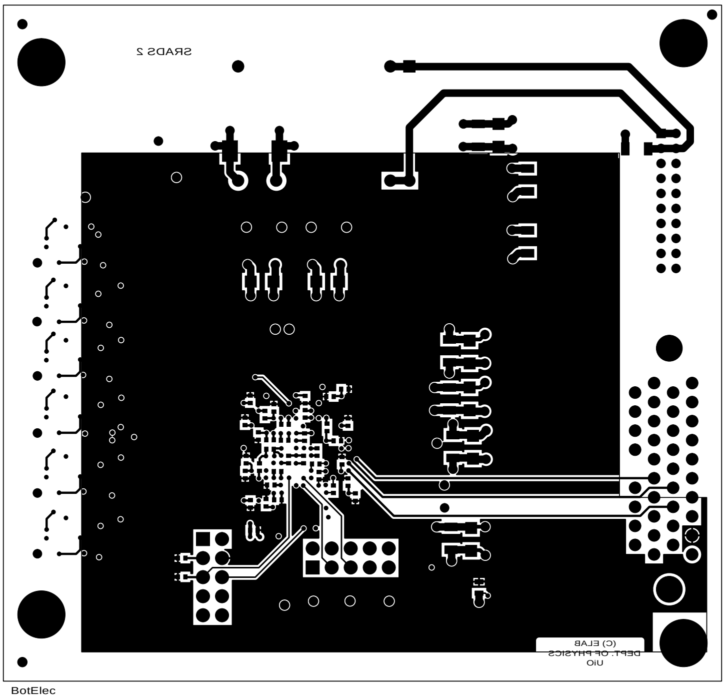


Figure A.13: Bottom electric layer

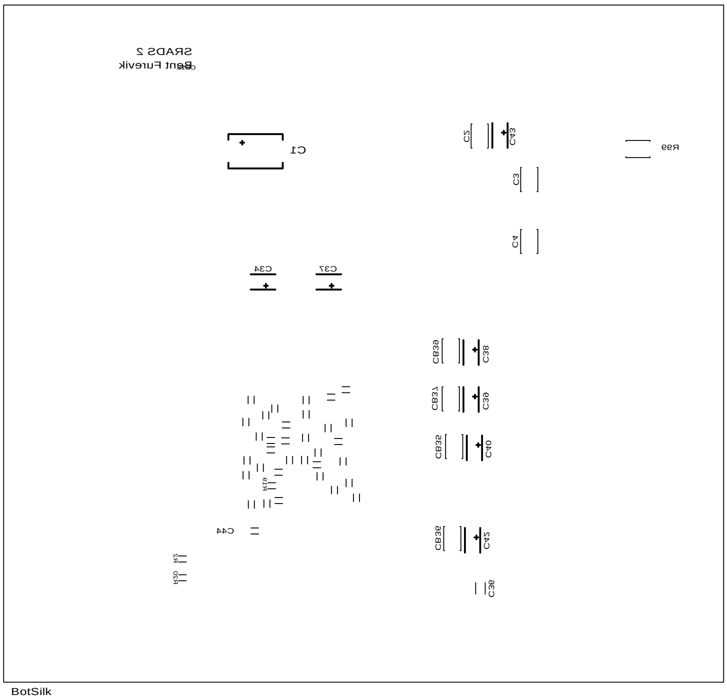


Figure A.14: Bottom silk print

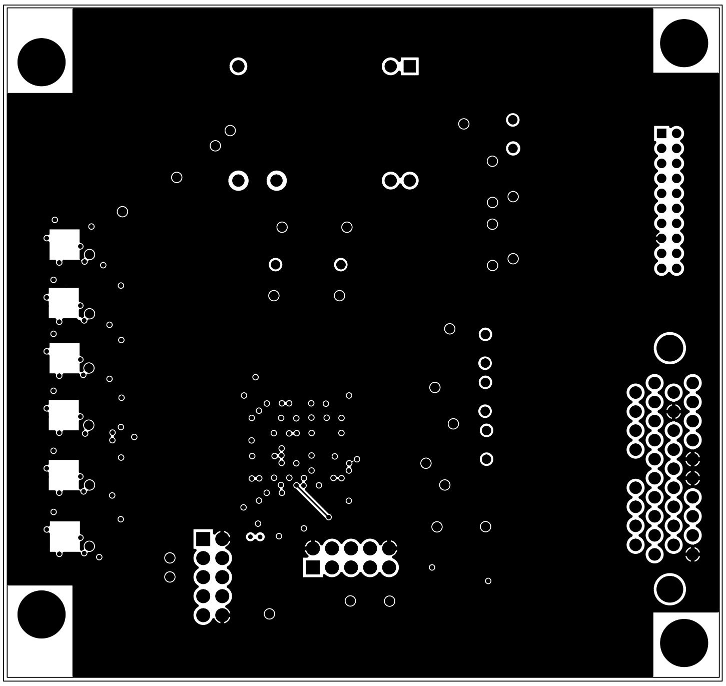


Figure A.15: Ground plane

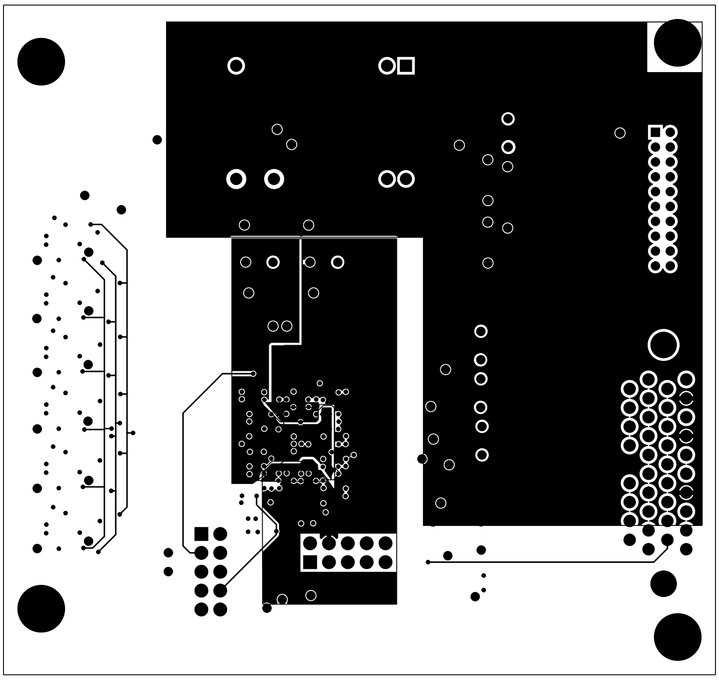


Figure A.16: Power plane

A.2 DSS2

Listing: Parts list for the DAQ board

```
------------------------------------------------------------------------------------
                                     Parts List

                        CADSTAR Design Editor Version 13.0.0.3
------------------------------------------------------------------------------------
Part Name                         Description                       Qty.   Comps.
---------                         -----------                       ----   ------

AD-SSM/AD7680/SMD                 1ch 16b ADC                        4     X5-8
AD-SSM/ADR3425/SMD                2.5V REFERENCE                     1     X11
ALTERA/EPM1270T144C5ESN/TQFP      ALTERA MAX II 1270 CPLD            1     IC1
CAP/100NF/0603R                   10% 16V 0603 X7R                   17    C1-4
                                                                           C7-10
                                                                           C13-20
                                                                           C33
CAP/100PF/0603R                   5% 50V 0603 NP0                    4     C21-24
CAP/10NF/0603R                    10% 50V 0603 X7R                   2     C28-29
CAP/10UF/1206R                    Kemet 10V 1206 Y5V                 1     C5
CAP/1U0F/0603R                    10% 25V 0603 X5R                   8     C11-12
                                                                           C25-27
                                                                           C30-32
CAP/BYPASS/0603R                  10% 16V 0603 X7R                   16    CB10-19
                                                                           CB110-115
CON/Datamate M80 2x5 RA/HOR       Datamate M80 2x5 pins Right An     1     CN3
CON/PR5X2                         5X2 SCOTT ELEC. PINROW             1     CN2
DIO/S5991                         2D PSD                             1     D1
MAXIM/MAX490/SMD                  FULL DUPLEX TRANCIEVER             2     X10
                                                                           X12
OPAMP/OPA129/SMD                  TI OPA129                          4     X1-4
RES/0R00/0402R                    RES KOA 0402 1% 63mW MINIREEL      9     R13-15
** 0402 RES **                                                             R21
                                                                           R24-25
                                                                           R32-34
RES/0R00/0603R                    RESISTOR KOA 0603 1% 0.1W          8     R5-8
                                                                           R28-31
RES/100K/0603R                    RESISTOR KOA 0603 1% 0.1W          1     R9
RES/10K0/0603R                    RESISTOR KOA 0603 1% 0.1W          2     R11-12
RES/120R/0402R                    RES KOA 0402 1% 63mW MINIREEL      3     R16-18
** 0402 RES **
RES/1K00/0603R                    RESISTOR KOA 0603 1% 0.1W          1     R10
RES/360R/0402R                    RES KOA 0402 1% 63mW MINIREEL      2     R19-20
** 0402 RES **
RES/4K70/0603R                    RESISTOR KOA 0603 1% 0.1W          4     R1-4
RES/5K10/0402R                    RES KOA 0402 1% 63mW MINIREEL      4     R22-23
** 0402 RES **                                                             R26-27
TI/TPS60403/SMD                   Voltage inverter                   1     X9
TI/TPS717xx/SMD                   LDO regulator                      2     X13-14
XTAL/ASEMB20MHZ                   ABRACON MEMS CLOCK OSCILLATOR      1     U1

------------------------------------------------------------------------------------
                                    End of report
------------------------------------------------------------------------------------
```

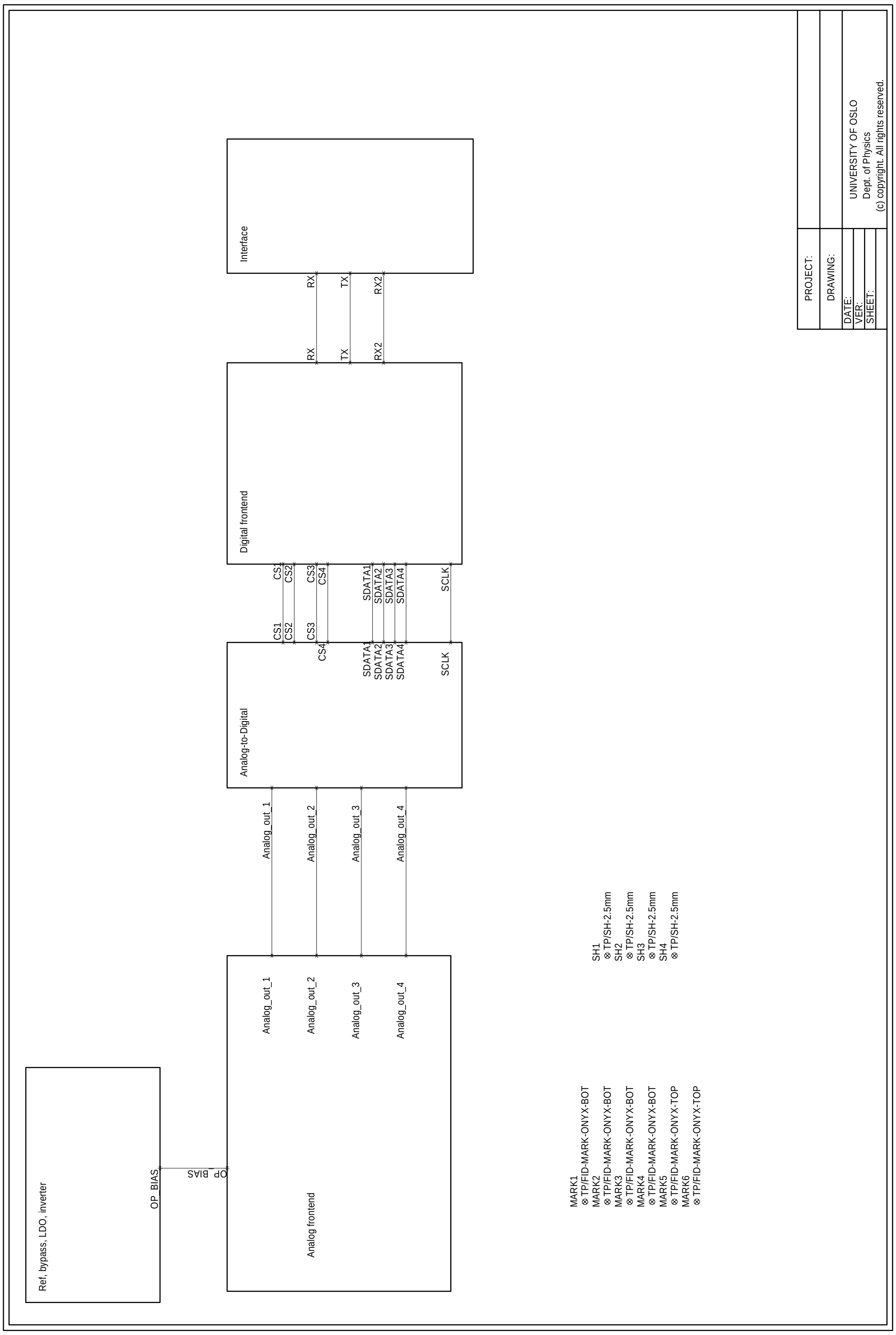


Figure A.17: DSS2 top level schematic.

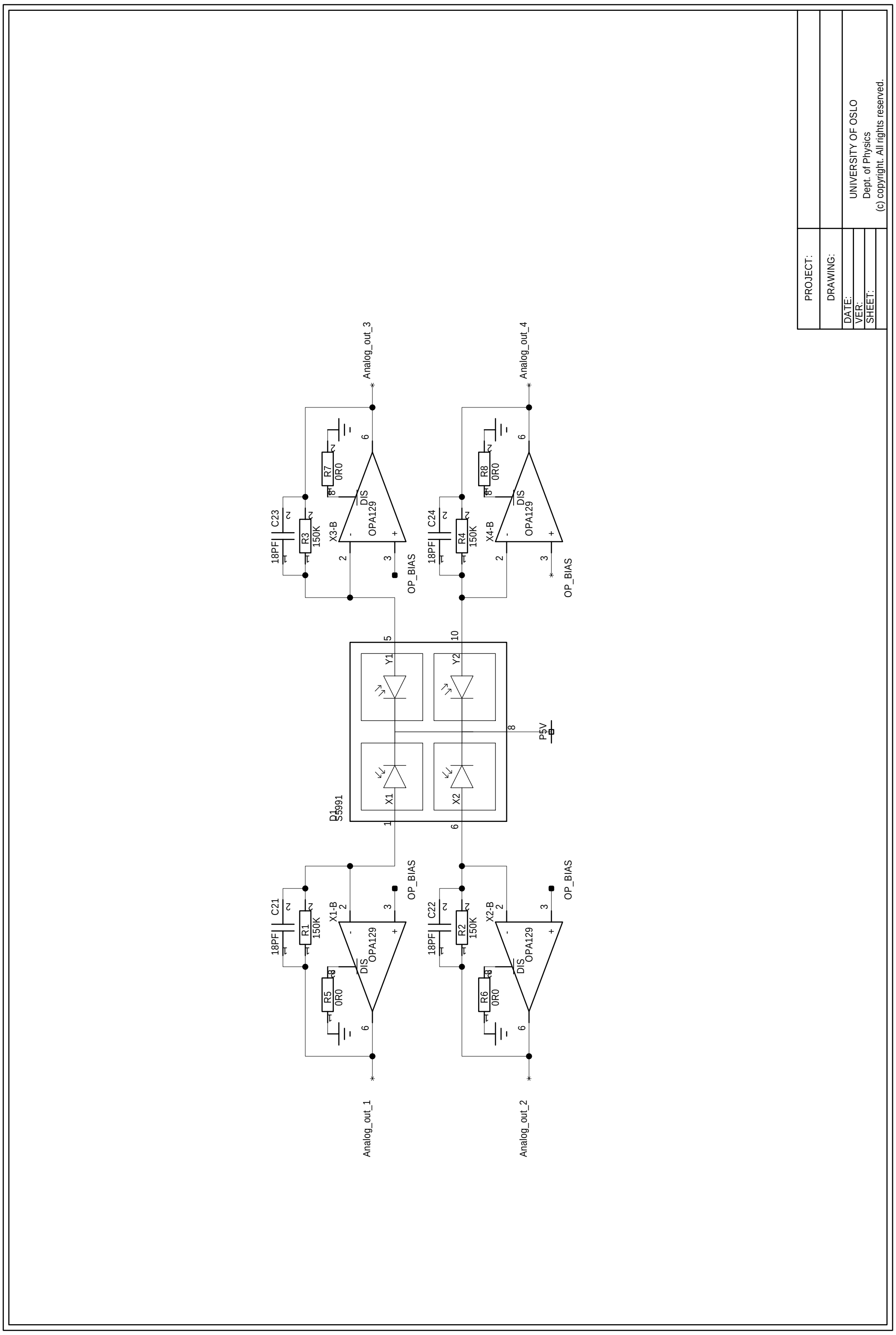


Figure A.18: DSS2 analog front-end.

