NTNU – Trondheim
Norwegian University of
Science and Technology

IMU for Transportation Units

Øyvind Lorgen

Master of Science in Engineering Cybernetics
Submission date: July 2012
Supervisor: Tor Engebret Onshus, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Inertial Motion Unit DAQ for transportation units

Oyvind Lorgen
Institute of Cybernetics,
Norwegian University of Science and Technology,
Trondheim,
Norway. Spring 2012
lorgen@stud.ntnu.no

July 9, 2012

NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for informasjonsteknologi, matematikk og elektroteknikk
Institutt for teknisk kybernetikk

MASTEROPPGAVE

Kandidatens navn: Øyvind Lorgen

Fag: Teknisk Kybernetikk

Oppgavens tittel (norsk): Inertial Motion Sensor datalogger for Transportenheter

Oppgavens tittel (engelsk): Inertial Motion Datalogger DAQ for Transportationunits

Oppgavens tekst:

Målet med denne masteroppgaven vil være å utvikle en sensorplattform i form av en datalogger for samling av bevegelsesdata. Denne vil benyttes til å samle bevegelses data, orientering og krefter fra fartøy under forflytting. Systemet må være i stand til å logge data fra de ulike sensorene lokalt, for å muligjør videre analyse av ytelse og dynamikk.

Ved design av et måleinstrument så er det viktig å kunne teste kvaliteten og egenskapene før det benyttes på tiltenkt system. Egenskaper vil bli vurdert ved å analysere serier av måledata fra sensorplattformen.

Første fase i oppgaven vil vere å kartlegge tilgjengelig ferdig hardware og benytte et utvalg av komponenter for å designe et system med ønskede egenskaper. Det vil være behov for å utvikle software for å samle informasjon fra sensorene effektivt og lagre disse uten tap av informasjon. Til slutt vil det vurderes om design basert på lavkost hardware komponenter kan gi brukeren data som kan bearbeides og benyttes til tiltenk formål.

- Undersøke løsninger og velge nødvendig hardware for systemet
- Presentere de fysiske faktorer for sensorene
- Designe robust system for logging og lagring av målinger fra et sett med sensorer
- Fokusere på enkelt design og mobilitet
- Testing av sensorplattformen, kartlegge egenskaper ved løsningene.
- Presentasjon av resultater

Oppgaven gitt: 06. februar 2012

Utført ved Institutt for teknisk kybernetikk

Veileder: Professor, Dr.Ing Tor Onshus

Trondheim, ___. _____ 2012

Veileder 2: Tore Eide, Ms.c

Trondheim, ___. _____ 2012

Faglærer Tor Onshus

Denne masteroppgaven er besvarelse p nsket Cargosafe Systems har til utvikle en datalogger som kan lagre bevegelsdata. Oppgaven er et initiativ fra Cargosafe og er gjort i sammarbeid med Norges teknisk-naturvitenskapelige universitet. Det er nsket at enheten skal kunne samle data selvstendig uten noe kabling utenifra og forhndsbestemte omrder. Denne masteroppgaven er utviklinger av en stand-alone datalogger og verkty.

Dataloggeen er basert p hyllevarekomponenter, egen utviklede komponenter og software utvikling. Grunnleggende forstelse av begrensninger, virkemte, kalibrering og tema angende bevelegses sensorer blir presentert. Dette vil vre rettningslinjer ta hensyn til for denne typen anordning.

For yeblikket s er dataloggeren et verkty for samling av sensordata, og disse dataene blir etterarbeidet p en datamaskin. Dette for minske belastningen p enheten, og nr man mottar RAW data s er det mulig studere sensorene uten tap eller komprimering av informasjon. Egenlaget C kode blir brukt som rammeverk og muligjr ndvendig funksjonalitet. JavaSkript er benytttet p den ene grafiske grensesnittet.

Lsningen utarbeidet i denne masteroppgaven er n en fult brukbar platform for samling av bevegelsesdata og lagre disse p portabelt minne. Logging til minnekort muligjr analyse etter at en test er gjennomfrt. Dataloggeren gr p batteri og lang kjretid og stor lagringskapasitet gjr den godt egnet til samle sensordata. Det er forskt minimalisere feil fra miljpvirkning og feil fra selve sensorene.

Dette er en lsning til en portabel datalogger platfor for sensor-integrasjon system, som er enkelt, plitelig og noe robust. Bevegelses sensor platformer innbakes i stadig flere lsninger, men sjeldent som en total lsning med batteri og lokal lagring. Dette gir fleksibilitet og enkel bruk der brukeren kan motere plattformen hvor man mtte nske. Enkelt design som benytter utskiftbare lag og kostnadsvennlige komponenter. Den robuste software ytelsen oppnes ved bruk av enkel, men plitelig software.

Abstract

This thesis is made upon a request from Cargosafe Systems., where the overall goal is to develop a motion based Data-logger. The project initiated by Cargosafe is developed in collaboration with the Norwegian University of Science and Technology(NTNU). To be able to collect sensor data from motion sensors from user defined placement, it is favourable that the unit is emancipated from cabling and a predefined environment. The thesis is the development of the stand-alone datalogger and tools.

The datalogger is based on "off-the-shelf" hardware, own made hardware and software developed during the project. In order to produce reliable data, basic understanding of limitation, functionality, calibration and issues regarding motion sensors are presented. The information presented will act as a guideline when designing this type of devices. It is also important to regard these issues due to real tests, to ensure correctness of sampled data.

As for now, the datalogger will act as a tool for collecting sensor data, and the collected data are post-processed on a computer. This to reduce load on the unit, and receiving the RAW measurement makes it possible to study the data without loss of information. Custom made C code is used as framework providing the necessary functionality. JavaScript is used for the graphical user interface(GUI).

The solution developed in this thesis is now a usable platform for collecting motion data and store these to portable memory. Logging to memory makes it possible to analyse the data after the test have been conducted. The datalogger is battery-operated with long runtime and large data space making it useful for collecting motion from sensors. An effort is made to minimize errors caused by environmental disturbance and errors located in the sensor itself

This is a solution to a stand-alone datalogger based sensor fusion system, which is simple, reliable and robust for motion based measuring. Currently, motion sensor platform are created in a variety of applications, but seldom as a total system with local storage capability and battery operated. This add flexibility and ease of use as the device can be mounted at location needed by the user. Here a solution to a datalogger in a small package, with modularity on a platform with a 8-bit controller in combination with high frequency and large program memory in a compact package. Following a simple design with replaceable sandwich approach this system is based on a small number of low-cost components. Its robust performance is achieved by using simple but reliable software.

Preface

This thesis is written during the spring semester 2012, and is the result of Master degree at Institute of Engineering Cybernetics. The thesis presents the development and testing of a low-power datalogger intended for motion sensing and measurement.

The design and development of this system is based on a request from Cargosafe Systems. They needed to investigate the utilization of motion based MEMS sensors, and highlight issues regarding low cost MEMS based technology. The task was mainly to crate a stand-alone datalogger with a minimum of performance, low cost and battery-powered. This would be a base for their further design of motion sensor, with the capability of storing information about motion. The unit should act as a tool for collecting sensor data.

First of all I want to thank my teaching supervisor, professor Tor Onshus at ITK. His guidance, demand for a tight schedule and periodic follow-up has made it possible to have a continuous progress with focus on crating the solution. I want to send a thanks to Tore Eide at Cargosafe Systems for support, guidance, enthusiasm, highlight weakness with the unit, being highly available at any hour and for personal help and interest during testing. Knut Reklev for lending development tools on the spot. Last I want to send a great thanks to the department engineer John Olav Horrigmo for lending me equipment, guidance during PCB design, soldering and production, and for exceptional service when acquiring components.

Oyvind Lorgen, Spring 2012.

Contents

List of Figures

List of Tables

1 Introduction

Rapid development in micro-electronics and falling productions cost have made motion sensors highly available. Motion based sensors are implemented in all sorts of application spreading from a industrial range to personal applications. Reason development of MEMS based sensor fusion boards, have created small, low-power packages with a high accuracy.

The main goal for this thesis is to develop a motion based datalogger platform from base. Hardware must be selected fitting the requirements from Cargosafe System. Necessary software for a fully functioning datalogger must be designed. The system will be tested and documented for its performance. The long time stability, response and speed of the total solution will be deciding factor for a successful result.

Chapter 2 will give a brief introduction to physical aspects regarding motions. The Chapter 3 introduces the requirements to the platform included thoughts about designing the system. Followed by Chapter 4 that gives a short overview of the sensors used. Chapter 5-, 6-, and 7 goes through the proses of selecting and designing the hardware for the system. Chapter 8 then presents the software implementation. The calculations and calibration routine used are presented in Chapter 9. For communicating with the sensor platform and read data, the graphical user interfaces are included in Chapter 10. Testing and results are presented in Chapter 11, and Chapter 12, 13, 14 will give a short discussion, a conclusion and thoughts for further improvements. Appendix and Biblography can be found in the end of this report.