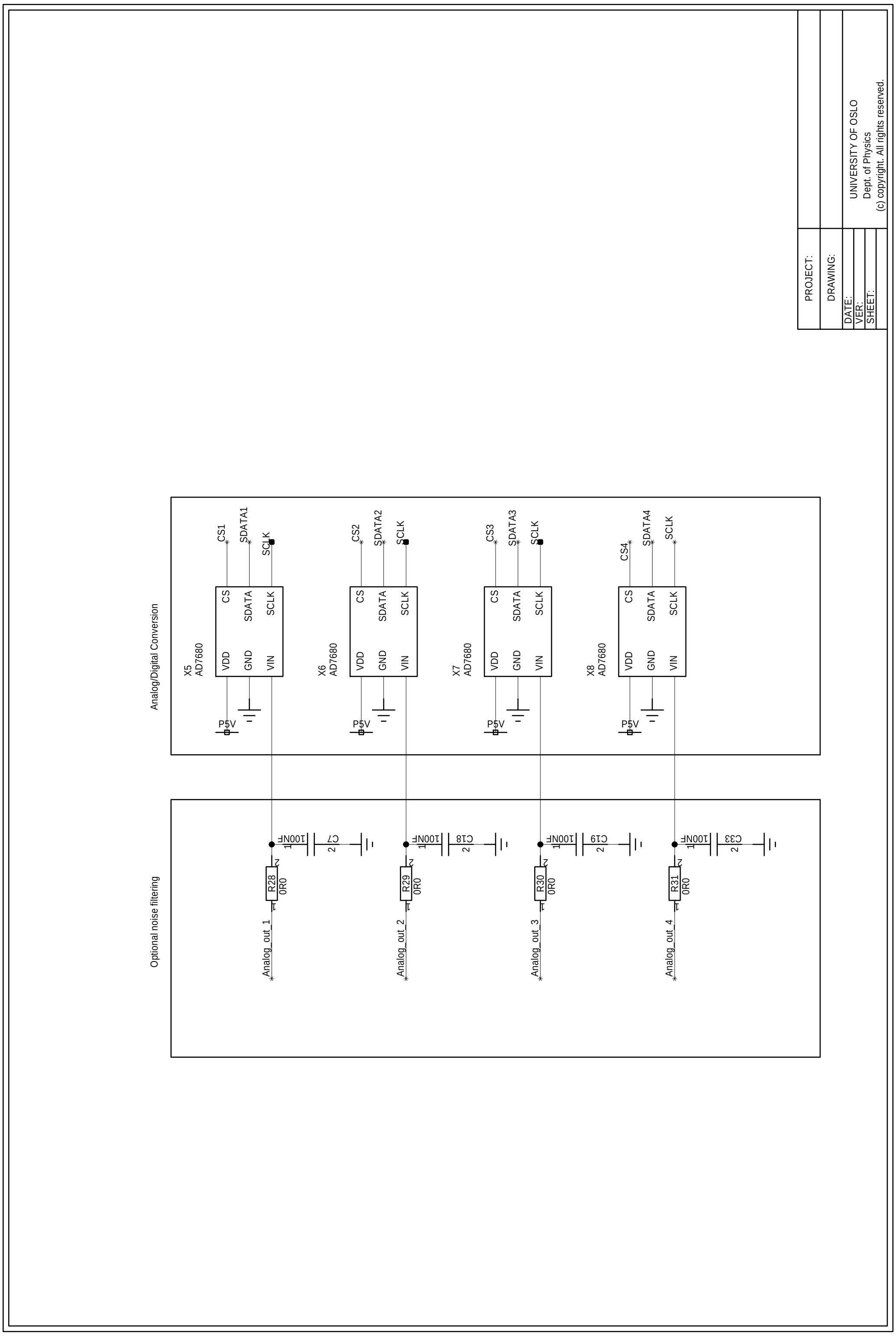


Figure A.19: DSS2 analog to digital and optional noise filtering.

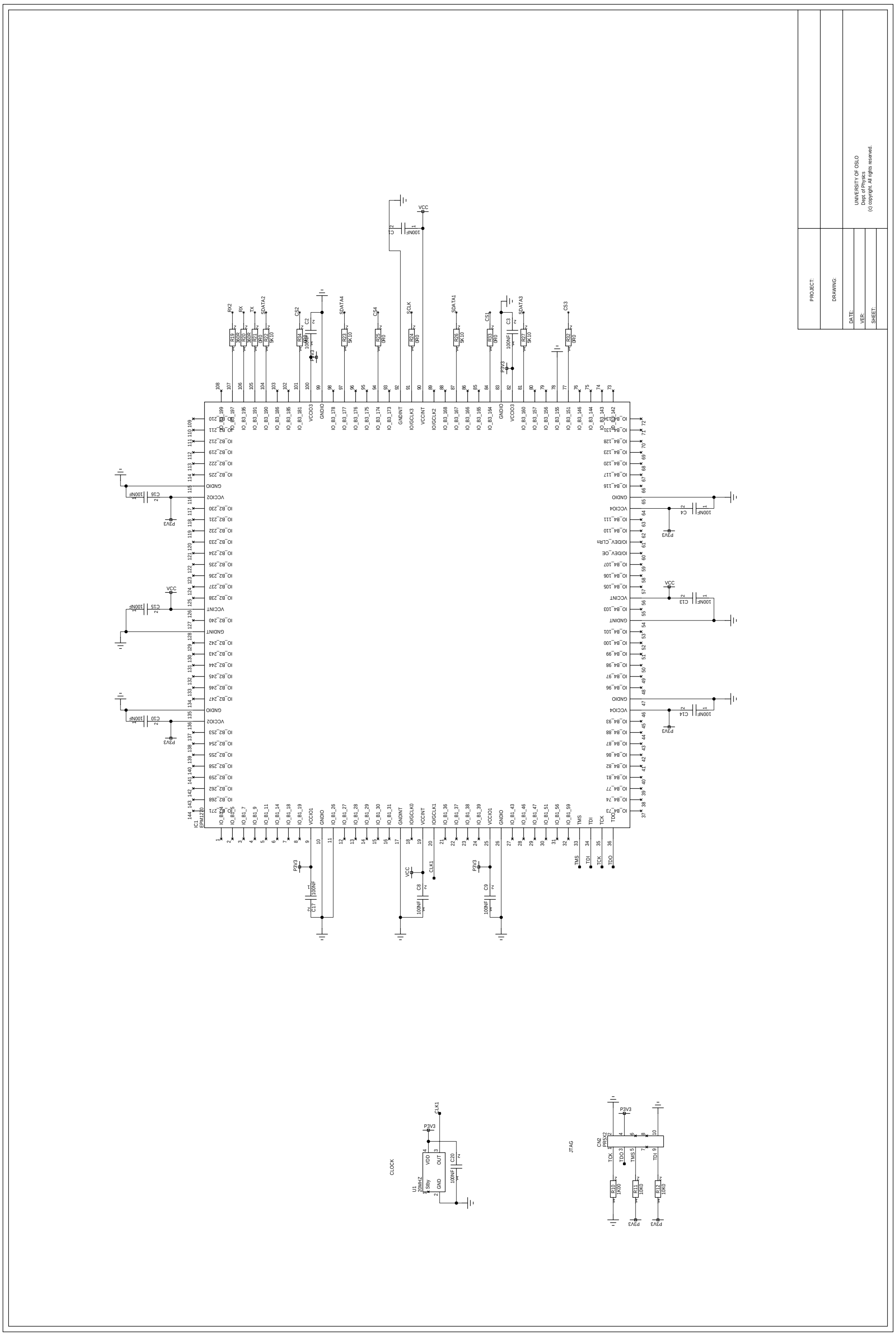


Figure A.20: DSS2 digital front-end.

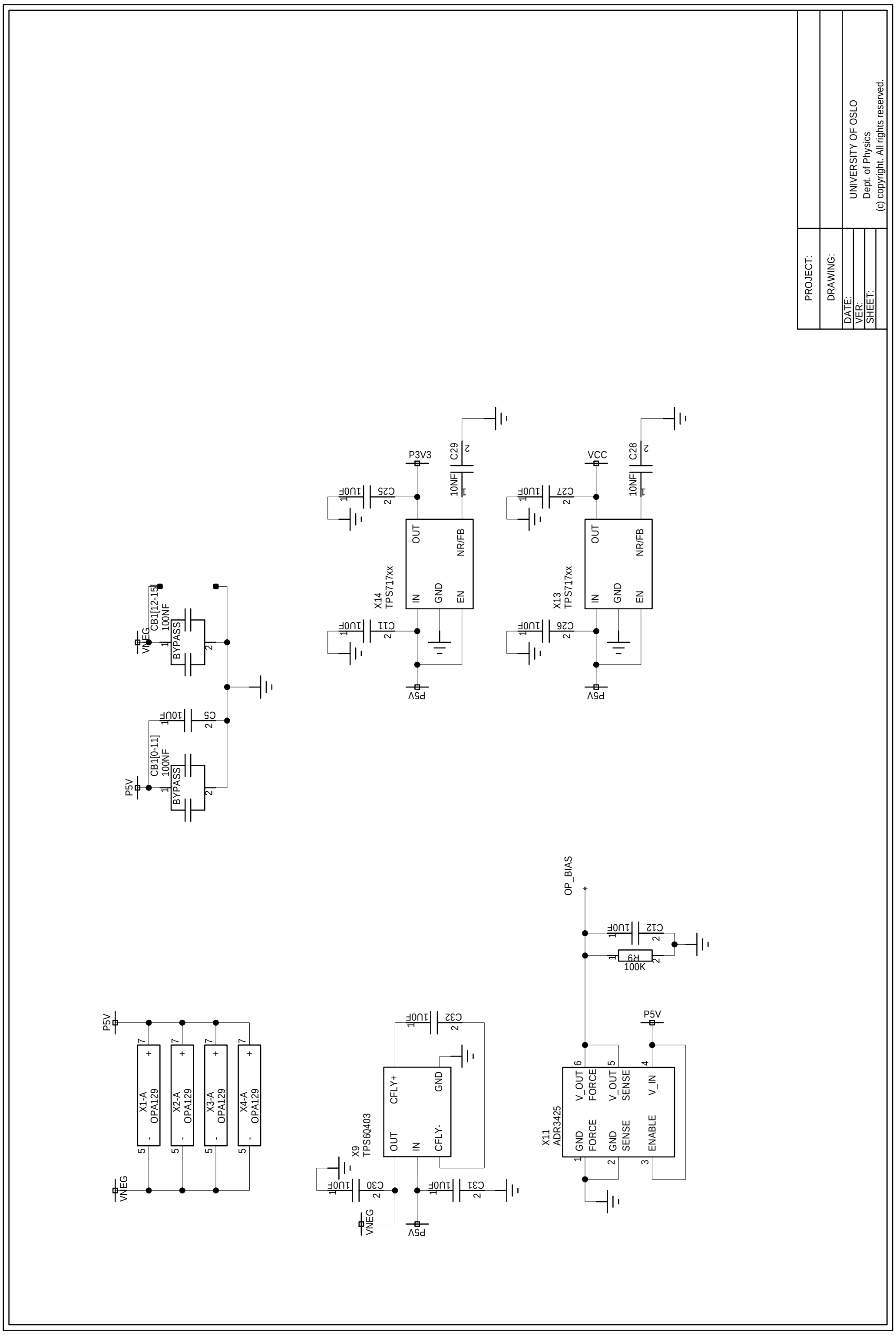


Figure A.21: DSS2 power supply and reference voltage.

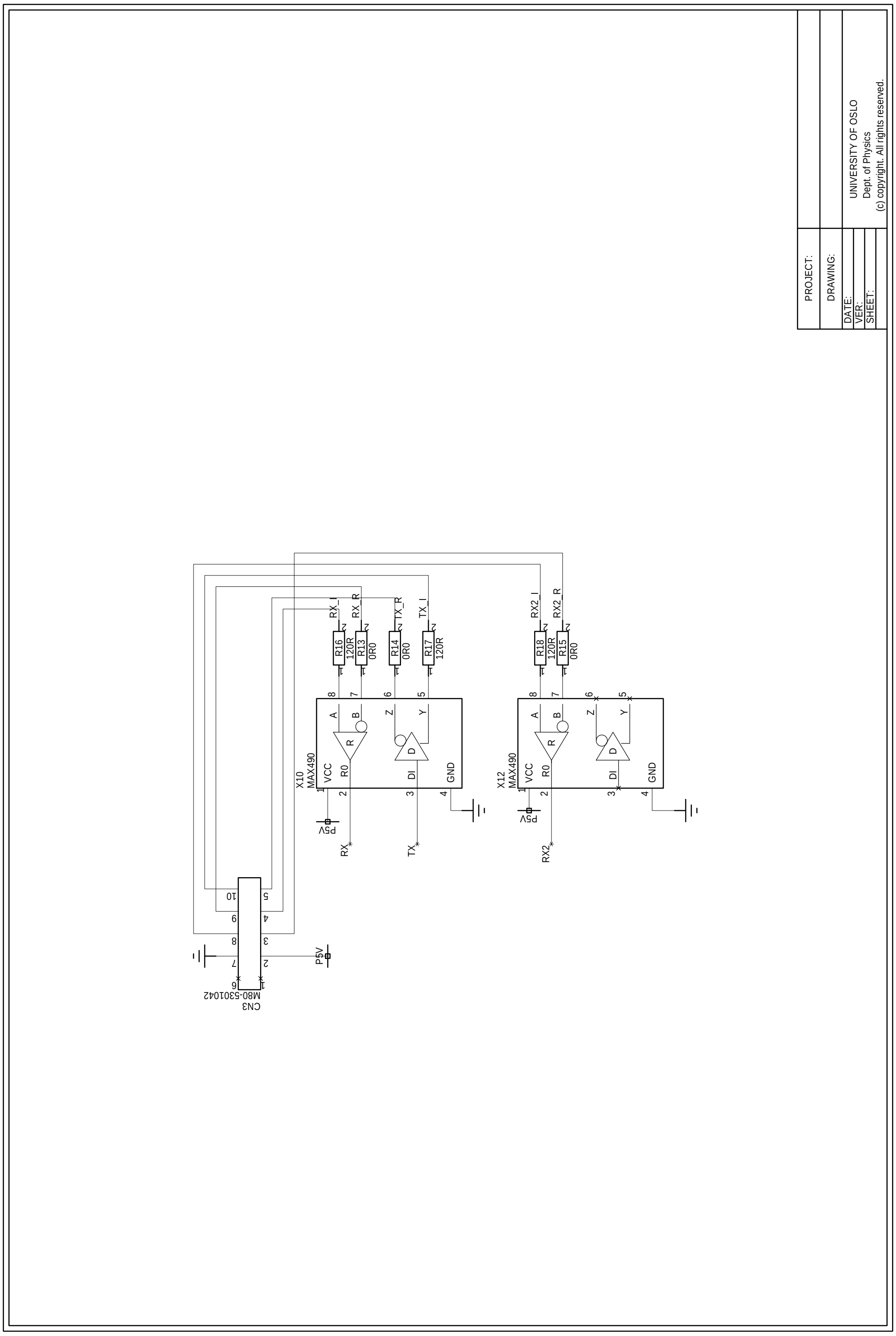


Figure A.22: DSS2 interface to DAQ-board.

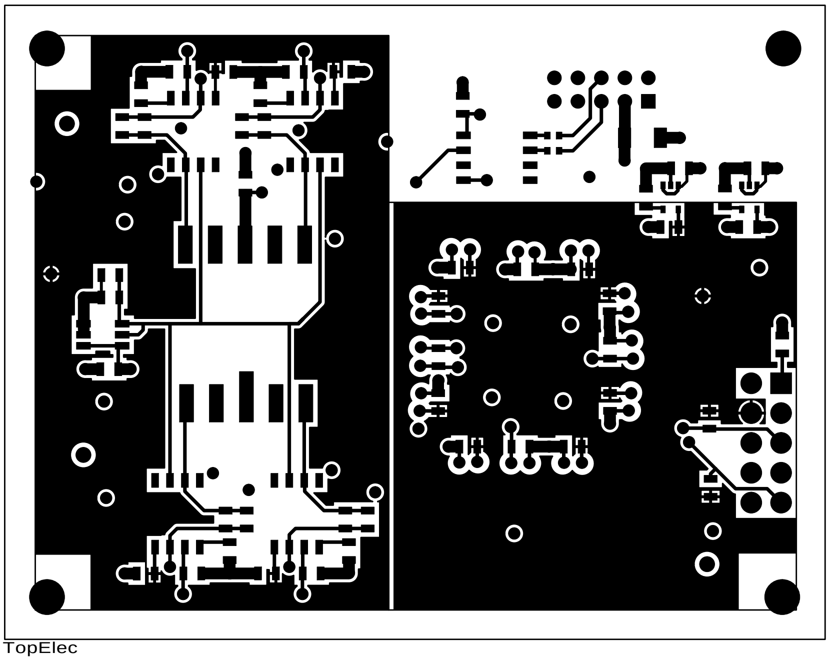


Figure A.23: Top electric layer

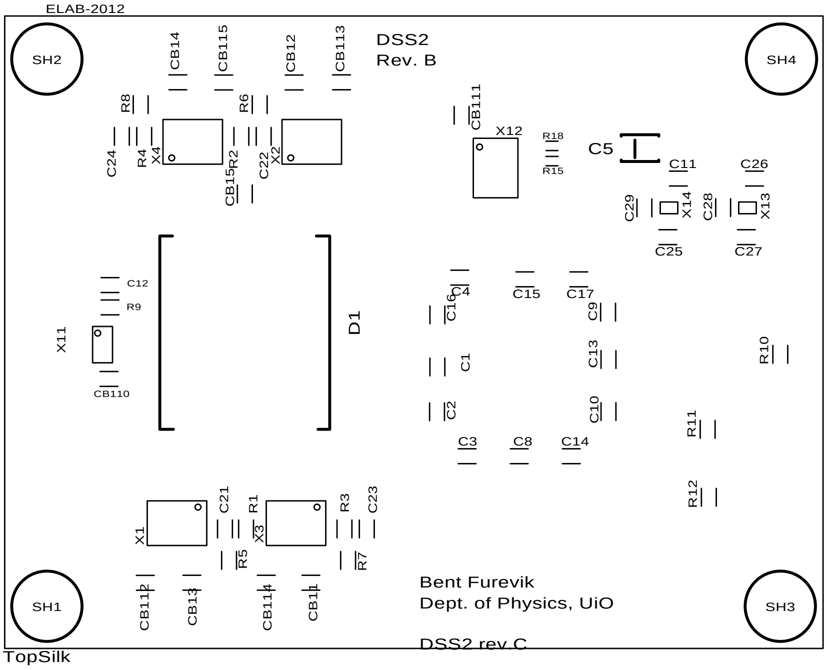


Figure A.24: Top layer silk print

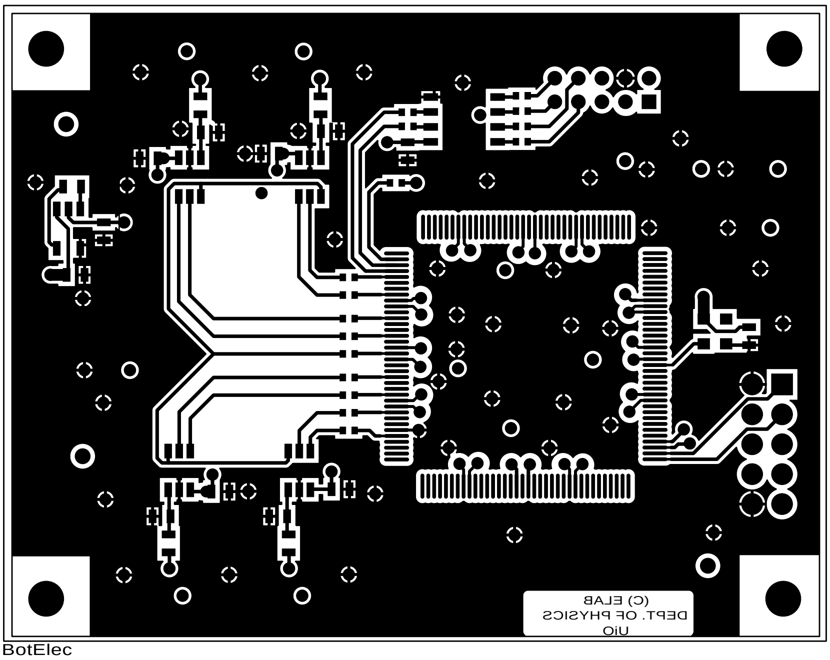


Figure A.25: Bottom electric layer

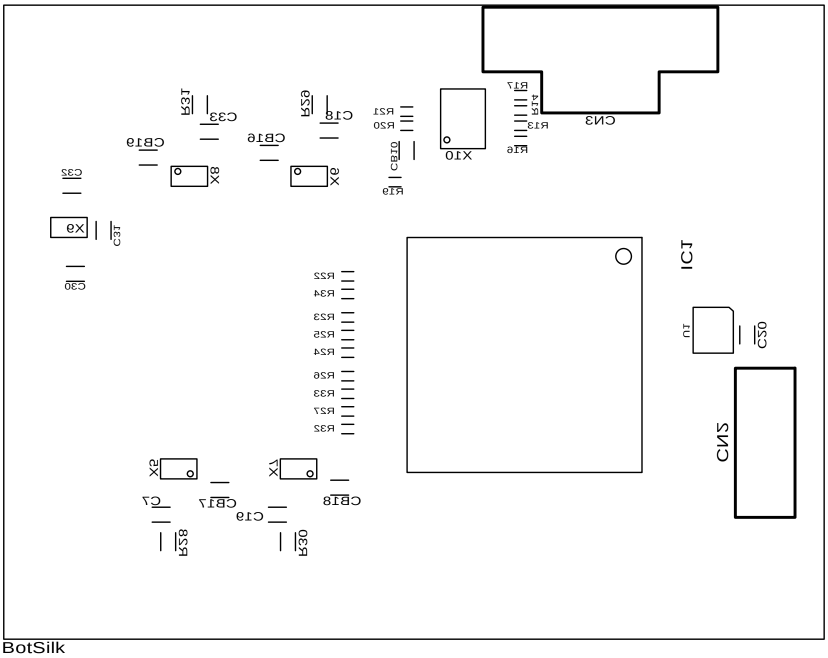


Figure A.26: Bottom silk print

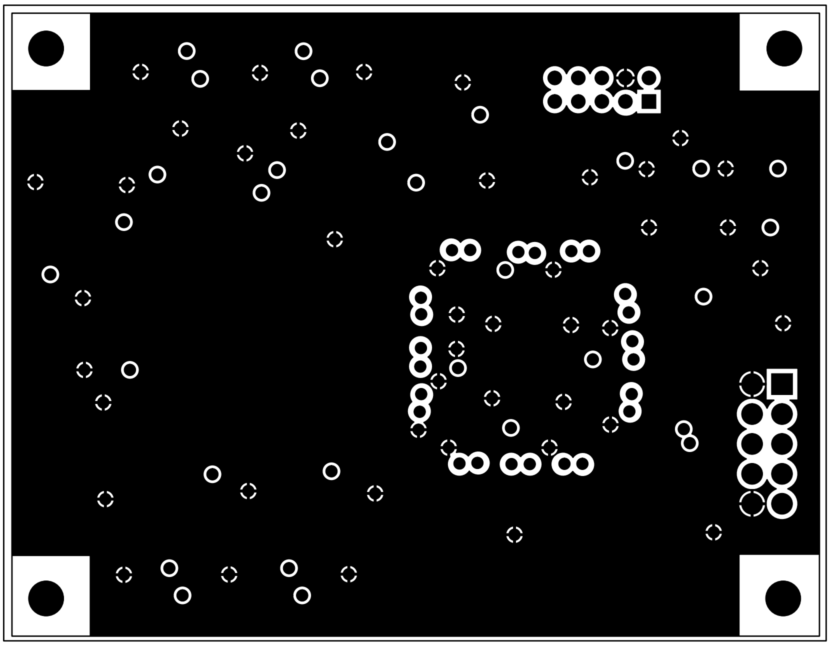


Figure A.27: Ground plane

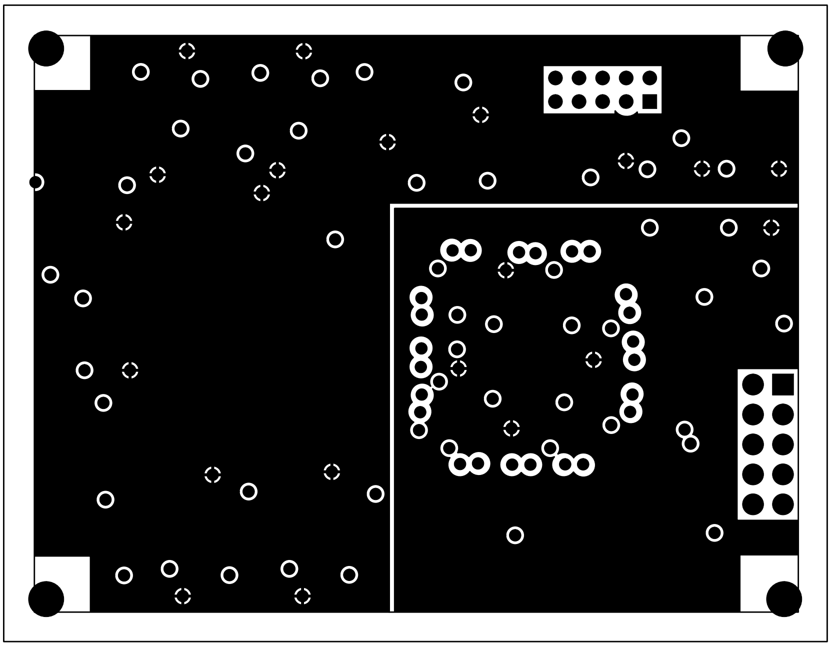


Figure A.28: Power plane

A.3 Design revisions

This section covers the design revisions of the DAQ and DSS2 PCBs. Design changes and reasons for making these are briefly covered.

A.3.1 DAQ

Revision A

The first version of the DAQ board was not operational, as it was impossible to program the FPGA. This turned out to be caused by the configuration mode selection pins of the FPGA was set wrong. It was assumed that the setting of these pins was irrelevant for JTAG mode, as the datasheet specifies that the setting will be ignored when using JTAG programming. As it was planned to program the configuration circuit by using the Serial Flash Loader over JTAG, the option of an active serial connection was deemed unnecessary.

It was also noticed that the TPS71733 LDO regulator were unable to produce a monotonically rising voltage for the 3.3 V net. This could possibly be caused by a too high capacitive load on the net.

Revision B

The second revision, pictured in Figure A.29, was a minor correction on the previous design. The configuration mode selection pins of the FPGA were connected correctly. A connector was added for Active Serial configuration, in case the JTAG would fail again.

The TPS71733 LDO was replaced by the LM1117-N-3.3, which has a higher maximum current output of 800 mA compared to the 150 mA of the TPS717 series. When the board was mounted, it became clear that the footprint in the database was erroneous.

Revision C

The only change in the third revision of the board is that the footprint of the LM1117-N-3.3 LDO regulator was fixed. Pictures of the board are included in Section 4.6.4.

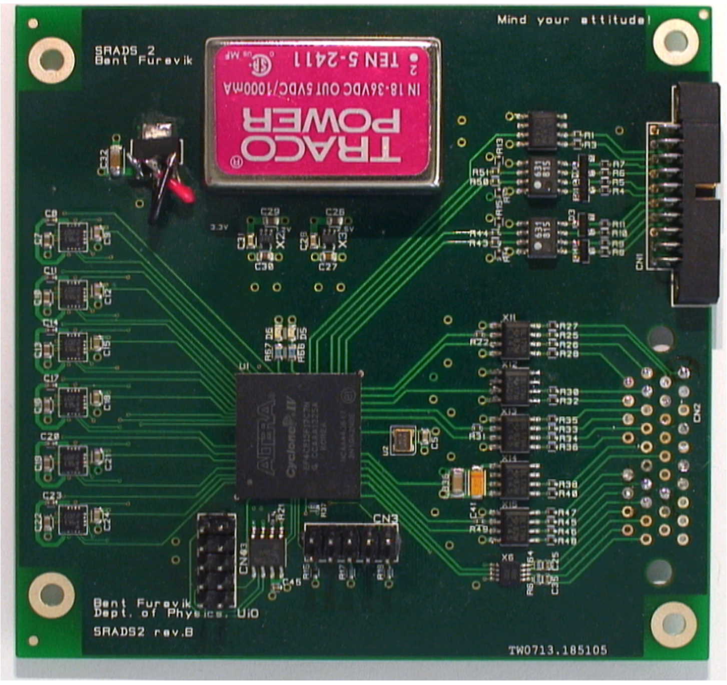


Figure A.29: The DAQ PCB, revision B.

A.3.2 DSS2

Alpha

The alpha board, seen in Figure A.30, was a circuit created to evaluate the function of the Hamamatsu S5991 and the other components. It is a simple two-layer board with mostly 1206 components, to allow for easy debugging. The ± 12 V and 5 V rails used throughout the board is supplied on the small pinrow connector on the bottom of the board.

The ADC used in this revision was the AD7894.

It was discovered that the ADR3425 2.5 V references were oscillating at a frequency of approximately 170 kHz with a magnitude of 2.5 V_{pk-pk}. The oscillations was completely eliminated by adding a 100 kOhm resistor to ground. This suggests that these references need some current drawn to be able to turn fully on.

Revision A

The first revision, as seen in Figure A.31 of the card added a CPLD and power supply circuits to the Alpha design.

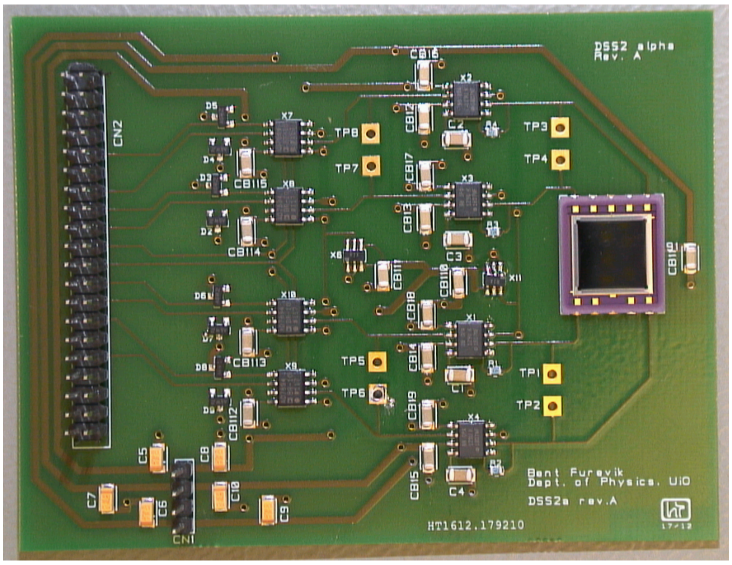


Figure A.30: The DSS2 Alpha PCB.

As can be seen in the picture of the bottom side, the power routing of this board became very complex. Because of the many power nets (+12 V, -12 V, 5 V, 3.3 V, 1.8 V), supplied externally, it became a difficult task to do this in an aestethically pleasing manner. This was most likely a source of noise and interrupted ground currents as well.

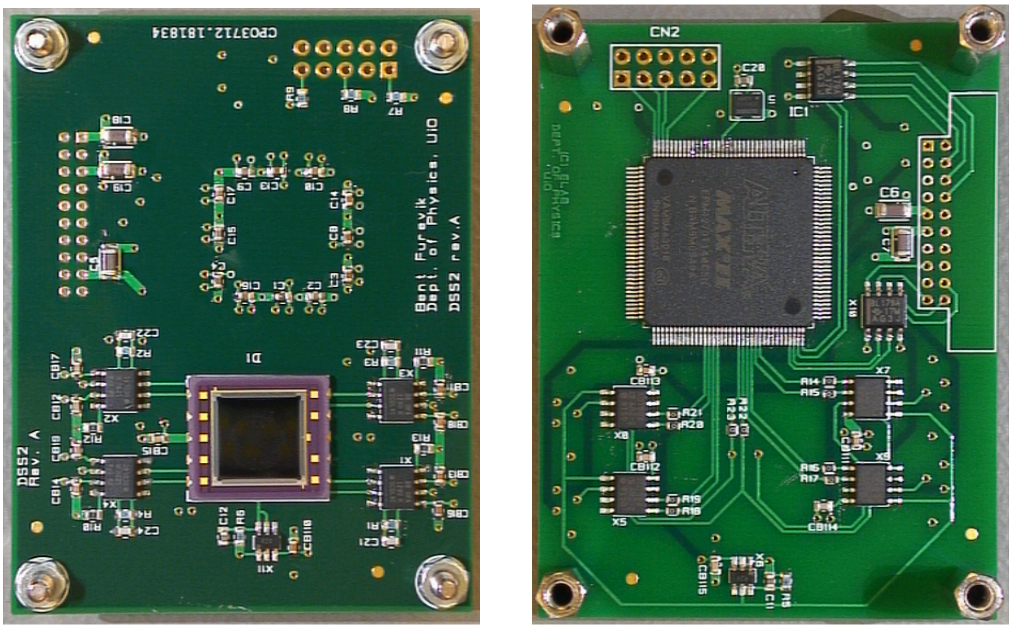
Revision B

The AD7984 ADC was exchanged for the AD7680, which had fewer signal lines. This lead to a simpler routing of the ADCs. As they had an internal reference, the external reference was not necessary either.

It was decided to lower the supply voltage of the amplifiers to 5 V, to make the routing simpler. This worked well with the 2.5 V reference level, which were the highest output voltage to be produced.

Two LDO regulators were implemented, the TPS71733 for 3.3 V and TPS71718 for 1.8 v, to supply the voltages for the CPLD. As the Max V CPLD was not available at first, a Max II was put in its place as it is compatible for this design. The 1.8 V LDO was swapped for another 3.3 V unit when using the Max II, which made the identical pinouts a big advantage. A voltage inverter was chosen to supply the -5 V rail for the opamps.

Supplying the -5 V, 1.8 V and the 3.3 V rail internally meant that the harness connecting the DSS2 to the DAQ could be reduced significantly. As the board only needed a 5 V supply instead of ± 12 V, 5 V, 3.3 V and 1.8 V,



(a) The top side of the PCB.

(b) The bottom side of the PCB.

Figure A.31: The DSS2 PCB, revision A.

it meant that the DAQ board would be simpler as well.

The second revision looks almost identical to the third revision. A picture was therefore not included.

Revision C

The only change done in the third revision was to modify the ground and power planes, as discussed in Section 4.6.5. It was done to increase shielding and to remove the planes that surrounded the screwholes. If the screwholes were tightened too hard, it could lead to a short circuit.

Pictures of the board are included in Section 4.6.5.

Appendix B

VHDL Code and ASM diagrams

This appendix includes the VHDL code and ASM diagrams developed through this thesis.

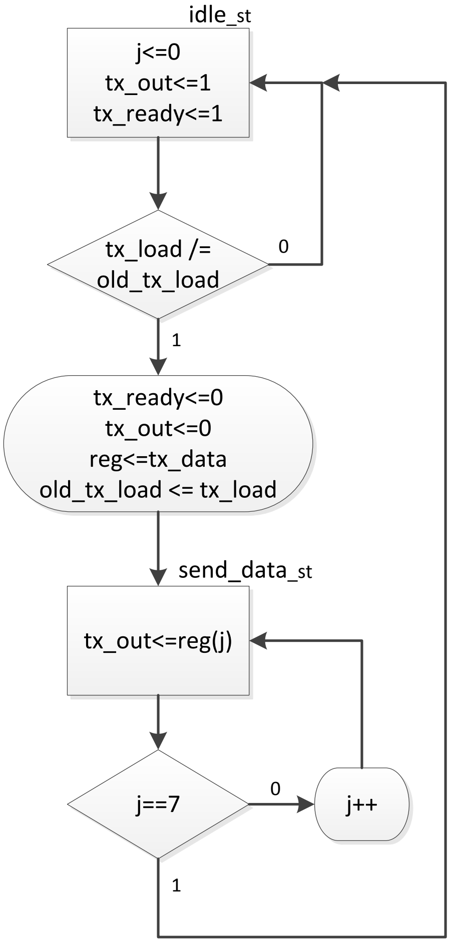
B.1 UART Module

The UART module developed in this thesis is used both in the DAQ and the DSS2. It is also employed in the external magnetometer by Danielsen and Strøm.