2

Sensor Physics Explained

Some insight of the physical factors about motion and sensors has to be known to study the data and consider limitations about motion capturing. Here, key-terms are presented. The Acceleration, Force, Measure Acceleration parts are the result of earlier work. [40, Project]

2.1 Acceleration

Any change in the velocity of an object results in an acceleration. This be a change in direction, increasing of or decreasing the decreasing speed of an object. So the change in the direction of motion will result in a change of acceleration. The velocity is a vector component with both magnitude and direction, as acceleration is dependent of velocity a change in velocity will affect the acceleration. The acceleration term can be split into two parts; Average acceleration and instantaneous acceleration. The average acceleration is determined over a time interval. This is based on the initial velocity v_0 and the final velocity V and divided on the time interval.

$$\bar{a} = \frac{\Delta V}{\Delta t} = \frac{v - v_0}{\Delta t} \tag{1}$$

In contrast, instantaneous acceleration is measured over a infinitesimal time interval having no duration whatsoever. This is a purely mathematical ideal that can only be realized as a limit. Instantaneous acceleration is then the limit of average acceleration as the time interval approaches zero. Or simply the derivative of velocity.

$$\vec{a} = \lim_{\Delta t \to \infty} \frac{\Delta V}{\Delta t} = \frac{d\vec{v}}{dt} \tag{2}$$

Acceleration is the derivation of velocity with time, and velocity is itself the derivation of displacement with time. This gives the following:

$$\vec{a} = \frac{d\vec{v}}{dt} = \frac{d}{dt}\frac{d\vec{r}}{dt} = \frac{d^2 r}{dt^2} \tag{3}$$

As calculating acceleration is the same as dividing distance by time twice, the SI unit of acceleration is given as meter per second squared: $\frac{\text{metre/s}}{\text{second}}$

However, the acceleration in this project is used due to earth's gravity - g. This seem convenient to use as the sensor principle here will apply gravitation for its measurement

of force, movement and orientation. Every object on the surface of Earth are influenced by the gravitation force of $1g$. When an object is weightless the gravitation is $0g$. The gravitational unit is precisely defined to 9.80665 however it is more convenient to use 9.81 which also is most common. The gravitational force also differs depending on where the object is located on the earth surface.

The acceleration definition used in this project will be the proper acceleration. This is the physical acceleration experienced by an object and it is the acceleration relative to a free-fall, or inertial, observer who is momentarily at rest to the object being measured. The acceleration of gravity or the force of gravity does not contribute to proper acceleration. So the proper acceleration felt by observers standing on the ground is due to the mechanical force from the ground, and not due to gravity. Simply said, if ground is removed an object would experience free fall and a feeling of weightlessness. Acceleration now would be coordinate acceleration. [18]

A lot of special cases and reference frames exist for the dynamics and observation of acceleration. This is out of context for this project and only the acceleration due to change in velocity and the proper acceleration are continued.

2.2 Force

In classic mechanics the Newtons law's of motion describe forces acting on a body and its motion due to those forces.

The first and second laws of Newton combined together are used to define force. A force is an external influence on an object that causes it to accelerate relative to an inertial reference frame. The direction of the acceleration will give the direction of the applied force. The magnitude of the force can be seen as the product of the mass of the object and the magnitude of the acceleration. This is used directly in measurement of acceleration and is given by the definition of the Second law.

The second law of newton states: The acceleration a of a body is parallel and directly proportional to the net force $\vec{F}$ and inversely proportional to the mass m, i.e., $\vec{F} = m\vec{a}$ [2].

$$\sum \vec{F} = m\vec{a} \tag{4}$$

This is also called the proper force and its equal to the opposing reaction force that is measured as an object's "operational weight". The proper force will always be equal or opposite to its measured weight.

To remind the reader, the force required to produce an acceleration of 1 $[\mathrm{m\,s^{-2}}]$ on the standard object is defined to be 1 *newton* [N]. In theory, if air resistance is neglected all objects will have the same acceleration towards the earth. This is what is called

free-fall acceleration $\vec{g}$. The force causing this acceleration is the gravitational force on the object, called its weight. From Newton's second law the weight is defined as

$$\vec{w} = m\vec{g} \tag{5}$$

As $\vec{g}$ is the same for every objects it follows that the weight of an object is proportional to its mass. The vector $\vec{g}$ us the force per unit mass exerted by the earth on any object an is called the gravitational field of the earth. At the earth's surface the vector $\vec{g}$ has the magnitude of: [2].

$$g = 9.81 \tag{6}$$

For the rest of this project the acceleration due to gravity of the earth is defined $\vec{a} = 9.81\text{m/s}$ at -90°

2.3 Angular velocity

Every point in a body rotating about a fixed axis moves in a circle whose centre is on the axis and whose radius is the distance of that point from the axis of rotation[2, page267-268]. A line drawn from this axis to any point sweeps out the same angle in the same time. As the disk rotates through an angle $d\theta$, the particle moves through a circular arc of length ds_i, such that

$$ds_i = r_i d\theta \tag{7}$$

where $d\theta$ is measured in radians.

For one complete revolution, the arc length ΔS_i is $2\pi r$ and the angular displacement $\Delta\theta$ is

$$\Delta\theta = \frac{2\pi r_i}{r_i} = 2\pi r rad = 360^\circ = 1 rev \tag{8}$$

The time rate of change of the angle is the same for all particles of the disk, and is called the angular velocity ω of the disk:

$$\omega = \frac{d\theta}{dt} \tag{9}$$

For counter-clockwise rotation θ increases, so ω is positive. The unit ω is given as radians per second. The magnitude of angular velocity is called the angular speed. To convert between revolutions, radians and degrees, we use that one $rev = 2\pi rad = 360^\circ$

2.4 Angle measurements

On a moving vehicle, the angle measurement has to be done without access to a common or known reference point. Like in autopilot for planes is it essential to have a good measurement of the vehicle angles (pitch, roll, yaw). A inertial motion unit can be used for this task. This is done by measuring linear acceleration and angular velocity along three axis, and the position or velocity can be estimated by more or less accurate precision. The basis components in such systems are accelerometer for measuring linear motion and gyroscope for angular velocity measurement around one axes.

-Identify signature; Force, Vibration, Free fall, Any Motion, Roll, Pitch, Yaw, Rotation, Acceleration, Stop

2.5 Problem using g-force as reference

If the accelerometer sensor is placed in a box or cart that is fixed to earth, we may sense the tilt by decomposing the gravitational force pull on the system. However, if the system leave the earth's ground and fly and it exceeds 1g force value. A sensor monitoring gravity will lose any sense of the earth's pull and its associated reference. So that the acceleration sensor can not distinguish between acceleration and gravity. Therefore the tilt in the earth's frame of reference cannot be measured any longer. A solution to read orientation in this case is the use of magnetometers.

Weak flux lines makes it hard to read correctly, and the fluctuations that may be induced by metallic surfaces does affect its performance.

2.6 5 motion of senses

In this thesis, 5 different motions can be detected and measured. These motions are:

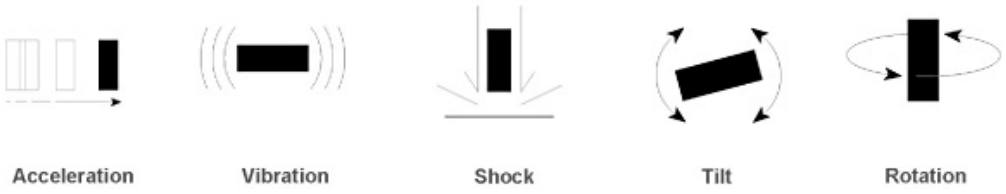


Figure 1: 5 Motions

To measure these motions a variation of sensors will be used. These sensors are presented in a later chapter. Now, some thoughts about motion based applications will be described.

From testing it may be proved that the accelerometer can pick up small vibrations of a certain limit making it possible to warn about motions that should be inspected.

Gesture recognition is another possibility, like shock, large force or other external motions experienced can be dealt with or reported.

Tilt sensing can be used for precise operation and stability warning. Accelerometer would recognise rotation as tilt in case of measuring low-speed changes in inclination in the presence of gravity detecting the change in gravity vector. Here the direction can be identified in clockwise or counter clockwise direction.

Angular measurement by gyroscopes can be provided for compensation of heading error in compass, calculate weight accurately by display the vector of the force or report inclination when experience "free-fall".

Combining rotation sensing in the application offers real-world benefit combined with other inertial sensing. In practice a combination of accelerometer and gyroscope is required for orientation sensing. Inertial Measurement Unit (IMU) for high heading accuracy include all accelerometer, gyroscope and magnetometer. IMU provides full 6 degrees of freedom measurement.

2.7 Measure Acceleration - Accelerometer

A common accelerometer measure what is know as proper acceleration. As described earlier this is the phenomena of weight experienced by a test mass that resides in the frame of reference of the accelerometer device. So the accelerometer in practice measure the weight per unit of test-mass, which is also known as specific force, or g-force. The ability to detect and sense the magnitude and direction of proper acceleration makes it possible for the single and multi-axis accelerometer to know its orientation, measure shock, tilt, vibration and coordinate acceleration.