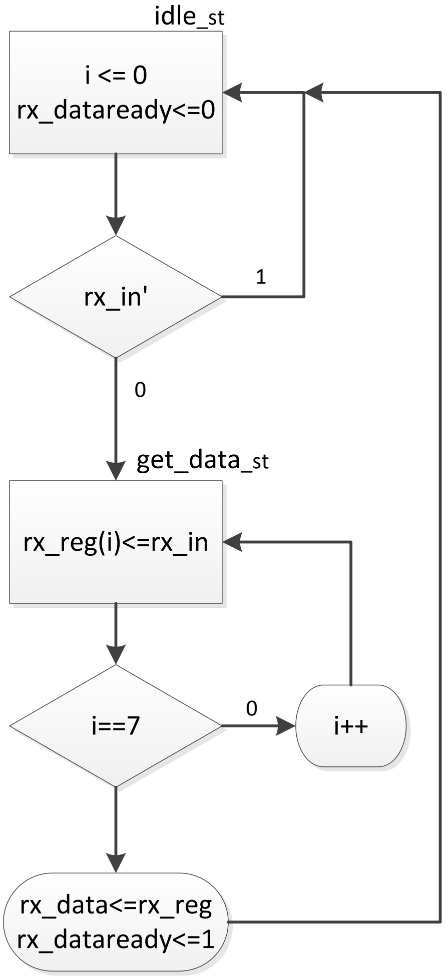
Listing: Top level and encoder interface code

```
-- Author       : Bent Furevik
-- File name    : uart_module.vhd
-- Last updated : 17.04.2013
-- Project      : General
-- Function     : General UART module. No parity, 1 stop bit, variable bit width, variable o
-- Simulation pass: Yes
-- On-chip test pass: Yes
-- Contact: www.linkedin.com/in/bentfurevik
-- Future changes: Parity bit, 1/2 stop bits, parity_error output, maybe frame_error output

------------------------------------ COMMENT ------------------------------------
-- All clocks are user supplied
-- If using X times oversampling, supply a rx_clk X times higher than the baud rate.
-- Instantiation, component declaration and helper function examples at the bottom
----------------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity uart_module is
    generic(
        width : integer := 8;
        oversampling : integer := 4
        );
    port(
        rx_clk : in std_logic;
        rx_in : in std_logic;
        rx_data : out std_logic_vector(width-1 downto 0);
        rx_data_ready : out std_logic;

        tx_clk : in std_logic;
        tx_out : out std_logic;
        tx_data : in std_logic_vector(width-1 downto 0);
        tx_load : in std_logic; -- flip bit for loading
        tx_ready : out std_logic
    );
end uart_module;


architecture rtl of uart_module is

--TX
type tx_state_type is (idle, send_data);
signal tx_state : tx_state_type;

signal reg : std_logic_vector(tx_data'range); -- temporary storage for tx data

signal j : integer range 0 to width; -- bitcounter for recieved data
signal old_load : std_logic;


--RX
type rx_state_type is (idle, get_data, delay);
signal rx_state : rx_state_type;
signal rx_reg : std_logic_vector(rx_data'range); -- temporary storage for rx data
signal i : integer range 0 to width; -- bitcounter for recieved data
signal rx_d1, rx_d2 : std_logic;
signal samplecount : integer range 0 to oversampling;

begin

genlabel1: if oversampling = 1 generate
RX_FSM: process(rx_clk)
begin
    if falling_edge(rx_clk) then
        rx_d1 <= rx_in; -- synchronization of the input
        rx_d2 <= rx_d1;
        rx_data_ready <= '0';
        i <= 0;
        case rx_state is
            when idle =>
                if rx_d2 = '0' then
                    rx_state <= get_data;
                end if;

            when get_data =>
                i <= i + 1;
                rx_reg(i) <= rx_d2;
                if i = width-1 then
                    rx_data_ready <= '1';
                    rx_state <= idle;
                    rx_data <= rx_reg;
                end if;
            when others =>
                rx_state <= idle;
        end case;
    end if;
end process;
end generate;

genlabel2: if oversampling > 1 generate
RX_FSM: process(rx_clk)
begin
    if falling_edge(rx_clk) then
        rx_d1 <= rx_in; -- synchronize the input through two flip-flops
        rx_d2 <= rx_d1;
        rx_data_ready <= '0';
        i <= 0;
        samplecount <= 0;
        case rx_state is
            when idle =>
                if rx_d2 = '0' then
                    rx_state <= delay;
                    samplecount <= samplecount + 1;
                end if;

            when delay =>
                samplecount <= samplecount + 1;
                if samplecount = oversampling-1 then
                    rx_state <= get_data;
                    samplecount <= 0;
                end if;

            when get_data =>
                i <= i;
                samplecount <= samplecount + 1;
                if i < 7 then
                    if samplecount = oversampling/2 then --welcome to the middle of the bit!
                        rx_reg(i) <= rx_d2;
                    end if; -- This looks ugly, but it is neccessary to support 2x oversampl
                    if samplecount = oversampling-1 then
                        i <= i + 1;
                        samplecount <= 0;
                    end if;
                else
                    if samplecount = oversampling/2 then
                        rx_reg(i) <= rx_d2;
                    end if;
                    if samplecount = oversampling-1 then
                        rx_data_ready <= '1';
                        rx_state <= idle;
                        rx_data <= rx_reg;
                        samplecount <= 0;
                    end if;
                end if;
            when others =>
                rx_state <= idle;
        end case;
    end if;
end process;
end generate;


TX_FSM: process(tx_clk)
begin
    if rising_edge(tx_clk) then
        tx_ready <= '1';
        j <= 0;
        tx_out <= '1';
        case tx_state is
```

```
            when idle =>
                if tx_load /= old_load then
                    old_load <= tx_load;
                    reg <= tx_data;
                    tx_state <= send_data;
                    tx_ready <= '0';
                    tx_out <= '0';
                end if;

            when send_data =>
                tx_ready <= '0';
                if j < width-1 then
                    tx_out <= reg(j);
                    j <= j + 1;
                    tx_state <= send_data;
                elsif j = width-1 then
                    tx_out <= reg(j);
                    tx_state <= idle;
                end if;

            when others =>
                tx_state <= idle;
        end case;
    end if;
end process;


end architecture;
```



(a) The UART TX state machine.



(b) The UART RX state machine.

Figure B.1: ASM diagrams for the UART module state machines. This illustration is without the oversampling.

B.2 DAQ

B.2.1 DAQ Top level

Listing: Top level and encoder interface code

```
-- Author        : Bent Furevik
-- Company       : University of Oslo
-- File name     : top.vhd
-- Date          : 17.01.2013
-- Project       : SRADS2
-- Function      : Top-level of SRADS2
-- Simulation pass: N/A
-- On-chip test pass: N/A

-- Encoder interface has been written by Tore André Bekkeng


library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.numeric_std.all;

use ieee.fixed_float_types.all;
use ieee.fixed_pkg.all;
use ieee.float_pkg.all;


entity top is
  port (
    osc_clk      : in std_logic;
    LED          : buffer std_logic_vector(1 downto 0);

    -- magnetometers
    MAG_SCLK     : out std_logic;
    SDI          : in  std_logic;
    SDO          : out std_logic;
    SOC          : out std_logic_vector(5 downto 0);
    DRDY         : out std_logic_vector(5 downto 0);
    CS           : out std_logic_vector(5 downto 0);

    -- External magnetometers
    MAG_TX       : out std_logic;
    MAG_RX       : in  std_logic;
    MAG_RESET    : out std_logic; --extra control signal

    -- STIM210
    STIM210_RX   : in  std_logic;
    STIM210_TX   : out std_logic;

    -- DSS2
    S1_TX        : out std_logic; --extra control signal
    S1_RX        : in  std_logic;
    S1_CLK       : out std_logic;
    S2_TX        : out std_logic; --extra control signal
    S2_RX        : in  std_logic;
    S2_CLK       : out std_logic;

    -- Interface, encoder
    data         : out std_logic;
    not_minf     : in  std_logic;
    not_majf     : in  std_logic;
    not_gate     : in  std_logic;
    not_sclk     : in  std_logic

  );
end entity; -- top

architecture arch of top is

signal dss_treshold : integer := 5000;

-- CONFIGURATION CONSTANT FOR OFFSET BETWEEN POWER SUPPLY AND OPAMP BIAS/REFERENCE VOLTAGE
-- i_offset = ((Vsupply-Vref)/Vsupply)*65535
-- If reference = supply voltage, set i_offset to 0.
signal i_offset : integer := 16729; -- Supply=5.5V, Reference=4.096V
--signal i_offset : integer := 22282; -- Supply=5V, Reference=3.3V

signal offset : std_logic_vector(15 downto 0) := std_logic_vector(to_unsigned(i_offset, 16));


---------- SIGNALS FOR TORE ANDRÉ BEKKENG'S ICI ENCODER INTERFACE ----------
-- Signals from TM encoder
signal sclk             : std_logic;
signal minf             : std_logic;
signal gate             : std_logic;
signal majf             : std_logic;

-- Internal reset signals
signal reset            : std_logic;
signal reset_sync       : std_logic;    -- Synchronize to TM-encoder
signal reset_sync_majf  : std_logic;    -- Synchronize to TM-encoder

-- Internal TM counters
signal bit_cnt          : std_logic_vector(2 downto 0);
signal WordStrobe       : std_logic;
signal word_cnt         : std_logic_vector(7 downto 0);
signal frame_cnt        : std_logic_vector(1 downto 0);

signal gate_delayed     : std_logic;                          -- To not loose MSB in TM
signal DataSample       : std_logic_vector(16-1 downto 0);    -- Data to TM Shift reg.
signal load_sample      : std_logic;
signal load_sample_del  : std_logic;                          -- Enable/disable TM shift reg

-- Internal signals
signal conv             : std_logic;                          -- Signal to start ADC conversio

-- Type definitions
type statetype1 is (SyncCheck, IgnoreSt);
signal sync_state : statetype1;

type statetype2 is (SyncCheck_majf, IgnoreSt_majf);
signal sync_state_majf : statetype2;

component SR_par2ser_redge is
  generic (
    width : integer := 16);
  port (
    clk      : in  std_logic;
    DataIn   : in  std_logic_vector(width-1 downto 0);
    load     : in  std_logic;
```

```
        shift_en  : in  std_logic;
        reset     : in  std_logic;
        DataOut   : out std_logic);
    end component;
    --------------------------------------------------------------------------------

    signal STIM210_DataOut       : std_logic_vector(135 downto 0);
    signal STIM210_DataReady     : std_logic;
    alias STIM210_DataOutX       : std_logic_vector(23 downto 0) is STIM210_DataOut(135 downto 112);
    alias STIM210_DataOutY       : std_logic_vector(23 downto 0) is STIM210_DataOut(111 downto 88);
    alias STIM210_DataOutZ       : std_logic_vector(23 downto 0) is STIM210_DataOut(87
downto 64);
    alias STIM210_DataStatus     : std_logic_vector(7  downto 0) is STIM210_DataOut(63
downto 56);
    alias STIM210_DataTempX      : std_logic_vector(15 downto 0) is STIM210_DataOut(55
downto 40);
    alias STIM210_DataTempY      : std_logic_vector(15 downto 0) is STIM210_DataOut(39
downto 24);
    alias STIM210_DataTempZ      : std_logic_vector(15 downto 0) is STIM210_DataOut(23
downto 8);
    alias STIM210_DataCRC        : std_logic_vector(7  downto 0) is STIM210_DataOut(7
downto 0);

    component uart_module is
        generic(
            width : integer := 8;
            oversampling : integer := 4
            );
        port(
            rx_clk : in std_logic; -- rx_clk must have a frequency 4 times higher than baud rate if 4x o
            rx_in : in std_logic;
            rx_data : out std_logic_vector(width-1 downto 0);
            rx_data_ready : out std_logic;

            tx_clk : in std_logic;
            tx_out : out std_logic;
            tx_data : in std_logic_vector(width-1 downto 0);
            tx_load : in std_logic;
            tx_ready : out std_logic
        );
    end component;

    ---------------- CLOCK DIVIDERS AND PLL'S ----------------

    component ClockDiv is
        generic (
            factor : integer
        );
        port(
            sclk  : in     std_logic;
            clk   : buffer std_logic
        );
    end component;

    component alt_pll_921k6 IS
        PORT
        (
            inclk0        : IN STD_LOGIC  := '0';
            c0            : OUT STD_LOGIC;
            locked        : OUT STD_LOGIC
        );
    END component;
```

```
component alt_pll_3M6864 IS
    PORT
    (
        inclk0        : IN STD_LOGIC  :=  '0';
        c0        : OUT STD_LOGIC;
        locked        : OUT STD_LOGIC
    );
END component;

component alt_pll_4M0 IS
    PORT
    (
        inclk0        : IN STD_LOGIC  :=  '0';
        c0        : OUT STD_LOGIC ;
        locked        : OUT STD_LOGIC
    );
END component;

component alt_pll_460k8
    PORT
    (
        inclk0        : IN STD_LOGIC  :=  '0';
        c0        : OUT STD_LOGIC ;
        locked        : OUT STD_LOGIC
    );
end component;


signal clk_921k6              : std_logic;
signal pll_locked_921k6       : std_logic;
signal clk_3M6864             : std_logic;
signal pll_locked_3M6864      : std_logic;
signal clk_4M0                : std_logic;
signal pll_locked_4M0         : std_logic;
signal clk_460k8              : std_logic;
signal pll_locked_460k8       : std_logic;
signal clkdiv : std_logic;

------------ SENSORCARD SIGNALS ------------
signal s1_dataready           : std_logic;
signal S1data                 : std_logic_vector(63 downto 0);
alias S1data_ch1              : std_logic_vector(15 downto 0) is S1data(63 downto 48);
alias S1data_ch2              : std_logic_vector(15 downto 0) is S1data(47 downto 32);
alias S1data_ch3              : std_logic_vector(15 downto 0) is S1data(31 downto 16);
alias S1data_ch4              : std_logic_vector(15 downto 0) is S1data(15 downto 0);
signal S1_Xdata               : std_logic_vector(15 downto 0);
signal S1_Ydata               : std_logic_vector(15 downto 0);
signal DSS2_1_Xdata           : std_logic_vector(15 downto 0);
signal DSS2_1_Ydata           : std_logic_vector(15 downto 0);
signal DataReadyS1            : std_logic;

------------ READY DATA WORDS ------------
signal Gyro_Roll_t        : std_logic_vector(15 downto 0);
signal Gyro_Pitch_t       : std_logic_vector(15 downto 0);
signal Gyro_Yaw_t         : std_logic_vector(15 downto 0);
signal Gyro_HK_t          : std_logic_vector(15 downto 0);

------------ READY DATA WORDS ------------
signal DSS2_Xdata         : std_logic_vector(15 downto 0);
signal DSS2_Ydata         : std_logic_vector(15 downto 0);
signal Gyro_Roll          : std_logic_vector(15 downto 0);
signal Gyro_Pitch       : std_logic_vector(15 downto 0);
signal Gyro_Yaw         : std_logic_vector(15 downto 0);
signal Gyro_HK          : std_logic_vector(15 downto 0);
signal Mag_Int_X        : std_logic_vector(15 downto 0);
signal Mag_Int_Y        : std_logic_vector(15 downto 0);
signal Mag_Int_Z        : std_logic_vector(15 downto 0);
signal Mag_Ext_X        : std_logic_vector(15 downto 0);
signal Mag_Ext_Y        : std_logic_vector(15 downto 0);
signal Mag_Ext_Z        : std_logic_vector(15 downto 0);



component general_interface is
    generic(
        identifier : std_logic_vector(7 downto 0) := x"0A";
        data_words : integer := 8;
        word_length : integer := 8;
        oversampling : integer := 4
        );
    port (
        mclk         : in  std_logic;
        uart_clk_x   : in  std_logic;
        RXDATA       : in  std_logic;
        DataReady    : out std_logic;
        DataOut      : out std_logic_vector((data_words*word_length)-1 downto 0);
        Debug : out std_logic_vector(1 downto 0)
    );
end component;

signal ExtMag_DataReady : std_logic;
signal ExtMag_DataOut   : std_logic_vector(47 downto 0);
alias  ExtMag_X is ExtMag_DataOut(47 downto 32);
alias  ExtMag_Y is ExtMag_DataOut(31 downto 16);
alias  ExtMag_Z is ExtMag_DataOut(15 downto 0);

signal IntMag_DataReady : std_logic;
signal IntMag_DataOut   : std_logic_vector(47 downto 0);
alias  IntMag_X is IntMag_DataOut(47 downto 32);
alias  IntMag_Y is IntMag_DataOut(31 downto 16);
alias  IntMag_Z is IntMag_DataOut(15 downto 0);


signal test_tx_data          : std_logic_vector(7 downto 0);
signal test_rx_data          : std_logic_vector(7 downto 0);
signal test_tx_ready         : std_logic;
signal test_rx_data_ready    : std_logic;
signal test_tx_load          : std_logic;
signal lock, s1_lock, s1_tx_lock : std_logic;

function comp2_to_sign_magnitude(data : std_logic_vector) return std_logic_vector is
variable tmp : std_logic_vector(data'range);
begin
    if data(data'high) = '1' then
        tmp := not(data - 1);
        tmp(tmp'high) := '1';
        return tmp;
    else
        return data;
    end if;
end comp2_to_sign_magnitude;
```

```
signal data_l0, data_l1, data_l2, data_l3 : sfixed(15 downto 0);
signal stotal, total : sfixed(18 downto 0);

signal nomX, nomY : sfixed(18 downto 0);

signal X_l  : sfixed((nomX'left-stotal'right)+1 downto nomX'right-stotal'left);
signal Y_l  : sfixed((nomY'left-stotal'right)+1 downto nomY'right-stotal'left);
signal X_l2 : sfixed(0 downto -15);
signal Y_l2 : sfixed(0 downto -15);

signal X, Y : std_logic_vector(15 downto 0);

signal Gyro_X_test : std_logic_vector(15 downto 0);
signal Gyro_Y_test : std_logic_vector(15 downto 0);
signal Gyro_Z_test : std_logic_vector(15 downto 0);
signal Gyro_hk_test : std_logic_vector(15 downto 0);

begin

PLL_rx: alt_pll_3M6864
PORT map (
    inclk0 => osc_clk,
    c0 => clk_3M6864,
    locked => pll_locked_3M6864
);

PLL_DSS_rx: alt_pll_4M0
PORT map (
    inclk0 => osc_clk,
    c0 => clk_4M0,
    locked => pll_locked_4M0
);

PLL_mag : alt_pll_460k8
PORT MAP (
    inclk0   => osc_clk,
    c0   => clk_460k8,
    locked => pll_locked_460k8
);


clk_uart: ClockDiv
generic map(100) -- 20MHz to 200KHz
port map(sclk => osc_clk, clk => clkdiv);

--------------------------------------------------------------------
-- SENSOR INTERFACES
--------------------------------------------------------------------

DSS2_1_interface: general_interface
generic map(identifier => x"0A", data_words => 8, word_length => 8, oversampling => 4)
port map (osc_clk, clk_4M0, S1_RX, s1_dataready, S1data, open);

STIM210_interface: general_interface
generic map(identifier => x"A0", data_words => 17, word_length => 8, oversampling => 4)
port map (osc_clk, clk_3M6864, STIM210_RX, STIM210_DataReady, STIM210_DataOut, open);

ExtMag_interface: general_interface
generic map(identifier => x"4D", data_words => 6, word_length => 8, oversampling => 4)
port map (osc_clk, clk_460k8, MAG_RX, ExtMag_DataReady, ExtMag_DataOut, LED);
MAG_RESET <= '0';
```

```
--------------------------------------------------------------------------------
--                     ENCODER INTERFACE BY TORE ANDRÉ BEKKENG

-- Modified by Bent Furevik:
-- 16-bit words can now be added after each other, without having to concatinate them.
-- This modification makes the setup slightly more verbose, while saving a few logic elements and in
--------------------------------------------------------------------------------

-- Data transfer shift register
Com_SR: SR_par2ser_redge
generic map (16)
port map (sclk, DataSample, load_sample_del, gate_delayed, reset, data);

-- Invert again the signals inverted in the optocouplers
sclk <= not(not_sclk);
minf <= not(not_minf);
majf <= not(not_majf);
gate <= not(not_gate);

--------------------------------------------------------------------------------
--                     TEST DATA MAGNETOMETER CHANNELS
--------------------------------------------------------------------------------
process(minf)
begin
    if rising_edge(minf) then
        IntMag_X <= IntMag_X + 1;
        IntMag_Y <= IntMag_Y + 1;
        IntMag_Z <= IntMag_Z + 1;
    end if;
end process;
--------------------------------------------------------------------------------
--                     DSS2 CONVERSION CONTROL
--------------------------------------------------------------------------------
-- Control of the DSS2's

--DSS2_CONVERSION:
process(sclk)
begin
    if rising_edge(sclk) then
        if ((word_cnt <= 32 and word_cnt >= 22)  or (word_cnt <= 80 and word_cnt >= 70)) then
            S1_TX <= '1';
            S2_TX <= '1';
        else
            S1_TX <= '0';
            S2_TX <= '0';
        end if;
    end if;
end process;

--------------------------------------------------------------------------------
--                     DATA HANDLING / STORAGE
--------------------------------------------------------------------------------

total <= resize(data_l0 +data_l1 + data_l2 + data_l3, total);
stotal <= total;

nomX <= resize((data_l0+ data_l1) - (data_l2 + data_l3), nomX);

nomY <= resize((data_l0+ data_l2) - (data_l1 + data_l3), nomY);
```

```
X_l <= resize((nomX/stotal), X_l);
Y_l <= resize((nomY/stotal), Y_l);
X_l2 <= resize(X_l, X_l2);
Y_l2 <= resize(Y_l, Y_l2);

DSS2_1_Xdata <= to_slv(X_l2);
DSS2_1_Ydata <= to_slv(Y_l2);

--DATA_SHIFTING:
process(sclk)
variable temp : std_logic_vector(15 downto 0);
begin
    if rising_edge(sclk) then
        if (s1_dataready = '1') then
            temp := S1data_ch3 - offset;
            data_l0 <= resize(to_sfixed("000"&temp(15 downto 3), data_l0), data_l0);
            temp := S1data_ch4 - offset;
            data_l1 <= resize(to_sfixed("000"&temp(15 downto 3), data_l1), data_l1);
            temp := S1data_ch1 - offset;
            data_l2 <= resize(to_sfixed("000"&temp(15 downto 3), data_l2), data_l2);
            temp := S1data_ch2 - offset;
            data_l3 <= resize(to_sfixed("000"&temp(15 downto 3), data_l3), data_l3);
        end if;

        if (frame_cnt = 0 and word_cnt < 71) or (frame_cnt = 3 and word_cnt > 77) then
            if STIM210_DataReady = '1' then
                Gyro_Roll     <= comp2_to_sign_magnitude(STIM210_DataOutX)(23 downto 8);
                Gyro_Pitch    <= comp2_to_sign_magnitude(STIM210_DataOutY)(23 downto 8);
                Gyro_Yaw      <= comp2_to_sign_magnitude(STIM210_DataOutZ)(23 downto 8);
                Gyro_HK       <= x"00" & STIM210_DataStatus;
            end if;

            -- if IntMag_DataReady = '1' then
                Mag_Int_X <= IntMag_X;
                Mag_Int_Y <= IntMag_Y;
                Mag_Int_Z <= IntMag_Z;
            -- end if;

            if ExtMag_DataReady = '1' then
                Mag_Ext_X <= ExtMag_X;
                Mag_Ext_Y <= ExtMag_Y;
                Mag_Ext_Z <= ExtMag_Z;
            end if;
        end if;
    end if;
end process;

--------------------------------------------------------------------------------
--                              WORD STROBE
--------------------------------------------------------------------------------

BITCOUNTER:
process(sclk, reset_sync)
begin
    if reset_sync = '1' then            -- asyncron
        bit_cnt <= (others => '0');
    elsif falling_edge(sclk) then
        bit_cnt <= bit_cnt + 1;
    end if;
end process;

WORDSYNC:
```

```
process(sclk)
begin
    if rising_edge(sclk) then
        if (bit_cnt = 7) then
            WordStrobe <= '1';
        else
            WordStrobe <= '0';
        end if;
    end if;
end process;

--------------------------------------------------------------------------------
--                          WORD COUNTER
--------------------------------------------------------------------------------

-- 144 words per frame

WORDCOUNTER:
process(sclk, reset_sync)
begin
    if (reset_sync = '1') then
        word_cnt <= (others  => '0');
    elsif falling_edge(sclk) then
        if (WordStrobe = '1') then
            if (word_cnt < 143) then
                word_cnt <= word_cnt + 1;
            else
                word_cnt <= (others => '0');
            end if;
        end if;
    end if;
end process;

--------------------------------------------------------------------------------
--                          FRAMECOUNTER
--------------------------------------------------------------------------------

FRAMECOUNTER:
process(sclk, reset_sync_majf)
begin
    if (reset_sync_majf = '1') then
        frame_cnt <= (others  => '0');
    elsif rising_edge(sclk) then
        if (word_cnt = 143 and bit_cnt = 0) then
            frame_cnt <= frame_cnt + 1;
        end if;
    end if;
end process;

--------------------------------------------------------------------------------
--                          FSM  SYNCRONIZERS
--------------------------------------------------------------------------------

-- Synchronize / reset bit & word counter to minf from PCM encoder

-- Synchronize / reset bit & word counter to minf and majf from encoder
-- Creates a short reset pulse. For the reset pulse to be short compared to a
-- clock period from the system (encoder) clock, the PCB oscillator clock
-- frequency must be considerably higher that the system clock frequency. A ratio
-- where the PCB oscillator is 6 timer higher than the system clock has been tested
-- and verified to work without glitches.
```

```
FSM_MINF_MAJF_SYNC:
process(osc_clk)
begin
    if rising_edge(osc_clk) then
        case sync_state is
            when SyncCheck =>
                if (minf = '1' or majf = '1') then
                    reset_sync <= '1';
                    sync_state <= IgnoreSt;
                else
                    reset_sync <= '0';
                    sync_state <= SyncCheck;
                end if;

            when IgnoreSt =>
                reset_sync <= '0';
                if (minf = '0' and majf = '0') then
                    sync_state <= SyncCheck;
                else
                    sync_state <= IgnoreSt;
                end if;
        end case;
    end if;
end process;
-- Synchronize / reset frame counter to majf from encoder. Implement if MAJF is in use

FSM_MAJF_SYNC:
process(osc_clk)
begin
  if rising_edge(osc_clk) then
    case sync_state_majf is
      when SyncCheck_majf =>
        if (majf = '1') then
          reset_sync_majf <= '1';
          sync_state_majf <= IgnoreSt_majf;
        else
          reset_sync_majf <= '0';
          sync_state_majf <= SyncCheck_majf;
        end if;

       when IgnoreSt_majf =>
        reset_sync_majf <= '0';

         if (majf = '0') then
           sync_state_majf <= SyncCheck_majf;
         else
           sync_state_majf <= IgnoreSt_majf;
         end if;
    end case;
  end if;
end process;

--------------------------------------------------------------------------------
--                              GATE DELAY
--------------------------------------------------------------------------------
-- Delay of the gate signal, to avoid losing msb

GATEDELAY:
process(sclk)
begin
    if rising_edge(sclk) then
        if(gate = '1') then
```

```
            gate_delayed <= '1';
        else
            gate_delayed <= '0';
        end if;
    end  if;
end process;

--------------------------------------------------------------------------------
--                      DATA COMMUNICATION WITH ENCODER
--------------------------------------------------------------------------------

-- load data into the SR in the word before transfer
-- First transmission of data in current minf is from last conversion in previous minf
DATA_TRANSFER:
process(sclk)
begin
    if rising_edge(sclk) then
        load_sample <= '0';

        -- Word1+Word2 into 32-bit SR
        if bit_cnt = 7 then
            if (word_cnt = 23 or word_cnt = 95) then
                if data_l0+data_l1+data_l2+data_l3 < dss_treshold then
                    DataSample(15 downto 0) <= (others => '0');
                else
                    DataSample(15 downto 0)      <= DSS2_1_Xdata;      --DSS2 X
                end if;
                load_sample <= '1';

            elsif (word_cnt = 25 or word_cnt = 97) then
                if data_l0+data_l1+data_l2+data_l3 < dss_treshold then
                    DataSample(15 downto 0) <= (others => '0');
                else
                    DataSample(15 downto 0)      <= DSS2_1_Ydata;      --DSS2 Y
                end if;
                load_sample <= '1';

            elsif (word_cnt = 71) then
                load_sample <= '1';
                if frame_cnt = 0 then
                    DataSample(15 downto 0)      <= Gyro_Roll;     --GYRO roll
                elsif frame_cnt = 1 then
                    DataSample(15 downto 0)      <= Gyro_Pitch;   --GYRO pitch
                elsif frame_cnt = 2 then
                    DataSample(15 downto 0)      <= Gyro_Yaw;      --GYRO yaw
                elsif frame_cnt = 3 then
                    DataSample(15 downto 0)      <= Gyro_HK;       --GYRO HK
                end if;

            elsif (word_cnt = 73) then
                load_sample <= '1';
                if frame_cnt = 0 then
                    DataSample(15 downto 0)      <= Mag_Int_X;    --Internal Mag X
                elsif frame_cnt = 1 then
                    DataSample(15 downto 0)      <= Mag_Int_Y;    --Internal Mag Y
                elsif frame_cnt = 2 then
                    DataSample(15 downto 0)      <= Mag_Int_Z;    --Internal Mag Z
                elsif frame_cnt = 3 then
                    DataSample(15 downto 0)      <= x"0000";       --Zero fill
                end if;

            elsif (word_cnt = 75) then
```

```
                    load_sample <= '1';
                    if frame_cnt = 0 then
                        DataSample(15 downto 0)     <= Mag_Ext_X;   --External Magnetometer X
                    elsif frame_cnt = 1 then
                        DataSample(15 downto 0)     <= Mag_Ext_Y;   --External Magnetometer Y
                    elsif frame_cnt = 2 then
                        DataSample(15 downto 0)     <= Mag_Ext_Z;   --External Magnetometer Z
                    elsif frame_cnt = 3 then
                        DataSample(15 downto 0)     <= x"0000";     --Zero fill
                    end if;
                end if;


            end if;
        end if;
end process;


-- Delay the load signal to SR with one half clock periode,
-- because data are transmitted from the shift register on the
-- rising edge, the same flank as the data transfer process uses.
LOAD_DELAY:
process(sclk)
begin
    if falling_edge(sclk) then
        if (load_sample = '1') then
            load_sample_del <= '1';
        else
            load_sample_del <= '0';
        end if;
    end if;
end process;

end arch;
```

Parallel to serial shiftregister

Listing: Top level and encoder interface code

```
-- Author      : Tore André Bekkeng
-- Company     : University of Oslo
-- File name   : SR_par2ser_redge.vhd
-- Date        : 08.09.2010
-- Version     : 1
-- Project     : ICI-3 2011
-- Function    : Parallel to serial skift register, rising edge




library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity SR_par2ser_redge is
  generic (
    width : integer := 16);
  port (
    clk      : in  std_logic;
    DataIn   : in  std_logic_vector(width-1 downto 0);
    load     : in  std_logic;
    shift_en : in  std_logic;
    reset    : in  std_logic;
    DataOut  : out std_logic);

end SR_par2ser_redge;

architecture SR_par2ser_arch of SR_par2ser_redge is

  signal data_int : std_logic_vector(width-1 downto 0);


begin

  SHIFT_REG: process (clk, reset)
  begin

    if reset = '1' then                       -- asynchronous reset
      data_int <= (others => '0');

    elsif rising_edge(clk) then
      if (load = '1') then
        data_int <= DataIn;

      elsif (shift_en = '1') then
        for i in width-2 downto 0 loop
          data_int(i+1) <= data_int(i);
        end loop;
      end if;

    end if;
  end process SHIFT_REG;

  DataOut <= data_int(width-1);



end SR_par2ser_arch;
```

B.2.2 General Interface Module

Listing: Top level and encoder interface code

```
-- Author      : Bent Furevik
-- Company     : University of Oslo
-- File name   : general_interface.vhd
-- Date        : 13.04.2013
-- Project     : SRADS2
-- Function    : General interface for recieving data from UART-based sensors


library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.std_logic_unsigned.all;

entity general_interface is
```

```
    generic(
        identifier : std_logic_vector(7 downto 0) := x"0A";
        words : integer := 8;
        length : integer := 8;
        oversampling : integer := 4
        );
    port (
        mclk         : in  std_logic;
        uart_clk_x   : in  std_logic; -- must be X times higher than the actual uart_clk
        RXDATA       : in  std_logic;
        DataReady    : out std_logic;
        DataOut      : out std_logic_vector((words*length)-1 downto 0);
        Debug        : out std_logic_vector(1 downto 0)
    );
end general_interface;

architecture arch of general_interface is

component uart_module is
    generic(
        width : integer := length;
        oversampling : integer := oversampling
        );
    port(
        rx_clk : in std_logic;
        rx_in : in std_logic;
        rx_data : out std_logic_vector(width-1 downto 0);
        rx_data_ready : out std_logic;
        tx_clk : in std_logic;
        tx_out : out std_logic;
        tx_data : in std_logic_vector(width-1 downto 0);
        tx_load : in std_logic;
        tx_ready : out std_logic
    );
end component;

signal tx_data : std_logic_vector(length-1 downto 0);
signal rx_data : std_logic_vector(length-1 downto 0);
signal rx_data_ready : std_logic;


type state_type is (pre_idle, pre_idle2, idle, get_data);
signal state : state_type;

signal i : integer range 0 to words;
signal out_reg : std_logic_vector((words*length)-1 downto 0);
type   rx_reg_type is array(words-1 downto 0) of std_logic_vector(length-1 downto 0);
signal rx_reg : rx_reg_type;

signal lock : std_logic;


begin

UART: uart_module
generic map(
    width => 8,
    oversampling => oversampling
)
port map(
    rx_clk => uart_clk_x,
    rx_in => RXDATA,
    rx_data => rx_data,
    rx_data_ready => rx_data_ready,
    tx_out => open,
    tx_clk => '0',
    tx_data => (others => '0'),
    tx_load => '0',
    tx_ready => open
);


FSM: process(mclk, state, rx_data_ready)
begin
    if rising_edge(mclk) then
        i <= words-1;
        DataReady <= '0';
        case state is
            when pre_idle =>
                DataReady <= '1';
                Debug <= "01";
                lock <= rx_data_ready;
                if rx_data_ready = '1' and lock = '0' then
                -- if new_data_ready then -- since new_data_ready_s1 is an impure function returning
                    if rx_data = x"0D" then
                        state <= pre_idle2;
                    end if;
                end if;

            when pre_idle2 =>
                Debug <= "10";
                DataReady <= '1';
                lock <= rx_data_ready;
                if rx_data_ready = '1' and lock = '0' then
                    if rx_data = x"0A" then
                        state <= idle;
                    end if;
                end if;

            when idle =>
                Debug <= "11";
                DataReady <= '1';
                lock <= rx_data_ready;
                if rx_data_ready = '1' and lock = '0' then
                    Debug <= "10";
                    if rx_data = identifier then -- we can now be sure that this is the start of the
                        state <= get_data;
                    end if;
                end if;

            when get_data =>
                i <= i;
                Debug <= "00";
                lock <= rx_data_ready;
                if rx_data_ready = '1' and lock = '0' then
                    rx_reg(i) <= rx_data;
                    if i = 0 then
                        state <= pre_idle;
                        DataReady <= '1';

                        for j in 1 to words loop
                            out_reg((j*length)-1 downto (j-1)*length) <= rx_reg(j-1);
                        end loop;
                    else
```

```
                              i <= i - 1;
                          end if;
                      end if;

                  when others =>
                      state <= pre_idle;
              end case;
          end if;
      end process; -- FSM

      DataOut <= out_reg;

      end architecture arch; -- arch
```