When the accelerometer is at rest relative to the Earth's surface it will indicate approximately $1g$ upwards as the point of the earth is accelerating upwards relative to the local inertial frame.

2.7.1 Principe of measurement

The accelerometer behaves conceptually as a damped mass on spring. The mass inside the accelerometer is displaced to the point that the spring is able to accelerate the mass at same rate as the casing. By measuring the displacement of the mass the magnitude of the acceleration is given. The support beam inside a accelerometer act as a spring, and air trapped inside the integrated circuit(IC) acts as a damper. This result in a second order lumped physical system and is the main source of the operational bandwidth and non-uniform frequency response of an accelerometer. There exist a variety of configurations to

resolve the acceleration electro-mechanically. Piezoelectric, piezoresistive and capacitive components are commonly used to convert the mechanical motion into an electrical signal. Care must be taken when selecting measurement technology as some methods are better suited for some application than others. This can be in type of response, dynamic or functionality. This project will use a capacitative sensor unit as it is most commonly used in MEMS element. The use of capacitive accelerometers has proven to be superior in the low frequency range[30].

The capacitive accelerometer uses capacitances in order to measure the acceleration. This way to calculate the acceleration uses two basic principles of physics. The first one is Hooke's Law which states that a spring, when stretched, will exert a restoring force F that's proportional to its increase in length x.

$$F = kx \tag{10}$$

The second is Newton's Second Law, which was stated earlier in equation (4). By combining (10) and (4) give the equation

$$A = \frac{kx}{m} \tag{11}$$

This mean that the body with mass m will stretch a spring by a distance of x if its acceleration is A.

Normally a capacitive accelerometer system consist of a bar of silicon(the mass) that is held by four tethers(anchor in every corner).

The four tethers compose the spring system. When the mass is subjected to acceleration, it moves with respect to the anchored feet of the tethers, causing the tethers to 'stretch' like a spring. The greater the acceleration the greater the displacement. This system allow the accelerometer to be measured by translate the displacement into acceleration.

The amounts and rates of change of capacitance measurement are translated by on-chip signal conditioning circuits into an output voltage that indicates the strength and direction of the acceleration. The on-chip signal conditioning circuitry may consist of amplifiers, filters, oscillators, demodulators, and even self-test circuitry [14].

2.7.2 Orientation

The orientation of the device is critical to get a good response and sensitivity. Simple calculation can derive the difference in placing the measurement in $0g$ and $1g$ orientation. This is illustrated by the following

$$G_N = g \times \cos(\theta) \tag{12}$$

Tilted by an angle of $\theta = 1^\circ$ gives $G_N = 0.9998^\circ g$

Compared to measurement in the $0g$ orientation this gives

$$G_N = g \times \sin(\theta) \tag{13}$$

By tilting $\theta = 1^\circ$ in this direction the resulting g-force becomes $G_N = 1.7x10^-2g$

As shown above a change of 1° in the $0g$ orientation causes 58x bigger change in sensor output versus the $\pm 1g$ orientation. As for above the X-axis is 58x more sensitive than the Z-axis measurement.

To select the appropriate sensor for the design some understanding of important parameters and keyword are needed to make a good choice when trying to select the best sensor. This chapter presents the key factors to consider when introducing acceleration measurement into the design.

2.8 Measure Rotation - Gyroscope

It is interesting to get meaningful information from the gyroscope in terms of angular velocity and angular displacement. The MEMS gyroscopes uses the Coriolis Effect to measure the angular rate[47]. Rotating the gyro sensor subjects its structure to the Coriolis force. The rotational velocity about some axis is exploited by the physical property of vibration. Vibrating structure gyroscopes, like MEMS devices are easily available commercially, affordable and very small in size.

2.8.1 Principe of Measurement

Principal, it is the same as a rotating gyroscope but since the assumption of linear velocity in a rotating system, the resultant force is named the Coriolis force. When a mass(m) is moving in direction $\vec{v}$ and angular rotation velocity $\vec{\Omega}$ is applied, the mass will experience a tangential force as a result of Coriolis force. The velocity of v is given by a resonating vibrating mass. The specific resonance frequency makes these gyroscopes especially sensitive for mechanical noise at this frequency. (5-20 kHz).

Resulting physical displacement caused by the Coriolis force is then read from a capacitive sensing structure. Highly used configuration in these sensing devices utilizing Coriolis are tuning fork[48].

Considering two masses oscillate and move constantly in opposite directions. The Coriolis force on each mass will act in opposite direction when angular velocity is applied. This further results in a change of measured capacitance. Differential value in capacitance is proportional to the angular velocity . The measurement is converted into either voltage or LSBs output from the device.

If linear acceleration is applied to two masses, they will move in the same direction. For this reason, no capacitance difference are detected. Due to this, MEMS gyroscopes are not sensitive to linear acceleration like tilt, shock or vibration.

Fundamental to an understanding of the operation of an vibrating structure gyroscope is an understanding of the Coriolis force. In a rotating system, every point rotates with the same rotational velocity. As one approaches the axis of rotation of the system, the rotational velocity remains the same, but the speed in the direction perpendicular to the axis of rotation decreases. Thus, in order to travel in a straight line towards or away from the axis of rotation while on a rotating system, lateral speed must be either increased or decreased in order to maintain the same relative angular position (longitude) on the body. The act of slowing down or speeding up is acceleration, and the Coriolis force is this acceleration times the mass of the object whose longitude is to be maintained. The Coriolis force is proportional to both the angular velocity of the rotating object and the velocity of the object moving towards or away from the axis of rotation.

$$\vec{a_{cor}} = 2\vec{V} \times \sigma\vec{\Omega} \tag{14}$$

Vibrating structure gyroscopes contain a micro-machined mass which is connected to an outer housing by a set of springs. This outer housing is connected to the fixed circuit board by a second set of orthogonal springs.

The mass is continuously driven sinusoidally along the first set of springs. Any rotation of the system will induce Coriolis acceleration in the mass, pushing it in the direction of the second set of springs. As the mass is driven away from the axis of rotation, the mass will be pushed perpendicularly in one direction, and as it is driven back toward the axis of rotation, it will be pushed in the opposite direction, due to the Coriolis force acting on the mass.

The Coriolis force is detected by capacitive sense fingers that are along the mass housing and the rigid structure. As the mass is pushed by the Coriolis force, a differential capacitance will be detected as the sensing fingers are brought closer together. When the mass is pushed in the opposite direction, different sets of sense fingers are brought closer together; thus the sensor can detect both the magnitude and direction of the angular velocity of the system

2.9 Inertial Measurement Unit

A 3-axis gyroscope implemented with a 3-axis accelerometer can provide a full 6DOF motion tracking system. A Inertial Measurement Unit measure and reports velocity, orientation and gravitational force. It will calculate its current position based on velocity and time.

An inertial measurement unit is an electronic device that measures and reports a moving object's velocity, orientation and gravitational forces. This by using a combination og

accelerometers and gyroscopes. The data from the IMU sensors allows a computer to track position using a method known as *Dead Reckoning.*[49]

A combination of sensors are used to determine changes in vehicle movement. Sensors are typically accelerometers, gyroscopes and magnetometers. How many sensors incorporated into an IMU is referred to as its number of degrees of freedom (DOF). For example, if you have an IMU with accelerometers for the X, Y and Z-axis and a same number of gyroscopes, you have an IMU with 6 degrees of freedom, commonly abbreviated as 6DOF. Three more axes from yet another sensor, a magnetometer, is often referred to as 9 DOF system. However this is not quite true. The magnetometer is used for error correction on heading, so the real numbers of freedom will still be 6DOF. Using a IMU to transform the vehicles orientation to the earths orientation turns an IMU into an AHRS an attitude/heading reference system.

Some IMU output only the rate of rotation from gyros(IMU), while other output Euler-Angles or vector references.[49]

Dead Reckoning Calculation of current position based on previously determined position and advancing that position upon known or estimated speeds over elapsed time and course. Using best estimates of speed and direction is subject to cumulative errors. As each estimate of position is relative to the previous one, errors are cumulative.

Error All inertial navigation systems suffer from integration drift[citation]; Small errors in the measurement of acceleration and angular velocity are integrated into progressively larger errors in velocity, which are compounded into still greater errors in position. Since the new position is calculate from the previous calculated position and the measured acceleration and angular velocity,these errors are cumulative and increase at a rate roughly proportional to the time since the initial position was input. Therefore the input ha to be corrected. For terrestrial use the inertially tracked velocity is intermittently updated to zero by stopping, the position will remain precise for a much longer time, a so-called zero velocity update.

3 Specification of System

The task is to create a stand-alone system, embedded system, that fit the requested performance and quality set by Cargosafe System owner's representative. Here, criteria and expectation for further investigation are presented.