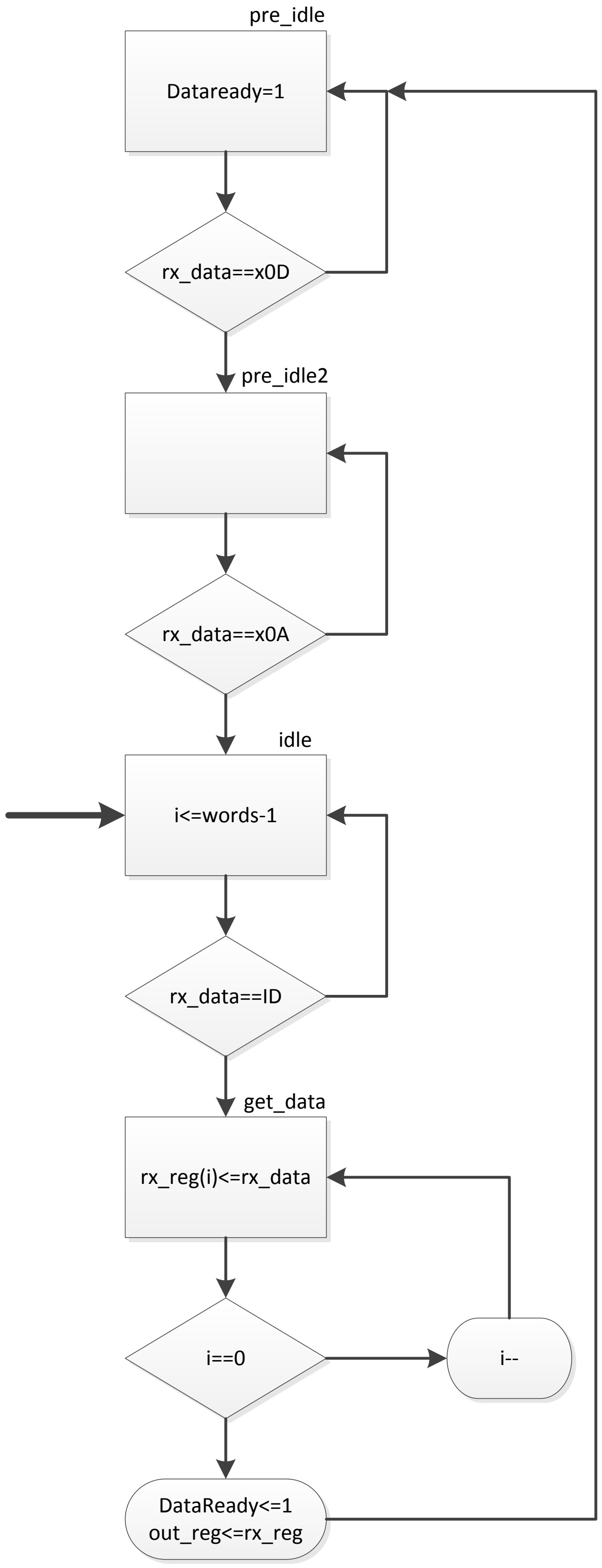


Figure B.2: ASM diagram for the general interface module state machine. The lock bits stopping the state machine from reading multiple copies of the same byte is removed from this diagram.

B.3 DSS2

B.3.1 Clock Divider

Listing: Top level and encoder interface code

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity ClockDiv is
    generic (
        factor : integer := 20
    );
    port (
        sclk  : in     std_logic;
        clk   : buffer std_logic := '0'
    );
end ClockDiv;

architecture main of ClockDiv is

signal div : integer range 0 to (factor/2)+1 := 0;
begin

CLOCK_DIVIDER: process (sclk)
-- variable div : integer range 1 to factor;
begin
    if rising_edge(sclk) then
        div <= div + 1;
        if div = factor/2 then
            if clk = '1' then
                clk <= '0';
            else
                clk <= '1';
            end if;
            div <= 1;
        end if;
    end if;
end process;

end architecture;
```

B.3.2 DSS2 Top level

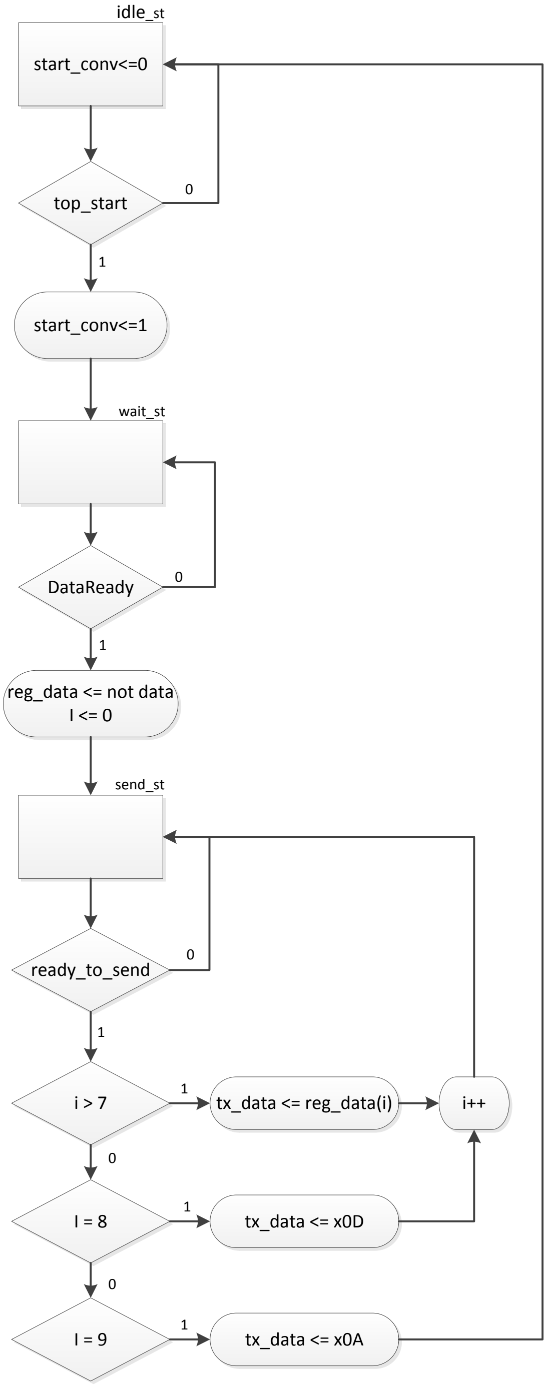


Figure B.3: ASM diagram for the DSS2 top level state machine.

Listing: Top level and encoder interface code

```
-- Author        : Bent Furevik
-- Company       : University of Oslo
-- File name     : top.vhd
-- Date          : 25.11.2012
-- Project       : SRADS2/DSS2
-- Function      : DSS2 top module
-- Simulation pass: Yes
-- On-chip test pass: Yes

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.std_logic_unsigned.all;

entity top is
    port(
        -- -- -- Global
        mclk      : in  std_logic; --crystal 20MHz (rev B, montering 2)

        -- -- -- ADC
        SDATA     : in std_logic_vector(3 downto 0);
        CS        : out std_logic_vector(3 downto 0);
        SCLK      : out std_logic;

        -- -- -- Interface to mainboard
        rx_in     : in std_logic;
        tx_out    : out std_logic;
        uart_clk  : in std_logic -- sync clk


    );
end top;

architecture arch of top is

component ad7680 is
    port(
        start_conv : in std_logic;
        SCLK : in std_logic;
        CS : out std_logic; -- active low
        SDATA : in std_logic;
        DataOut : out std_logic_vector(16-1 downto 0);
        DataReady : out std_logic
    );
end component;


signal data_CH0, data_CH1, data_CH2, data_CH3 : std_logic_vector(16-1 downto 0);
signal reg_data_CH0, reg_data_CH1, reg_data_CH2, reg_data_CH3 : std_logic_vector(16-1 downto
signal total, A, B, C, D, X, Y : std_logic_vector(16-1 downto 0);

signal DataReady : std_logic_vector(3 downto 0);

component uart_module is
    generic(
        width : integer := 8
        );
    port(
        rx_clk : in std_logic;
        rx_in : in std_logic;
        rx_data : out std_logic_vector(width-1 downto 0);
        rx_data_ready : out std_logic;

        tx_clk : in std_logic;
        tx_out : out std_logic;
        tx_data : in std_logic_vector(width-1 downto 0);
        tx_load : in std_logic;
        tx_ready : out std_logic
    );
end component;


signal tx_data       : std_logic_vector(7 downto 0);
signal rx_data       : std_logic_vector(7 downto 0);
signal tx_ready      : std_logic;
signal rx_data_ready : std_logic;


component ClockDiv is
    generic (
        factor : integer
    );
    port(
        sclk : in     std_logic;
        clk  : buffer std_logic
    );
end component;

signal adc_sclk : std_logic;

signal start_conv : std_logic;

type state_type is (idle_st, wait_st, send_st);
signal state : state_type;

signal uart_divclk : std_logic;
signal rx_command : std_logic_vector(tx_data'range);

signal i : integer;

signal tx_load : std_logic;

signal tx_lock : std_logic;
impure function ready_to_send return boolean is
begin
    tx_lock <= tx_ready;
    return ((tx_ready xor tx_lock) and tx_ready) = '1';
end ready_to_send;


begin

clk_uart: ClockDiv
generic map(20) -- 20MHz to 1MHz
port map(sclk => mclk, clk => uart_divclk);

clk_adc: ClockDiv
generic map(10) -- 20MHz to 2MHz
port map(sclk => mclk, clk => adc_sclk);

SCLK <= adc_sclk;
```

```
adc_ch0: ad7680
port map (start_conv, adc_sclk, CS(0), SDATA(0), data_CH0, DataReady(0));

adc_ch1: ad7680
port map (start_conv, adc_sclk, CS(1), SDATA(1), data_CH1, DataReady(1));

adc_ch2: ad7680
port map (start_conv, adc_sclk, CS(2), SDATA(2), data_CH2, DataReady(2));

adc_ch3: ad7680
port map (start_conv, adc_sclk, CS(3), SDATA(3), data_CH3, DataReady(3));

INTERFACE: uart_module
port map(
    rx_clk => '0',
    rx_in => '0',
    rx_data => open,
    rx_data_ready => open,

    tx_clk => uart_divclk,
    tx_out => tx_out,
    tx_data => tx_data,
    tx_load => tx_load,
    tx_ready => tx_ready
);



TOP_CTRL:
process(uart_divclk, state, tx_ready)
begin
    if rising_edge(uart_divclk) then
        start_conv <= '0';
        case state is
            when idle_st =>
                state <= idle_st;
                if rx_in = '1' then
                    state <= wait_st;
                    start_conv <= '1';
                    tx_data <= x"0A"; -- \n
                    tx_load <= not tx_load;
                end if;

            when wait_st =>
                state <= wait_st;
                if DataReady(0) = '1' then
                    start_conv <= '0';
                    reg_data_CH0 <= not data_CH0;
                    reg_data_CH1 <= not data_CH1;
                    reg_data_CH2 <= not data_CH2;
                    reg_data_CH3 <= not data_CH3;
                    state <= send_st;
                    i <= 0;
                end if;

            when send_st =>
                state <= send_st;
                if ready_to_send then
                    i <= i + 1;
                    if    (i=0) then
                        tx_data <= reg_data_CH0(15 downto 8);
                        tx_load <= not tx_load;
                    elsif (i=1) then
                        tx_data <= reg_data_CH0(7  downto 0);
                        tx_load <= not tx_load;
                    elsif (i=2) then
                        tx_data <= reg_data_CH1(15 downto 8);
                        tx_load <= not tx_load;
                    elsif (i=3) then
                        tx_data <= reg_data_CH1(7  downto 0);
                        tx_load <= not tx_load;
                    elsif (i=4) then
                        tx_data <= reg_data_CH2(15 downto 8);
                        tx_load <= not tx_load;
                    elsif (i=5) then
                        tx_data <= reg_data_CH2(7  downto 0);
                        tx_load <= not tx_load;
                    elsif (i=6) then
                        tx_data <= reg_data_CH3(15 downto 8);
                        tx_load <= not tx_load;
                    elsif (i=7) then
                        tx_data <= reg_data_CH3(7  downto 0);
                        tx_load <= not tx_load;
                    elsif (i=8) then
                        tx_data <= x"0D"; -- x0A = \n
                        tx_load <= not tx_load;
                    elsif (i=9) then
                        tx_data <= x"0A"; -- x0A = \n
                        tx_load <= not tx_load;
                        state <= idle_st;
                    end if;
                end if;

            when others =>
                state <= idle_st;
        end case;
    end if;
end process;

end arch;
```

B.3.3 AD7680 interface

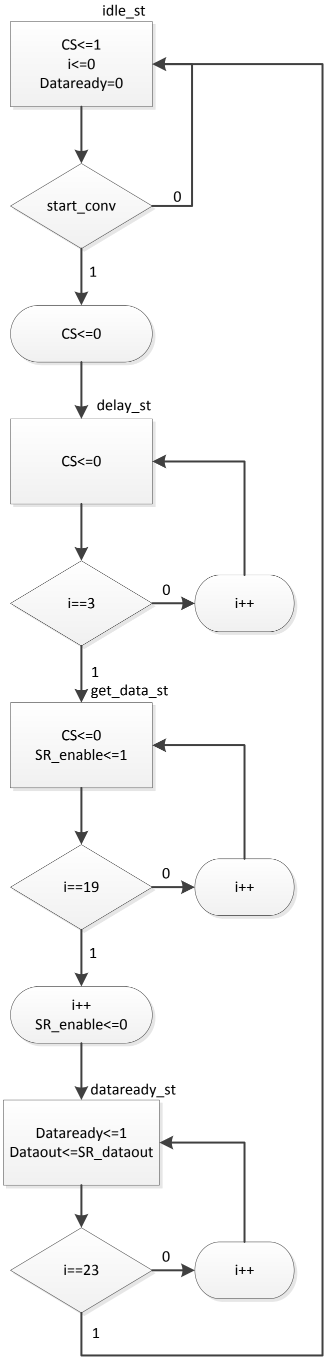


Figure B.4: ASM diagram for the AD7680 control state machine.

Listing: Top level and encoder interface code

```
-- Author          : Bent Furevik
-- Company         : University of Oslo
-- File name       : ad7680.vhd
-- Date            : 21.11.2012
-- Project         : SRADS2/DSS2
-- Function        : Interface for the AD7680 AD-converter
-- Simulation pass: Yes
-- On-chip test pass: Yes


library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;


entity ad7680 is
    port(
        start_conv : in std_logic;
        SCLK : in std_logic;
        CS : out std_logic; -- active low
        SDATA : in std_logic;
        DataOut : out std_logic_vector(16-1 downto 0);
        DataReady : out std_logic
    );
end ad7680;

architecture ad7680_rtl of ad7680 is

component SR_SerIn_redge is
    generic (
        width : integer := DataOut'length
    );
    port
    (
        clk       : in  std_logic;
        DataIn    : in  std_logic;
        shift_en  : in  std_logic;
        DataOut   : out std_logic_vector(width-1 downto 0)
    );
end component;

type state_type is (idle_st, get_st, delay_st, dataready_st);
signal state : state_type;

signal i : integer range 0 to DataOut'left+8;


signal SR_enable   : std_logic;
signal SR_DataOut  : std_logic_vector(DataOut'range);


begin

SR: SR_SerIn_redge
port map(
    clk => SCLK,
    DataIn => SDATA,
    shift_en => SR_enable,
    DataOut => SR_DataOut
);
```

```
FSM: process(SCLK, state)

begin
    if rising_edge(SCLK) then
        SR_enable <= '0';
        i <= 0;
        DataReady <= '0';
        CS <= '1';

        case state is
            when idle_st =>
                if start_conv = '1' then
                    CS <= '0';
                    state <= delay_st;
                    i <= 0;
                end if;

            when delay_st =>
                CS <= '0';
                i <= i + 1;
                if i >= 3 then -- 4 leading zeros
                    state <= get_st;
                end if;

            when get_st =>
                CS <= '0';
                i <= i + 1;
                SR_enable <= '1';
                if i >= DataOut'left+4 then
                    state <= dataready_st;
                    SR_enable <= '0';
                end if;

            when dataready_st =>
                DataReady <= '1';
                DataOut <= SR_DataOut;
                state <= dataready_st;
                if i >= DataOut'left+8 then -- 4 trailing zeros
                    state <= idle_st;
                    i <= 0;
                else
                    i <= i + 1;
                end if;

            when others =>
                state <= idle_st;
        end case;
    end if;
end process;


end ad7680_rtl;
```

Appendix C

Matlab code

C.1 Code for noise estimation

Listing: Noise calculation code