3.1 Technical requests

The design must be based upon a datalogger, capable of storing motion data from sensors to a portable device. A list of requirements from Cargosafe are as follows:

- The system *must* be of 6-DOF or more.
- NO loss of sensor data, and error free communication.
- Data *must* be stored locally.
- Samplings frequency of minimum 20 Hz, 50Hz and above preferred.
- Good accuracy from sensors
- Compact, easy to use design.
- Must be Battery operated.
- Emphasise on low-power design.
- Operation time of minimum 1-week.
- Offer selectable resolution and bandwidth.
- Re-configurable parameters during field operation.
- Remember last parameter settings.

The design and requirements for the product is set in collaboration with Cargosafe. Further reading is going to present the hardware and software developed to meet the expectation of Cargosafe.

This is the guidelines worked out in collaboration with Cargosafe. Further investigation and design will are based on these demands, and the thesis will present the outcome of trying tho reach these goals.

3.2 Ideas for system design

3.2.1 Embedded systems

Embedded systems is a special purpose computer system/board, which encapsulates all the devices such as processor, memory, interface and control in a single package or board to perform only a specific apllication task. Wherever the microcontroller is used it's embedded computer

3.2.2 Datalogger - DAQ

DAQ or data aguiciton device are very favourably for this task. The purpose of a datalogger or DAQ is to retrieve and store the data from sensors over a period of time. It is often a necessity when there are needs to analyse complex situations, process large amount of data, diagnose an error. The motivation for the design of this device is the challenging task of designing a complete system. The idea of combine hardware research with development of own circuitry to implementing software is intriguing.

There are several operation modes that can be considered for the DAQ.

-1: IDLE - Measure every channel every 3 sec, then write/display
-2: INSTANT - Measures a channel as given in a command, then report back immediately.
-3: BURST - For given channel. period and sample count, take a burst measurement and report when finished.
-4: PROGRAM - For a given channel and period, take measurement every fixed period and report back until stop.
-5: CONDITION: For a given condition on a channel, take measurement and report back.

From these options the following approach wil be the use of continuous logging. The disadvantage here is it demand a lot of memory and will consume a lot of power running continuously.

There are three demands to achieved to make a successful data-logger. Typically, data loggers are very simple devices that contain just three basic parts:

- Sensor
- Microcontroller
- Powersupply

Sensor or the sensors task are to measure an event. What to be measured is due to the task; Humidity, temperature, light, voltage, pressure, fault occurrences etc.

Micro-controller or the processors task is to collect the data from the sensor and store this information. Information usually is stored as time-stamped data sets. For example, if you have a data-logger that has to measure some process over a period of time, it vil both record the sensor data and the time of the event detected. This data can be sent to other mechanism through UART for user analysis.

Power supply are crucial for the data logger to function. Typically, and for this device a battery is used. And it of course important that the battery will last the duration of the measurements.

3.2.3 MEMS based sensor design

MEMs sensors are System-in-Package solutions, delivering high resolution and low power consumption, in an extremely small size. In consumer marked, multi-axis MEMS solutions are mandatory since consumers want to activate any function from any initial physical position. In a handheld application, there is no constant frame of reference. The new trend is to integrate multipple sensors - accelerometer, gyroscopes and magnetometers - into one single package, leading to leaps in functionality and performance in a wide variaty of application besides motion monitoring.

Besides the combination of a 3-axis accelerometer with gyroscope has enabled the creation of Inertial Measurement Units, devices that track and deliver complete information of the type, rate and direction of motion of any object.

MEMS designers are consetrated on developing 'smart sensors'. This is sensors with integrated processing capabilities, that makes them able to run sensing-related algorithms independently of main processor unit. This can reduce the power consumption at the system level, which is crusial in battery driven portable devices.

3.2.4 Powersupply

An idea is to create a device can be powered from both a USB supply or different battery-technology.

Operating from battery means that the voltage regulator must be extremely efficient with low bias current consumption, which means that it should draw almost nothing while there is nothing connected to the regulator at its output load. The battery device should sleep as much as possible to save energy. Since there are several power sources, there must be a way to choose the active one. This can be made through a jumper to avoid error.[10]

A good power supply is essential subject to microcontrolles, it can be seen as the heart that gives life to the controller. By not having an adequate power supply to the microcontroller it can lead to unexpected result or behaviour from the circuit. When driving the

clock-speed of the controller to higher the execution time, we need more power. There is a balance between speed and power consumption. Next to voltage is the amount of current needed to make the microcontroller work properly. But nowadays, the needed direct current on AVR's are in a minimum range consuming only current in size of µA.

One of the basic main thing to be considered when designing a microcontroller circuit is the maximum current that is allowed to pass through the microcontroller. Pass beyond this limit will certainty damage the controller. So, to avoid this, one may keep sure that the sum of microcontroller operation current and the load put together on the I/O ports does not exceed this value.

It is a good practice to run the microcontroller at the half of maximum current allowed by ins electrical specification and make sure that sourcing and sinking on ports within its electrical specification.

3.2.5 Battery power

Today's modern battery is evolve to be more compact yet more powerful to provide the power required by the device. A great widely available battery for the device is the NiMH (Nickel Metal Hybide) rechargeable. By using 5 of these in series you get a 6V supply which gives a great amount of power to source the project, dependent on the operating frequency of the controller.

LiPO battery on the other hand offers nearly twice the energy density(Wh/kg) of NiMH battery packs. It maintains its voltage during high loads. This is a good choice providing high capacity at reduced space.

3.2.6 Voltage regulator

Voltage regulators must be considered on how effective they are regulation the voltage. Low idle current and effective regulating is mainly in focus.

3.2.7 Wired communication via USB

For debugging purposes and setup, a option of connecting the unit to a computer must be supplied. By not selecting a wireless module, complexety is reduced and the current consumption is highly reduced. Using a USB-to-Serial solution like the FT-232 seem like a good approach, keeping extra hardware to a minimum and introduce a widely used interface to computer. A RS-232 would do the same task, but as very few computers offers serial interface, a USB-Serial controller seem more useful.

3.2.8 Microcontroller

Micro-controllers are getting smaller, faster and more powerful. It contain a CPU, program memory, computer memory, timers and GPIO on a single chip. However, this is often on sacrifice of performance and flexibility to achieve a simplified design and lower the cost, even if the performance is continuing growing. [51, Onshus,5.13.3]

The ATMega1284p is a good brain for the system. It offers a good amount of I/O and has all the connection needed for the basis in this thesis. The selection is based on supply of USART, I2C, SPI, Interrupt_Vect, External Oscillator, GPIO. Watchdog timer can wake the chip from sleep mode and directly execute an interrupt, thus not re-starting the program from the very beginning.

3.2.9 SD-card and FAT file system

A microcontroller can be connected to a MMC/SD card to store or read data from the memory[50]. However, a memory card without a fil system is simply an array of memory bits that can be read or written. If the card is plugged into the computer, the computer would not be able to do anything with it. Introducing a FAT library, the computer can be able to read the data stored by the microcontroller and read its content. On the WEB there exist a set of open-source code for FAT file system. Here, we can take advantage of the FatFS library created by elm. The only issue by introducing a FAT functionality is that it uses considerable memory, and one need a chip whit considerably resources.

3.2.10 Firmware

A Windows 7 64-bit with use of JTAG would be a good choice to program and download code into the AVR controller. Atmel toolchain is highly available, and compiler and software is offered for free.

On the controller a state-machine should be implemented so the datalogger can be controlled by the user.

3.2.11 Maintenance

As a part of the design and especially under running the instrument there may be need for calibration. Some reasons for this are:

- The measurement is a part of a safety system
- Reduce the amount of failure and improve reliability
- Assurance of quality

- Customers demand, governmental demands like Directory for Measurement when buying and selling

During operation it will be necessary to control the needed maintenance during operation. This is to ensure correct operation, avoid failure of sensors, reduce the overall cost. Plans for preventive and corrective maintenance has to be worked out. Examples of required periodic maintenance can be:

- Adjusting zero-reference
- Testing for correct operation
- Control response
- Validate the data output

Further investigation of maintenance can be read in [51, Chapter 17]

4 Sensors

4.1 MEMS and Digital feature

In general, gaseous dielectric capacitors are relatively insensitive to temperature. Capacitance sensing therefore has the potential to provide wider temperature range of operation, without compensation, compared to piezoresistive sensing. Capacitive sensing allows for response to DC accelerating as well as dynamic vibrations. That introduce a wider range in application to be used in. For a better understanding of design and the principle of capacitive sensing device the reader is urged to read Siliconde Designs technical note on the subject[30].

Common for these sensors are small size, high resolution, low power consumption, high reliability, signal conditioning circuitry and integrated functionality. The size of MEMS sub-components is in the range of 1 to 100 micrometers and the size of MEMS deveice itself measures in the range of 20 micrometers to a millimetre. It has features like[32]:

- Very small size, mass, volume
- Ultra low power consumption
- Small number of external components needed
- Low Cost
- Easy to integrate into systems or modify
- Small thermal constant
- Can be highly resistant to vibration, shock and radiation
- Batch fabricated in large arrays
- Improved thermal expansion tolerance
- Wast temperature range i.e -40 to +85 °C

The reason for selecting sensors with purely digital interface is to avoid extra ADC hardware an eliminate errors and noise that are introduced using an ADC device.

Having a direct digital output from the sensor allows direct connection to a inexpensive microprocessor without requiring analog-to-digital converter. A substantial insensitivity to electromagnetic interference is also provided compared with low level analog signals[31].

4.2 Accelerometer Sensor

In simple term an accelerometer sense how fast something is speeding up or slowing down. There are practical limitations in accelerometers in how fast they respond to changes in acceleration and can not respond over a certain frequency range. Due to mechanical design the sensor has a mechanical resonance frequency, and may be sensitive to vibrations an shock. The accelerometer can be of a single- or multi-axis design where it measure acceleration in one, two or three orthogonal axes. It measure the direction and magnitude of the proper acceleration, making it possible to sense orientation, coordinate acceleration, vibration, shock or free fall.

Accelerometer that uses differential capacitors to measure g-force cut the largest share of the market today. The measurement is converted into volt or bits. Newer development have made it possible to produce sensors with wider bandwidth[52].

Most common modes are:

- Inertial measurement of velocity and position.
- Sensor for inclination, tilt or orientation in 2 or 3 dimensions.
- Vibration or shock sensor.

There are countless applications for accelerometers, some examples are; levelling sensor, inclinometers,vibration detectors, vibration isolators, G-force detectors,alarm system, motion detection.

The design of the accelerometer makes it only possible to measure force only in a single direction, or axis of acceleration. This mean that you can only know the force along one direction of motion. So if the acceleration is perpendicular to the direction of the sensor, the unit would be oblivious to it. It is therefore necessary to have a multi axis device so the resultant acceleration can be decomposed into every axis.

4.3 Gyroscope

A gyroscope is used for measuring angular velocity. They are available in configurations that measures 1, 2 or even 3 directions.

Rotation is a measure of angular rate motion. This mode differs from the others because rotation may take place without any change in acceleration. The datalogger must measure rotation in addition of the other forms of motion, an is therefore included as sensor.

In systems with lower precision the flywheel in a gyroscope can be replaced with a vibrating mass at a reduced price and size as described in 3.8.1.

4.4 Magnetometer

This is a instrument for measuring strength and direction af magnetic fields. In addition, it can determine the magnetic fields orientation and direction. This sensor can be used as a compass for providing heading parallel to the surface of the earth. The calculate force of the magnetic field is done through what is called magneto-resistive sensors. The important thing is to be aware that these sensors does not provide the compass direction, but presents measured magnetism that can be used to calculate a direction.

Some of the aspects to encounter for when using a magnetometer as a compass will be enlightened here:

Magnetic declination It should be widely known that compasses points directly toward the magnetic north and not the true north. This is also known as magnetic declination. How much this vary depends on where you are and it also changes over time. The declination can be found on charts or use numbers from a geophysical data center.

Tilt Problem Traditional compass have to be held entirely flat to function. In this thesis we will later on encounter this problem as the algorithm used here is only based on X and Y axis of the earth's magnetic field. When sensor is not in parallel to these axis the amount of magnetism felt by the sensor will change based on the alignment of the sensor is to these axis. More advanced algorithms can resolve this.

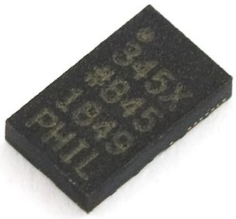
4.5 Selected Sensor Components

4.5.1 Analog Device - ADXL345

General description[33]
ADXL345 is a small, thin, low power 3-axis accelerometer and can offer high resolution. It outputs 16-bit twos compliment data that can be access through either SPI or I2C[SPI,I2C bus][40]. The producer present this sensor suited for applications measuring static acceleration of gravity in i.e tilt sensing application. But it is also well suited for measuring dynamic acceleration from motion or shock. With the high resolution of 4 mg/LSB enables measurement of inclination less than 1.0 °. Changing the g-range of the sensor increases the resolution so that the 4mg/LSB scale factor is maintained.

Functions functions are provided with the sensor, making it capable to detect absence of motion, tap sensing, free-fall or user level thresholds. It holds low-power modes.

Accelerometer can be configured to low-power mode saving additional energy, but at the cost of introducing slightly more noise to the measurements. In this thesis a bandwidth of 100Hz is mostly use. At this bandwidth the current consumption in normal mode will be 145µA compared to only 65µA in low power configuration[33, Table6,Table7]. Setting it to standby even lower current consumption to 100nA



Features
-Small 3mmx5mmx1mm LGA package
-4 mg/LSB resolution.
-Supply voltage range: 2.0V to 3.6V
-Start-up time 1.4ms -Low power: 145µA
-Temperature range: -40 to +85 °C -Power consumption scaled automatic with bandwidth -Good zero g bias stability

4.5.2 ITG3200

General description[34]
InvenSense ITG-3200 is a single-chip, digital-output, 3-axis MEMS gyro. It features enhanced bias and sensitivity temperature stability, that would reduce the need for calibration. Three 16-bits analog-to-digital converters outputs data from the gyro, along with user selectable internal low-pass filter bandwidth and Fast-Mode I2C interface. They offer a high performance, low noise and a low-cost semiconductor. The range is fixed to $\pm 2000 \frac{^\circ}{7} sek$.

Features
-4mmx4mmx0.9mm QFN package
-14.374 LSB $\frac{^\circ}{7} sek$ with full-scale range of $\pm 2000 \frac{^\circ}{7} sek$
-Supply voltage range: 2.1V to 3.6V
-Start-up time 50ms -Small power: 6.1 mA, Sleep mode: 5µA
-Individual power down on each axis. -FIFO buffer that can be used to reduce timing requirements

-Data collection on polling or interrupt
-Temperature range: -40 to +85 °C -Temperature sensor

4.5.3 KXTF9

General description[35]
The KXTF9 is a 3-axis MEMS accelerometer with integrated orientation, tap and activity detecting algorithms. Using sensing based on principle of differential capacitance one gain advantages like common mode cancellation, reduced errors from process variation, temperature and stress[35]. It uses a I2C interface for communication to the chip. This flexibility allows for easy system integration by eliminating analog-to-digital converter and providing direct communication to the microcontroller. Offering a selectable sensitivity of 8- and 12-bit, it produce high resolution of data. A drawback is the sensitivity changes with the selected range of the device, spanning from 2mg/LSB at 2g to 31mg/LSB at 8g setting with a 12-bir resolution. Another downside is that it seem to be a little noisy, which will be expressed in a later chapter.

Features
-4mmx4mmx1.45mm LFCP package
-Variable resolution, 2mg/LSB at best
-Supply voltage range: 1.8V to 3.6V
-Start-up time 2.5-80ms -Current cunsumption 700 µA, 290 µA on low resolution, Standby: 100nA
-Enhanced bias and sensitivity temperature stability
-Temperature range: -40 to +85 °C -Temperature sensor -I2C running at 400kHz

4.5.4 IMU3000

IMU-3000 offers a IMU solution with 6-axis sensor fusion. Embedded is a 3-axis gyroscope and a Digital Motion Processor hardware acceleration engine with a secondary I2C port for interfacing a third party accelerometer to deliver a complete sensor fusion output. Both linear and rotational motion are combined into a single data stream. Intensive motion processing computation can be offloaded the host processor by the provided sensor fusion output. It features multiple programmable full-scale ranges, low rate

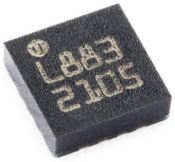
noise performance. The initial sensitivity is factory-calibrated and reduces the calibration requirements. Other qualities are 16-bit analog -to-digital converters, digital filters, precision clock, programmable interrupts and a low power consumption.

General description[36]

Features
-4mmx4mmx0.9mm QFN package
-Supply voltage range: 1.8V to 3.6V
-Start-up time 20-100ms -Current cunsumption 700 µA, 290 µA on low resolution, Standby: 100nA
-Enhanced bias and sensitivity temperature stability
-Temperature range: -40 to +85 °C -Temperature sensor -I2C running at 400kHz

4.5.5 HMC5883L

General description[**?**]
This is a three axis magnetometer/digital compass. The communication is simple and all done through the I2C interface. It offers high-resolution, magneto-resistive sensors with aoutomatic degaussing strap drives. It got offset cancellation and a 12-bit ADC for high-resolution earth field sensing. This enables 1° to 2° compass heading accuracy.