```
clc;
q=1.6e-19; %electronic charge
kb=1.38e-23; %boltzmann
Io=100e-6; %diode output current
Id=1e-9; % dark current
Ib=30e-15; %input bias current
T=20+273.1; %temp
BW=10e3; %bandwidth
Rf=25e3; %feedback resistor
Rie=10e3; %interelectrode resistance
fmax=BW; %Hz
fmin=1;  %Hz
Ki=2e-15; %flicker noise @1Hz
Kv=280e-9; %flicker noise @1Hz
In=0.1e-15; %input current noise density
En=15e-9; %input voltage noise density

Esh=sqrt(2*q*(Io+Id+Ib)*BW) * Rf

E1f=Kv*sqrt(log(fmax/fmin))

Eri=sqrt(4*kb*T*Rie*BW)

Erf=sqrt(4*kb*T*Rf*BW)

Eno=(1+(Rf/Rie))*En*sqrt(15e3)

Ex=In*Rf*sqrt(BW)

Vnout = sqrt(Esh^2 + E1f^2 + Eri^2 + Erf^2 + Ex^2 + Eno^2)

Output = Io*Rf
```

C.2 Code for plotting results

Listing: Plotting and position calculation code

```
clear 'all'
A = importdata('15.06.2013_0637_SunSensor_cp.txt');
cnt = 1;


base_angle = 100;
width = 1;
L=9;
distance=3.6; %distance from pin-hole to PSD

for i = 1:(length(A(:,1))-1)
    if (A(i+1,1)~=A(i,1))
        data(cnt, :) = A(i, :);
        cnt = cnt + 1;
    end
end
data(cnt, :) = A(end, :);

% Angle_x = unique(data(:,1));
% Angle_y = unique(data(:,2));
% Q1 = reshape(data(:,3)-1.28, length(Angle_x), length(Angle_y));
% Q2 = reshape(data(:,4)-1.28, length(Angle_x), length(Angle_y));
% Q3 = reshape(data(:,5)-1.28, length(Angle_x), length(Angle_y));
% Q4 = reshape(data(:,6)-1.28, length(Angle_x), length(Angle_y));

Betta = unique(data(:,1));
Alpha = unique(data(:,2));

Angle_x = sind(Betta);
Angle_x = asind(Angle_x);
Angle_x = repmat(Angle_x, [1,length(Alpha)]);
Angle_y = sind(Alpha)*cosd(Betta)';
Angle_y = asind(Angle_y);

Q1 = reshape(data(:,3)-1.404, length(Betta), length(Alpha));
Q2 = reshape(data(:,4)-1.404, length(Betta), length(Alpha));
Q3 = reshape(data(:,5)-1.404, length(Betta), length(Alpha));
Q4 = reshape(data(:,6)-1.404, length(Betta), length(Alpha));

total = Q1+Q2+Q3+Q4;

% PLOT THE X-AXIS DETECTABILITY
X = ((Q2+Q3) - (Q1+Q4))./(total);
X=X*L/2;
X=atand(X/distance);
figure; mesh(Angle_y', Angle_x, X); title('PSD␣X-axis');
xlabel('Y␣angle')
ylabel('X␣angle')
zlabel('Measured␣angle')
view([24 -45 24])
xlim([-40, 40])
ylim([-40, 40])
zlim([-60, 60])

% PLOT THE Y-AXIS DETECTABILITY
Y = ((Q2+Q4) - (Q1+Q3))./(total);
Y=Y*L/2;
Y=atand(Y/distance);
figure; mesh(Angle_x, Angle_y', Y); title('PSD␣Y-axis');
xlabel('X␣angle')
```

```
ylabel('Y␣angle')
zlabel('Measured␣displacement')
view([-24 45 24])
xlim([-40, 40])
ylim([-40, 40])
zlim([-60, 60])


% % PLOT CURRENTS FOR Y-AXIS AT X=0
% figure;
% plot(Angle_y, Q1(61,:))
% hold on;
% plot(Angle_y, Q2(61,:), 'r')
% plot(Angle_y, Q3(61,:), 'g')
% plot(Angle_y, Q4(61,:), 'm')
% legend('i1','i2','i3','i4')
% xlabel('Angle (degrees)')
% ylabel('Magnitude (V)')


% % PLOT CURRENTS FOR X-AXIS AT Y=0.27
% figure;
% plot(Angle_x, Q1(:,135))
% hold on;
% plot(Angle_x, Q2(:,135), 'r')
% plot(Angle_x, Q3(:,135), 'g')
% plot(Angle_x, Q4(:,135), 'm')
% legend('i1','i2','i3','i4')
% xlabel('Angle (degrees)')
% ylabel('Magnitude (V)')


Angle_xt = Angle_x';
Angle_yt = Angle_y';
```

C.3 Code generated from curvefitting

This code was generated with the Curve Fitting Toolbox in Matlab.

C.3.1 9th degree polynomial curvefit of the 1 axis measurements

Listing: Curvefitting code for PSD X-axis

```
function [fitresult, gof] = createFit(Angle_yy, XX)
%CREATEFIT(ANGLE_YY,XX)
%  Create a fit.
%
%  Data for 'untitled fit 1' fit:
%      X Input : Angle_yy
%      Y Output: XX
%  Output:
%      fitresult : a fit object representing the fit.
%      gof : structure with goodness-of fit info.
%
%  See also FIT, CFIT, SFIT.
%  Auto-generated by MATLAB on 07-Aug-2013 11:53:44

% Linear model Poly9:
%      f(x) = p1*x^9 + p2*x^8 + p3*x^7 + p4*x^6 +
%                     p5*x^5 + p6*x^4 + p7*x^3 + p8*x^2 + p9*x + p10
% Coefficients (with 95% confidence bounds):
%        p1 = -7.069e-012  (-8.483e-012, -5.655e-012)
%        p2 = -2.982e-011  (-5.078e-011, -8.865e-012)
%        p3 =    1.43e-008  (1.166e-008, 1.694e-008)
%        p4 =  9.249e-008  (5.792e-008, 1.271e-007)
%        p5 = -1.028e-005  (-1.194e-005, -8.61e-006)
%        p6 =  -8.31e-005  (-0.0001013, -6.487e-005)
%        p7 =     0.003081  (0.002674, 0.003488)
%        p8 =      0.02005  (0.01674, 0.02335)
%        p9 =        -1.71  (-1.741, -1.678)
%        p10 =        9.361  (9.214, 9.509)

% Goodness of fit:
%   SSE: 26.67
%   R-square: 0.9997
%   Adjusted R-square: 0.9997
%   RMSE: 0.3985


%% Fit: 'untitled fit 1'.
[xData, yData] = prepareCurveData( Angle_yy, XX );

% Set up fittype and options.
ft = fittype( 'poly9' );
opts = fitoptions( ft );
opts.Lower = [-Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf];
opts.Upper = [Inf Inf Inf Inf Inf Inf Inf Inf Inf Inf];

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );

% Create a figure for the plots.
figure( 'Name', 'untitled␣fit␣1' );

% Plot fit with data.
subplot( 2, 1, 1 );
h = plot( fitresult, xData, yData );
% Label axes
xlabel( 'Y␣angle' );
ylabel( 'Measured␣angle' );
grid on

% Plot residuals.
subplot( 2, 1, 2 );
h = plot( fitresult, xData, yData, 'residuals' );
% Label axes
xlabel( 'Y␣angle' );
ylabel( 'Measured␣angle␣residuals' );
grid on
```

C.3.2 1 degree polynomial

Listing: Curvefitting code for PSD X-axis

```
function [fitresult, gof] = createFit(Angle_x, Angle_yt, X)
%CREATEFIT(ANGLE_X,ANGLE_YT,X)
```

```
% Create a fit.
%
% Data for 'PSD X-axis' fit:
%      X Input : Angle_x
%      Y Input : Angle_yt
%      Z Output: X
% Output:
%      fitresult : a fit object representing the fit.
%      gof : structure with goodness-of fit info.
%
% See also FIT, CFIT, SFIT.

% Auto-generated by MATLAB on 28-Jun-2013 13:26:13

% Linear model Poly11:
%      f(x,y) = p00 + p10*x + p01*y
% Coefficients (with 95% confidence bounds):
%        p00 =       10.15  (10.11, 10.2)
%        p10 =     0.02706  (0.02496, 0.02916)
%        p01 =      -1.165  (-1.167, -1.162)
%
% Goodness of fit:
%   SSE: 1.304e+005
%   R-square: 0.9854
%   Adjusted R-square: 0.9854
%   RMSE: 3.007


%% Fit: 'PSD X-axis'.
[xData, yData, zData] = prepareSurfaceData( Angle_x, Angle_yt, X );

% Set up fittype and options.
ft = fittype( 'poly11' );
opts = fitoptions( ft );
opts.Lower = [-Inf -Inf -Inf];
opts.Upper = [Inf Inf Inf];

% Fit model to data.
[fitresult, gof] = fit( [xData, yData], zData, ft, opts );

% Create a figure for the plots.
figure( 'Name', 'PSD␣X-axis' );

% Plot fit with data.
subplot( 2, 1, 1 );
h = plot( fitresult, [xData, yData], zData );
legend( h, 'PSD␣X-axis', 'X␣vs.␣Angle_x,␣Angle_yt', 'Location', 'NorthEast' );
% Label axes
xlabel( 'Angle_x' );
ylabel( 'Angle_yt' );
zlabel( 'X' );
grid on
view( 107.5, -2 );

% Plot residuals.
subplot( 2, 1, 2 );
h = plot( fitresult, [xData, yData], zData, 'Style', 'Residual' );
legend( h, 'PSD␣X-axis␣-␣residuals', 'Location', 'NorthEast' );
% Label axes
xlabel( 'Angle_x' );
ylabel( 'Angle_yt' );
zlabel( 'X' );
```

```
grid on
view( 107.5, -2 );
```

C.3.3 5-5 degree polynomial

Listing: Curvefitting code for PSD Y-axis

```
function [fitresult, gof] = createFit(Angle_x, Angle_yt, X)
%CREATEFIT(ANGLE_X,ANGLE_YT,X)
%  Create a fit.
%
%  Data for 'PSD X-axis' fit:
%      X Input : Angle_x
%      Y Input : Angle_yt
%      Z Output: X
%  Output:
%      fitresult : a fit object representing the fit.
%      gof : structure with goodness-of fit info.
%
%  See also FIT, CFIT, SFIT.

%  Auto-generated by MATLAB on 28-Jun-2013 13:04:02

% Linear model Poly55:
%      f(x,y) = p00 + p10*x + p01*y + p20*x^2 + p11*x*y + p02*y^2 + p30*x^3 + p21*x^2*y
%                     + p12*x*y^2 + p03*y^3 + p40*x^4 + p31*x^3*y + p22*x^2*y^2
%                     + p13*x*y^3 + p04*y^4 + p50*x^5 + p41*x^4*y + p32*x^3*y^2
%                     + p23*x^2*y^3 + p14*x*y^4 + p05*y^5
% Coefficients (with 95% confidence bounds):
%        p00 =       10.48  (10.44, 10.52)
%        p10 =     0.05132  (0.04801, 0.05462)
%        p01 =      -1.119  (-1.123, -1.116)
%        p20 =    0.001887  (0.001781, 0.001992)
%        p11 =   7.902e-006  (-0.0001036, 0.0001194)
%        p02 =   -0.002351  (-0.002482, -0.002219)
%        p30 =  -5.414e-005  (-6.059e-005, -4.769e-005)
%        p21 =   -0.000129  (-0.0001357, -0.0001223)
%        p12 =  -3.348e-005  (-4.154e-005, -2.541e-005)
%        p03 =   0.0001247  (0.000116, 0.0001334)
%        p40 =   1.305e-007  (6.488e-008, 1.962e-007)
%        p31 =   3.823e-009  (-6.689e-008, 7.453e-008)
%        p22 =  -2.171e-006  (-2.257e-006, -2.085e-006)
%        p13 =     5.62e-008  (-3.853e-008, 1.509e-007)
%        p04 =      2.1e-007  (1.193e-007, 3.007e-007)
%        p50 =   2.796e-008  (2.473e-008, 3.119e-008)
%        p41 =  -3.674e-008  (-4.02e-008, -3.329e-008)
%        p32 =   2.341e-008  (1.913e-008, 2.769e-008)
%        p23 =  -1.072e-007  (-1.124e-007, -1.02e-007)
%        p14 =   5.226e-009  (-1.621e-010, 1.061e-008)
%        p05 =  -4.435e-008  (-4.91e-008, -3.961e-008)
%
% Goodness of fit:
%   SSE: 9460
%   R-square: 0.9989
%   Adjusted R-square: 0.9989
%   RMSE: 0.8106
%
%


%% Fit: 'PSD X-axis'.
```

```
[xData, yData, zData] = prepareSurfaceData( Angle_x, Angle_yt, X );

% Set up fittype and options.
ft = fittype( 'poly55' );
opts = fitoptions( ft );
opts.Lower = [-Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -In
opts.Upper = [Inf Inf Inf Inf Inf Inf Inf Inf Inf Inf Inf Inf Inf Inf Inf Inf Inf Inf Inf In

% Fit model to data.
[fitresult, gof] = fit( [xData, yData], zData, ft, opts );

% Create a figure for the plots.
figure( 'Name', 'PSD␣X-axis' );

% Plot fit with data.
subplot( 2, 1, 1 );
h = plot( fitresult, [xData, yData], zData );
legend( h, 'PSD␣X-axis', 'X␣vs.␣Angle_x,␣Angle_yt', 'Location', 'NorthEast' );
% Label axes
xlabel( 'Angle_x' );
ylabel( 'Angle_yt' );
zlabel( 'X' );
grid on
view( 107.5, -2 );