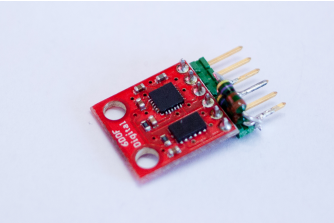


Features
-3mmx3mmx0.9mm LCC package
-Simple I2C interface
-2.16V-3.6V VDC Supply range
-Low current consumption: Measurement mode µA, Idle 2µA -Build-In self test
-High output rate, 75Hz continuous -Wide magnetic field range $\pm 8 Oe$ -5 milli-gauss resolution
-Temperature range: -30 to +85 °C

Dimensions: 17.78x17.78 mm

4.6 Sensors Breakout

4.7 6DOF - Sparkfun IMU



IMU Digital Combo Board from Sparkfun has the ITG3200 and ADXL345 mounted. Sparkfun offer these sensor mounted on a simple breakout for easy integration and use. They have used I2c as the communication channel and have broken out one interrupt pin for each of the sensors. Features are:

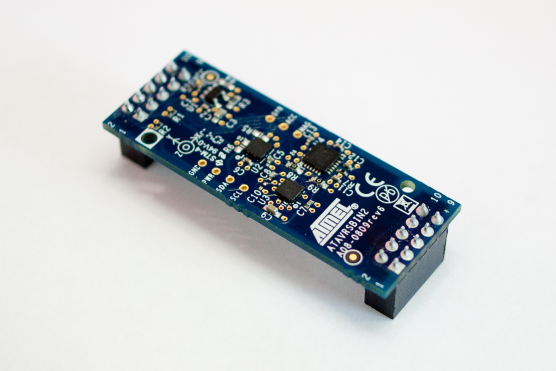
- Tiny!
- Mounting holes
- ADXL Accelerometer
- ITG3200 Gyroscope
- 3.3 V input
- I2C interface

Arduino forums offer great support for each of these sensors, they are widely used which gives a better support and makes it easier to get other peoples experience with these sensors. Sparkfun have good support releasing all the information they have of the sensor-board including schematic, Eagle files, a small tutorial and the components datasheets.

4.8 ATAVRSBIN2 - Atmel Sensor xPlained 2

ATAVRSBIN2 is a part of the Atmel xPlained series of evaluation and development tools they offer. They state that the Inertial Two System Board delivers a full 9 degree of freedom sensor platform using the Honeywell HMC5883L magnetometer, Kionix KXTF9-1026 Accelerometer and InvenSense IMU3000 Gyroscope. All of them 3-axis of motion sensing as described above.

Atmel have a lot of information and documentation of the sensor board offering hardware set-up to their xPlained series of development boards and detailed schematics and layout of the sensor board. The board itself has a regulator, reducing the risk of damaging the



sensors. 3 interrupt pins are broken out making it possible to use the interrupt feature within each of the sensors. Advance operation or utilization of the features of each and single sensor is possible.

All of the information and the device datasheets can be downloaded for free from Atmel Sensor xPlained webpage. In addition they have a great customer support who will answer questions regarding set-up, design and implementation.

The downside using this board is that the KXTF9 Accelerometer have been experienced noisy by several users, introducing the need for filtering the signals. And by adding a filter, it will most certainly affect the operation and band of the sensor[53]. Another difficulty is that Kionix have next to nothing of support on their own homepage making it harder to use the sensor since application examples or code are hard to find.

5 Design Consideration

5.1 MEMS Based Sensors

Saturnation There are infinite number of rates and motions at which you could turn or move the MEMS sensor within the design parameter. The output of the digital sensor will be quantezised and have a resolution range according to the ADC. If the output is saturnated the output will indicate incorrectly. For example, if the gyro is rated to 250 deg/sek and it is rotated at 400 deg/sec, it would saturate. The same occur for an acceleromater exceeding upper limit.

Resolution Connecting the output of a sensor into the input of an analog-to-digital converter would give you a digital output that would correspond to the measured output from the sensor, but it will be outputted in discrete (digital) steps so the the microcontroller can interpret and understand the information. Analog to digital converters are defined by their word length, or the maximum number of bits they will output for each reading of the sensor the bigger the word length, i.e., the more bits, the better the resolution of the information being converted from analog to digital. A 8-bit ADC would present the range in $2^8 - 1 = 255$ steps while a 12-bit ADC would produce $2^12 - 1 = 4095$ steps.

Sensitivity Motion-sensors are also rated for their sensitivity, or the degrees per second, g-force or magnitude of magnetic field that corresponds to a given output voltage. This greater sensitivity would result in greater precision when used in the calculations used later on to create actual angles from the rate-of-rotation.

Analog to digital conversion

Standard ADC resolutions are usually in a range of 8, 10, 12 or 16 bits. When the ADC chip samples an analog signal at its input, it divides the signal into a number corresponding to the chips word or bit structure design, then outputs that information digitally so that for instance a microcontroller can read the data. An 8-bit ADC divides the input signal into 256 parts (low resolution), a 10-bit ADC is 1024 parts, 12-bit is 4096, and a 16-bit chip divides the signal into 16,384 parts (very high resolution). The amount of resolution needed is a function of the application one are working on.

5.2 Specifications

A gyroscope can be identified by the following basic specifications:
Digital Interface

- Offer different power modes
- Smart embedded features(free-fall, sleep/wake-up, direction detection, thresholds)
- Interrupts
- On-board filters and reconfiguration

Measurement Range The maximum angular speed with which the sensor can measure, typical in degrees per second

Number of sensing axes Multiple configurations as mentioned earlier.

Non-linearity Non-linearity is measured as a percentage error from a linear fit over the full-scale range, or an error in parts per million.

Working temperature range Most electronics works only in some range of temeratures. Often a onboard temerature sensor is integrated in the design so there is no need to worry about temperature related calibration issues.

Shock Survivability Will give a sense of how much force the gyroscope can withstand before failing. MEMS a very robust, and can withstand lasge shocks without braking.

Bandwidth Is a indication on how many measurements it can do per second. Quoted in Hz. The bandwidth is all application depended and a thought of what the requirement for the application. The bandwidth of an accelerometer indicate how often a reliable reading can be taken.

Angular Random Walk (ARW) This is a measure of gyro noise and has units of deg/hour1/2 or deg/sec1/2. It can be thought of as the variation (or standard deviation), due to noise, of the result of integrating the output of a stationary gyro over time. So, for example, consider a gyro with an ARW of 1/sec1/2 being integrated many times to derive an angular position measurement: For a stationary gyro, the ideal result - and also the average result - will be zero. But the longer the integration time, the greater will be the spread of the results away from the ideal zero. Being proportional to the square root of the integration time, this spread would be 1 after 1 second and 10 after 100 seconds.

Bias Here there are multiple of categories like bias drift, or bias instability. The bias, or bias error, of a sensor is the signal output from the sensor when it is NOT experiencing any motion. Even the most precice sensors available have error sources and bias is one of these errors. Bias can be expressed as a voltage, bit-value or a percentage of full scale output, but essentially it represents a rotational velocity, acceleration or wrong heading. Again, in a perfect world, one could make allowance for a fixed bias error. Unfortunately bias error tends to vary, both with temperature and over time. The bias error of a MESM sensor can be caused by a number of factors:

-calibration errors
-switch-on to switch-on
-bias drift
-effects of shock (g level)

Individual measurements of bias are also affected by noise, which is why a meaningful bias measurement is always an averaged series of measurements.

Bias Drift This refers specifically to the variation of the bias over time, assuming all other factors remain constant. Basically this is a warm-up effect, caused by the self heating of the sensor and its associated mechanical and electrical components. This effect would be expected to be more prevalent over the first few seconds after switch-on and to be almost non-existent after (say) five minutes.

Bias Instability Bias Instability is a fundamental measure of the 'goodness' of a gyro. It is defined as the minimum point on the Allan Variance curve, usually measured in /hr. It represents the best bias stability that could be achieved for a given gyro, assuming that bias averaging takes place at the interval defined at the Allan Variance minimum.

5.3 Sampling

Usually most physical signals exist in analog form. A needed and attractive alternative is to convert the signal from analog to digital form and use accurate and convenient digital ICs to perform digital signal processing. The DPS can perform a variety of arithmatic and logic operations that implement a filter algorithm. Mostly the filter does the same task as the analogfilter perform - namely eliminate interference and noise. An adventage is corruption of amplitude of the analog signal represented as digital binary pulses is of no consequence. Analog signals are sensitive to noise. While the digital ones has greate imunity.

5.4 Signal Quantization

Signal converted to digital form are subject to quantizaion. Any analog value outside the steps of quantization will be rounded to the nearest binary integer, and is calle quantization error. Higher order binary values will reduce this effect increasing the resolution. [38]

5.5 D/A

The most important caracteristic with the D/A converter is the resolution. It is the smallest value the output signal can be changed by and is given by the lowest number

in sequence. If the converter has n-bit the resolution becomes 2^{-n} multiplied with referenvce voltage. Called LSB (Least significant bit)

The positive/negative value is signed by use og 2s complementary. The MSB is inverted to represent positive or negative values.

5.6 EMC and noise

Electromagnetic interferrence can have unvanted effect on the equipment. The device is sensitive to high emission from other sources. This error is reduced by the use of mems digital components reducing the analog coupled error. The magnetometer must be placed far away from subject that emit electromagnetig noise.

It is important that the device does not

- Disturb other systems
- Is not prone from disturbnace from other systems
- Does not induce noise on its own component.

5.7 ESD

The circuit is put together of several delicate components. Care should be taken not to damage the circuitry. Main problem with ESD is destruction of single electrical components, error in functionality or false databits. Hard-error is simple to solve as the solution will be raplacement of component, easy to lokalize and simple to get the system up an running.

Soft-errors like false data-bit is more delicate. Errors might be random, non-permanent and gives a mutual connection of errors. Usually it is hard to point out a singel error componant or part of system that is non-functioning. Components are weakened and degredading so a product can pass a test and functionality controll and quicly fail afterwards.[38]

5.8 Communication

Always if possible never use a uC for sourcing of an external chip/unit. The uC is better at draining than sourcing. A number of the I/O pins will be used when the controller is used for transmission. When communication is not needed, the bus should be set to a idle state or deactivated to conserve energy.

5.9 Power supply

Important parameters are:

- Linear regulation - sensitivity input voltage
- Load regulation - If load changes, how much the voltage changes

Linear regulator There are three types; Standard, LDO(low dropout) and Quasi LDO The important difference is the drop-out voltage between these. I.E voltage difference experienced between input and output for a satisfactory regulation.

Key features are

- Simple
- Cheap
- Low noise
- Good regulation
- Sufficient efficiency
- Common used

Important issues to consider when selecting regulator are
-Max current in circuit
-Type of power-source
-Required Precision
-Idle current
-Special functions

LDO is best suited for battery applications, has a relative high ground pin current and a low V_D.[38]

Switch mode regulator Consist of a inductor, capacitor and a diode added with PWM transistor. It has a high efficiency often 80-95 % grade of efficiency. It is very flexible where voltage can be regulated up,down, in. Very useful for battery applications as the voltage can be generated to a higher level.

Here a Boost set-up is selected meaning that th voltage is up-converted $v_out > v_in$. The downside is that switched regulators are prone to be noisy and often has a output rippel of 10-100mV. Ripple is experienced in supply. And the switching function in the inductor may result in emission of EMI [38]

Problem with the linear regulator is how they work makes them extremely inefficient. By taking the difference between the input and output and burn the excess voltage as waste heat. So the larger the difference is, the more heat is produced.

5.10 Batteries

NiMH batteries that are rechargeble are good for sourcing an application. They are both economicaly and more environmentaly friendly than regular batteries. A downside is that they discharge even if their not in use. And has low capasity compare to size.

Alkaline batteries is prefered for a longer run time. They sustain current over several years of storing. Long lasting at midium(high wattage consumption.)

Lithium resist cold very well. Are fresh for several years. But expensive and source low current.

Batteries has to have easy access for easy mantenance and changing.[39]

5.11 Sensor interpretation

We will need a way to interpret the data from the sensors into readable values for the processor. By checking the data-sheet for each of the devices, we can get an understanding of the different types of value the sensor will produce. Often it is enough to use a case baset guess of the data values. But when the resolution is high, the accuricy of the sensor is more important. In reading a sensor there are simply 3 steps to account for:
-Gather sensor data
-Graph sensor data
-Generate line equation.

There may be a need to use different equations based on which operation on the region the sensor have. Issues may be:
-Non-continuity: The sensor may not work properly out of its range or small range.
-Non-linearity: Due to the construction, the sensor may output the data on a non-linear scale. Producing a measurement that doesn't grow with the applied outer influence. Usually occure on the max, min readings of the sensor.
-Sensor Noise: Five readings in the same position and exact same situation could give five different outputs. It is nesesary to verify the amount of sensornoise there is. One way is to do multiple readings and average the noise away. Has to be tested for the appropriate environment.

Last issue is to know how many datapoint to record. The more non-linear the sensor is, the more reading should be done. The more reading the more accurate, but there is certant a balance!

6 Available Hardware

6.1 MEGA-1284P Xplained

The Atmel MEGA-1284P Xplained evaluation kit is a hardware platform for evaluating the Atmel ATmega1284P microcontrollers. The MEGA-1284P Xplained contain one QTouch button sensor, three mechanical buttons, four LEDs, 3 analog sensors, a UART to USB bridge. Headers provide easy access to analog and digital I/O pins. The board can be powered by USB cable and equipped with the standard 10-pin JTAG header for debuggers.

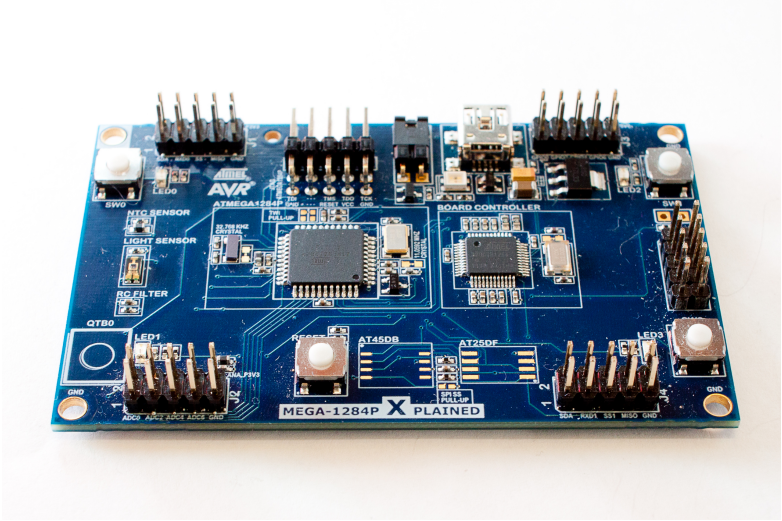


Figure 2: xPlained1284P development board

The 1284P microcontroller is selected as it offer great power and functions needed for the datalogger. This controller is also known to Cargosafe Systems, which gives them a system based on a known platform for further development.

Key Features
-Atmel ATmega1284P microcontroller
-Read temperature sensor with the ADC
-Read light sensor with the ADC
-Use the Atmel QTouch library to detect touches on the touch button
-Use PWM as input to an analog filter and ADC to measure the filter response
-3 push buttons to interact with the microcontroller
-4 LEDs to show status information

-Board controller that routes data from the ATmega1284P UART to the PC via USB
-Program the kit via bootloader or an Atmel programmer
-Expand the board with Xplained top modules

The xPlained1284P board controller offer USB-interface using the virtual USB AVR software. The board controller is connected to TDX1/RDX1 offering a USB interface directly to the board without extra cabling except for the USB power supply. But this solution is avoided for a reason. Each time the device is flashed, the xPlained1284P card randomly disconnects from the virtual com-port and some times it is necessary to unplug and reconnect the USB cable. This becomes frustrating in the long run, during testing of software and reprogramming. Irritation from un-plugging the cable, the FT-232 interface is added, giving a stable 'COM-port' to work with.

6.2 FT-232 USB-to-Serial

The FT232 USB-to-Serial Breakout board is based on FTDI's popular IC. The FTDI chip has an internal oscillator an EEPROM. The VCCIO in this shield is tied to 3.3V, but can be re-solder to cope with 5V configuration. To ensure the device is recognise by the computer, the serial FTDI driver must be installed. This can be downloaded from: http://www.ftdichip.com/Drivers/VCP.htm

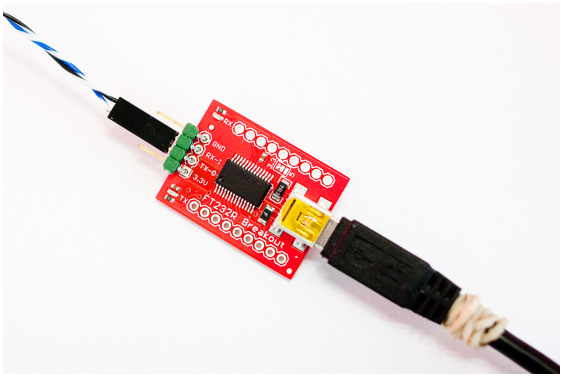


Figure 3: Breakout - FT-232

This device is used with the ATMega1284P running @ 11 MHz, and UART over terminal @ 115200bps without any problems.

Features:
-Implements full v2.0 USB protocol
-Needs no external crystal
-Internal EEPROM for device ID and Product Description strings
-Royalty-Free Driver support for Windows, Linux, and Mac OSX

6.3 OPENLOG

Openlog ia a plug'N'play device which works at the instant it is connected. The device can be reconfigured for baudrate or data sampling using the guidelines supported on Sparkfun's web-pages. The openLOG from Sparkfun is an open source data logger. This is a serial datalogger that simply just works, and fit nicely into this embedded project. This device is bought as a backup for the datalogger in the thesis because the ELM[50] library for the FAT16 interface proved to be harder to implement then expected. This datalogger does what the developer describe it will. Connect power, add Rx/Tx line from UART, and the datalogger will simply start logging on reception on the *RX channel. A micro-SD card has to be formatted with either FAT16 or 32 configuration. On power up, this little device will configure itself from a standard configuration file it makes. In this file it is possible to render the baudrate, commandos so forth. OpenLog firmware is open source and is based on Bill Greiman's sdfatlib. It support both FAT16 and FAT32.