% Plot residuals.
subplot( 2, 1, 2 );
h = plot( fitresult, [xData, yData], zData, 'Style', 'Residual' );
legend( h, 'PSD␣X-axis␣-␣residuals', 'Location', 'NorthEast' );
% Label axes
xlabel( 'Angle_x' );
ylabel( 'Angle_yt' );
zlabel( 'X' );
grid on
view( 107.5, -2 );
```

Appendix D

Miscellaneous

D.1 Debugging of the Digital Sun Sensor

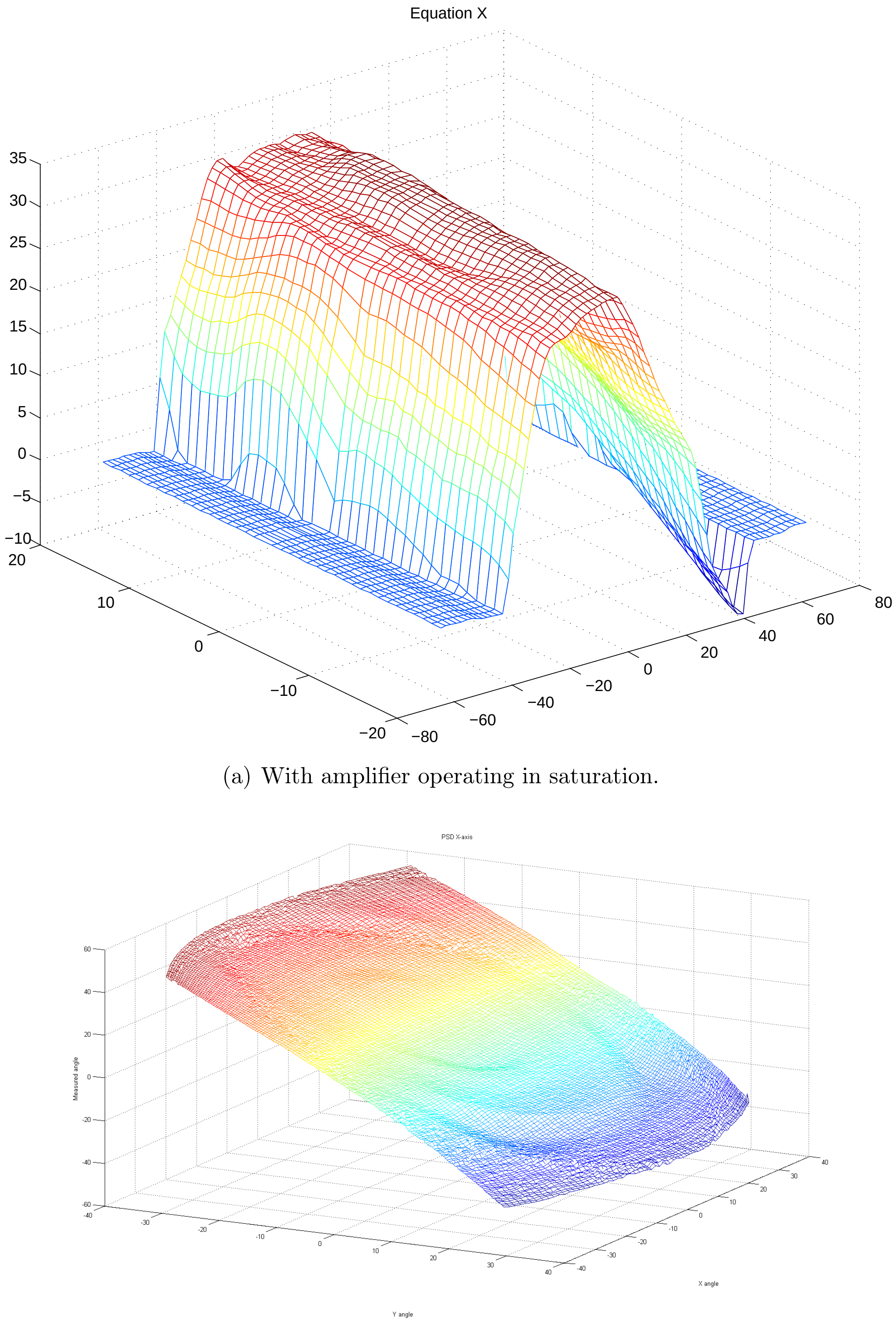
When the sun sensor was tested for the final performance test, it seemed that the sensor had stopped working. As seen in Figure D.1, the X-axis of the detector gave a very strange response, compared to the Y-axis in Figure D.2, which looked more like the correct curve.

It was suspected that the light source might have something to do with it, but the same results were measured with the sensor mounted with a 90° tilt.

Every component value in the analog front-end was controlled, to make sure that a mismatched resistor, or something of the like, was the error.

As discussed in Section 6.1.2, it was found that the operational amplifier was operating in saturation because of a too low supply voltage. Apparently, there had been a miscalculation when increasing the output voltage level, which lead to this mistake not being discovered. The reason that this wasn't discovered earlier is that the first tests were done with a laser, which makes a more intense light beam on the sensor than the car headlight. With the laser, the input current to the amplifier would almost always be large enough to eliminate this problem.

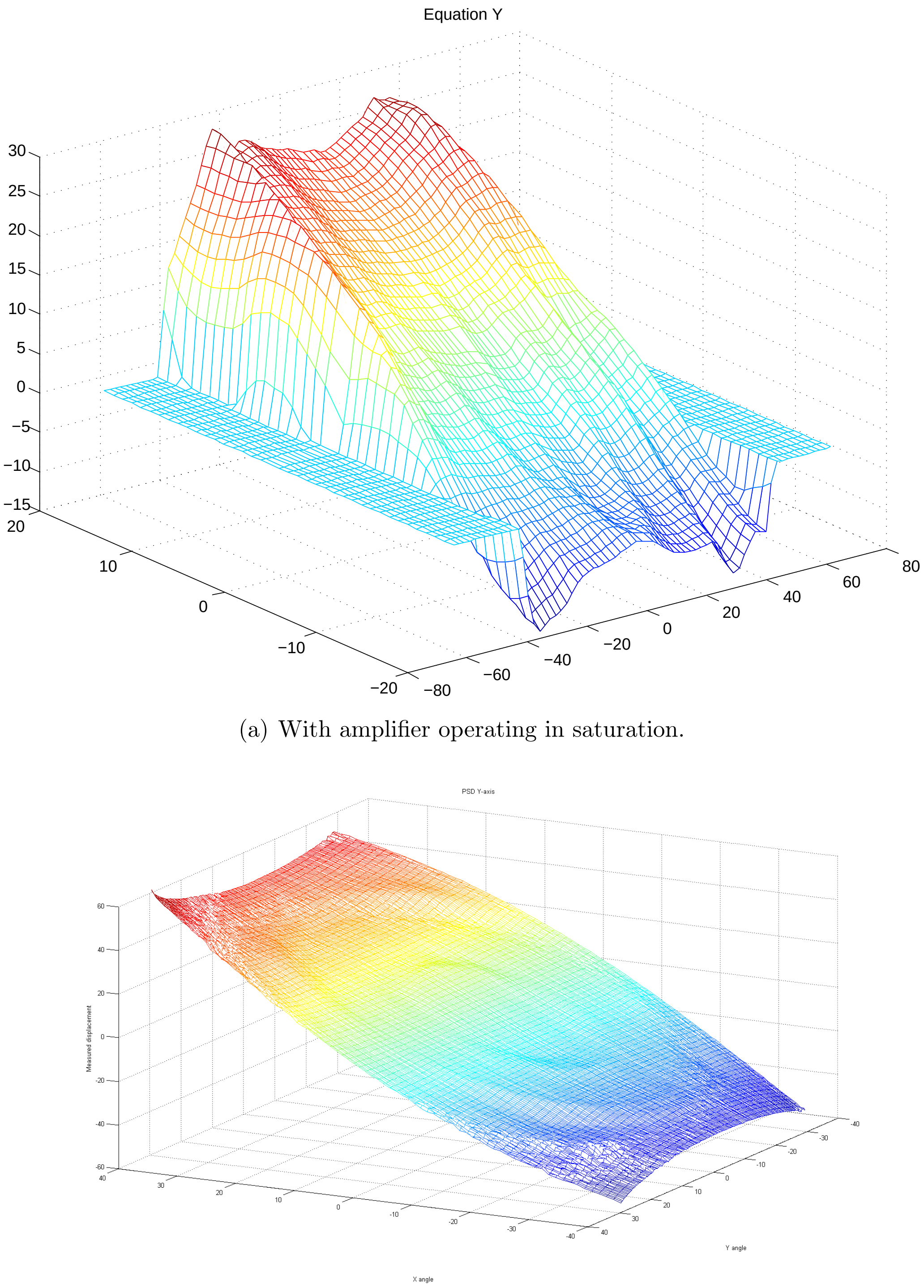
It can be seen in Figure D.3 and Figure D.4 that two of the electrodes gave much lower response than the others, which was caused by the amplifier being insensitive to very low current signals. Since the amplifier was operating in saturation, the output voltage did not change before a large enough current was present on the input.



(a) With amplifier operating in saturation.

(b) Without saturation.

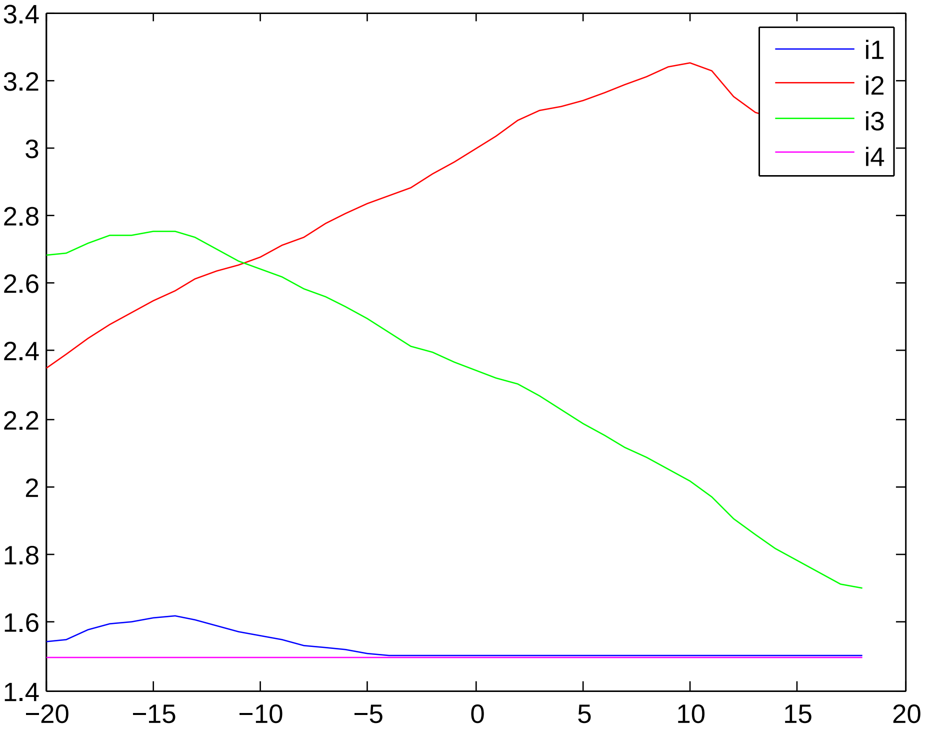
Figure D.1: The response of the PSDs X-axis before and after debugging.



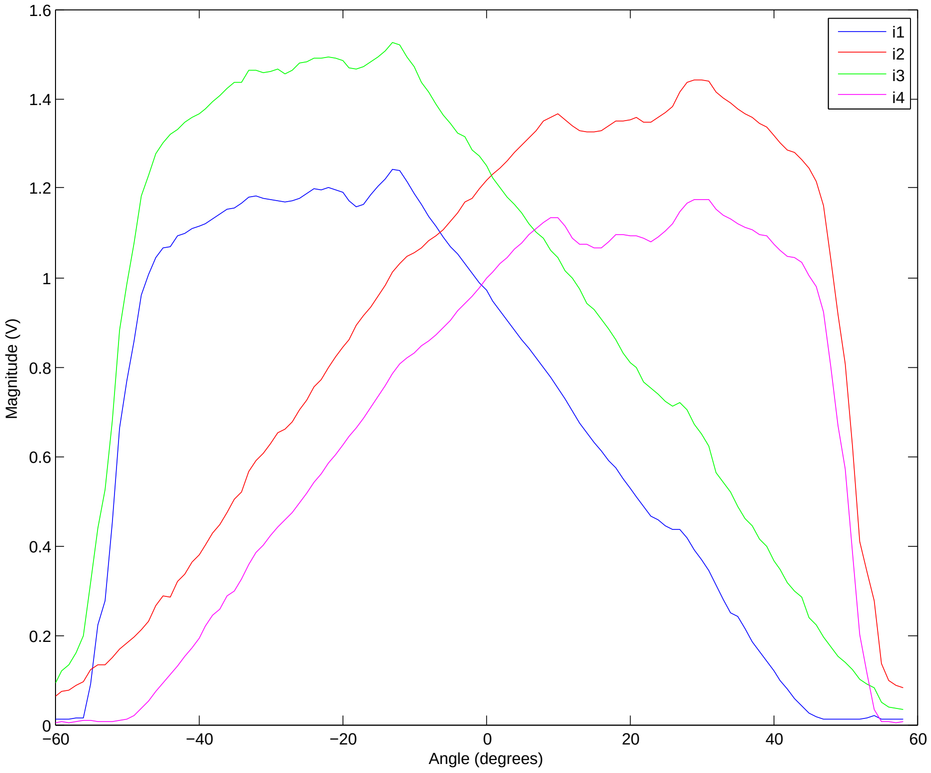
(a) With amplifier operating in saturation.

(b) Without saturation.

Figure D.2: The response of the PSDs X-axis before and after debugging.

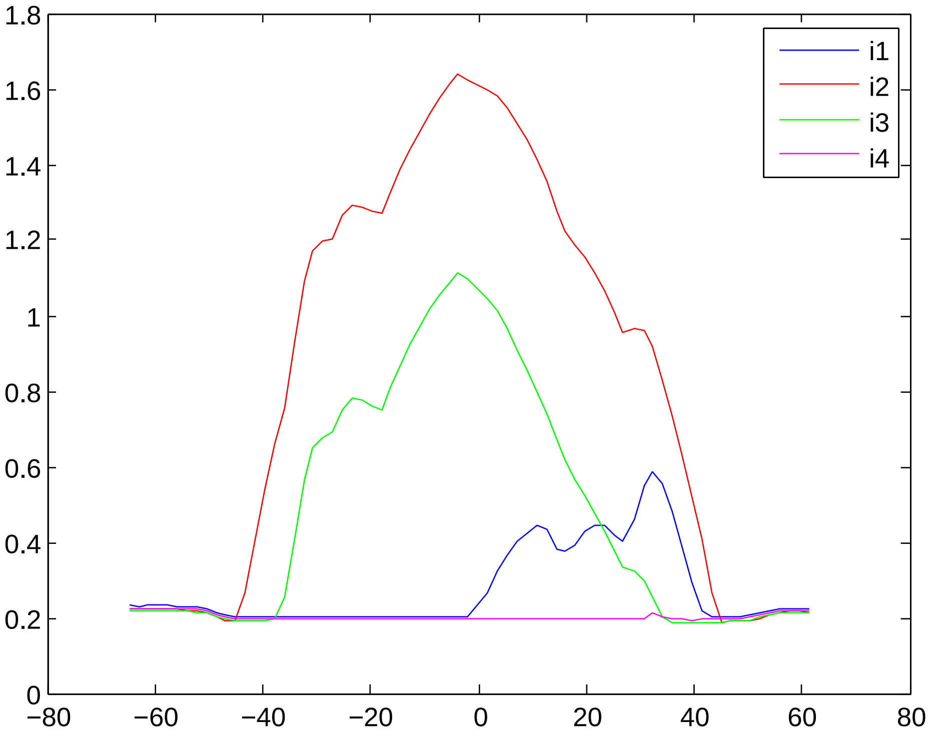


(a) X axis at 1.89° Y, with amplifier operating in saturation.

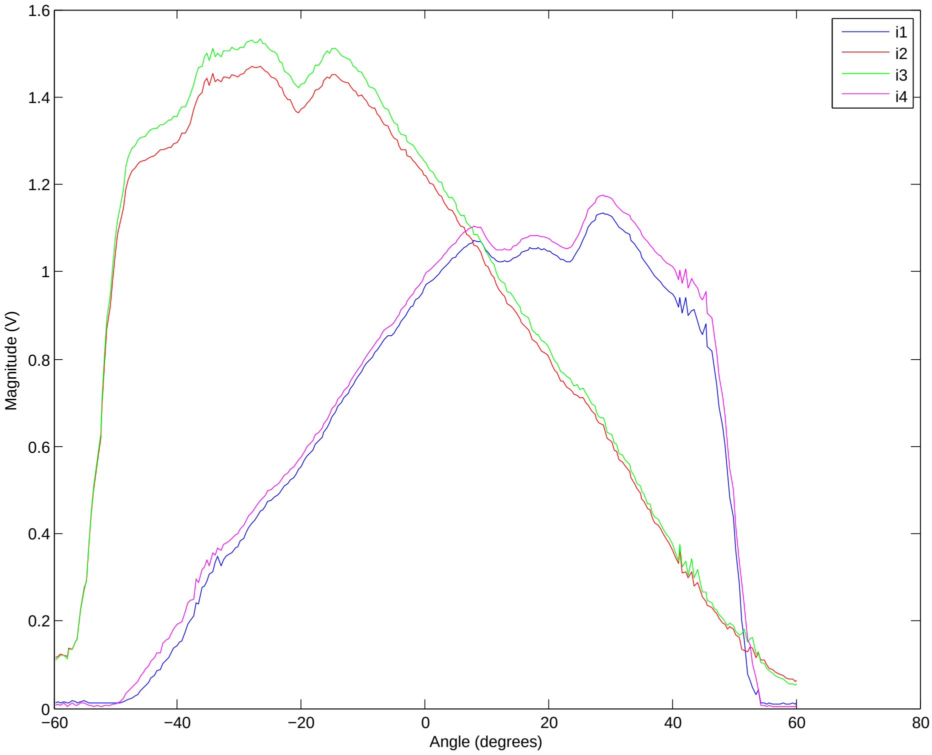


(b) X axis at 0.27° Y, without saturation

Figure D.3: The output signals measured by the ADC across the X axis before the problem was found. It is clear that something was wrong with the I_3 and I_4 signals.



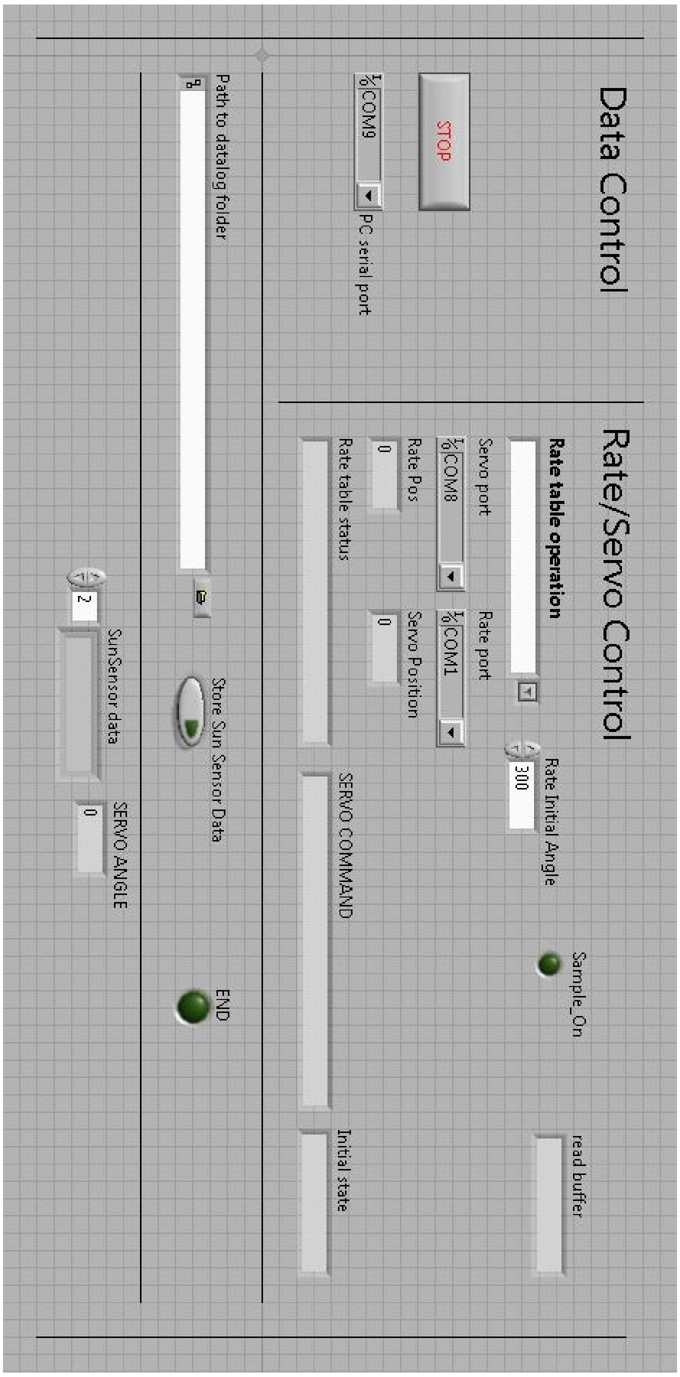
(a) Y axis at 1° X, with amplifier operating in saturation.



(b) Y axis at 0° X, without saturation

Figure D.4: The output signals measured by the ADC across the Y axis before the problem was found. It is clear that something was wrong with the I_3 and I_4 signals.

D.2 LabView calibration routine



PC serial port %COM9

Servo port %COM8

Rate port %COM1

Rate Pos 0

Servo Position 0

Rate Initial Angle 300

SunSensor data 2

SERVO ANGLE 0

Figure D.5: A screenshot of the LabView VI used for testing the sun sensor.