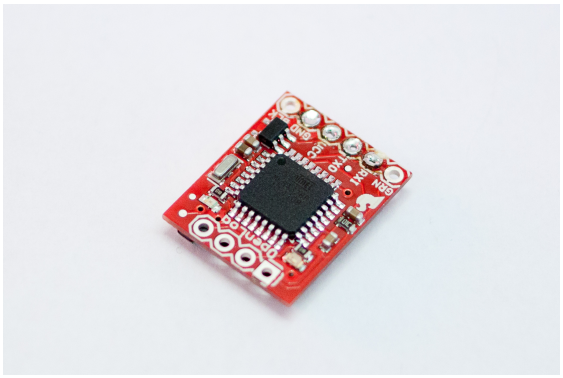


Figure 4: Breakout - OpenLOG

7 PCB Board design

During the thesis, several board designs are created and tested. As not all of the components were available "off the shelf", own circuitry design has to be made. This to adapt to the required specification of a low-power design, and for prototyping several solutions for a final design. All the boards presented here are designed using the Eagle CAD 6.1 Software[46] and PCBs are produced and assembled in the electro-component workshop at NTNU.

Both the schematics and PCB board were designed using Eagle PCB (v6.1.0) by CadSoft. This is a grate piece of software giving free use with some restrictions. The CAD files from the software can be implemented directly into the millingmachine, creating PCB.

7.1 ATAVRSBIN2 Socket

ATAVRSBIN2 is made to fit directly on top of the J1 and J2 header of the xPlained1284P board. But, putting this sensor board on top of these sockets conflicts with the use of RDX0 and TDX0. An obvious assumption would be using the RDX1 and TDX1, but as the 32UC3B1256 hogs these pins an atempt using these resulted in failure. Even resetting the board controller on xPlained1284P does not prevent error in transmitting over UART on RDX1/TDX1. A solution could be physical cut the connection between the board and board controller, or make an alternative solution. The solution became creating a separate socket board for the sensor-stick. Another reason for doing this due to the SD-Shield, as mentioned further below, has to use the SPI bus. Connecting the ATAVRSBIN2 directly to the J1 and J2 header will interfere with the SPI bus as the interrupt pins from the sensor-board will be plugged into the bus pins which would create errors during operation. Schematic is provided in Appendix B 31.

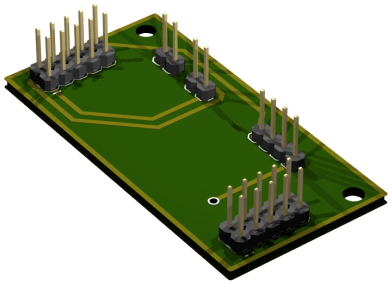


Figure 5: Socket for ATAVTSBIN2

7.2 Power board

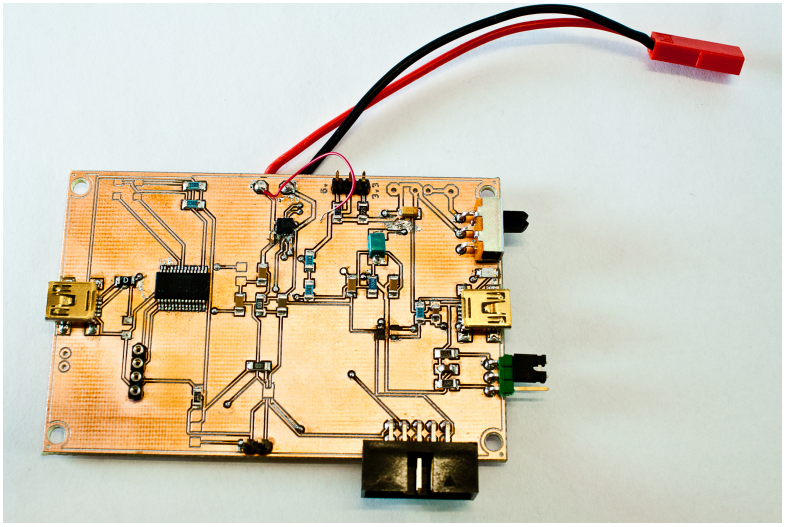


Figure 6: Power Board PCB

As mentioned earlier battery power to the xPlained1284P is needed to get it running as a stand-alone unit. The xPlained has a 3.3V regulator on the input, however the diode bridge on the USB port is directly sourced from the USB-port. Specification of this bridge upper voltage limit could not be found. Documentation from Atmel states that the board has to be supplied from either USB port or a 5V supply. [23]

For this reason, a power circuit is created. Because LiPO is selected as a preferred source of energy, the solution becomes to create a entire circuit offering supervise, charging and stable power supply from this battery. The LiPO is rated to 3.7V $4,2$V when fully charged Since the xPlained board on-board voltage regulator is 3.3V, the voltage from

the battery has to be boosted to ensure correct and stable operation for the on-board regulator.

7.2.1 Power management

7.3 BUCK-BOOST regulator

The xPlained Kit need at least 4.6V supply to power up its own regulator. Since the LIPO battery only supply 3.7 volt nominal, it is nessesary to boost the power. This is done by a buck boost converter. To avoid the battery get deep-discharged a voltage divider is set to the UVLO pin to assure the right cut-off voltage. The caculation of selecting the righet value is done by:

$$R3 = R4 \times (\frac{V_{inmin}}{V_{uvlo}} - 1) \tag{15}$$

From the datasheet(TPS61200, pg?) the R3 resistor must lie around 250kΩ and the wanted cut-off voltage is 3.2V (Using the equation the value of R3 then becomes 2.3MΩ

TPS61200 regulator is selected for this purpose, this is a Low-Input Voltage Synchronous Converter. It is specified to give $> 90\%$ efficiency at 300mA. With the possibility of output 600mA current at 5V, a quiescent current less than 55µA and a operation range from 0.3V to 5.5V makes this regulator a strong candidate. It also holds qualities like shorcircuit load protection, over-temperature protection and power save mode.[20]. The circuit is build based on the recommendations stated in the components datasheet. Mainly focused on the criteria given in *Programming the- Output Voltage, UVLO Threshold Voltage and Inductor Selection.*

Resistors for regulating the output voltage is selected by using the formula:

$$R_1 = R_2(\frac{V_{out}}{V_{FB}} - 1) \tag{16}$$

,where R_1 and R_2 are adjustment resistors and assuming V_{FB} is usually close to 500mV. According to the schematic, it is made possible to switch between 5V and 3.6V output. Closest resistor values available is $R_1 = 1.2M\Omega$ and $2M\Omega$ using $R_2 = 220M\Omega$. Using these values inserted into the equation should give a voltage output of 5.045V and 3.6V.

The cut-off voltage is set by using:

$$R_3 = R_4 \times (\frac{V_{inmin}}{V_{UVLO} - 1}) \tag{17}$$

Internal reference threshold V_{UVLO} is typically 250mA. [20, page 15]. By using the hardware considerations from the datasheet along with the redundancy in not overdischarge the LiPO battery. The R_3 value is selected to be 2.7MΩ. The schematic 32 shows R12 to be 2M, however this is changed before the assembly.

Inductor of 4.7µH is selected, and is valid to the maximum rating from the datasheet.

A MCP73811 component from Microchip is chosen for the charge circuitry. It provide a specific charge algorithms for single cell LiPO batteries to accheive optimal capacity in a short charge time. It is well suited for applications charging from a USB port. The idea is to integrate charging from a ordinary USB-B port for the LiPO battery. Doing this, the user would not need to disconnect the battery when it is drained, and there is no need for having a separate charger available. The chips requirement for a supply voltage between 3.7V to 6V makes this a good candidate for this purpose.Charger circuit is designed by following the recommended layout and specifications given in the ICs application note[21].

Providing a jumper and two resistors rating 10KΩ and 2.2KΩ at the program pin, the charge current can be set to either 454mA or 100mA. The theoretical values are calculated by:

$$I_{REG} = \frac{1000V}{R_{Prog}} \tag{18}$$

MAX4372 Current Sensing Amplifier is considered for battery management and supervision since it is supposed to perform well with a charger circuit. But the MAX17041 is selected as the best suited candidate for battery supervision. MAX17041 is a compact 1S/2S fuel gauge, and is good for track the battery's relative state-of-charge. It uses a I2C interface to provide estimated and measurement data. It has a precision of ±12.5mV up to 5V. Working in a supply voltage range of 2.5V to 4.5V makes it perfect for the circuitry. Typical current drain is only 50µA making it well suited for a low-power application. Again, the circuit design are according to the design considerations presented in the components datasheet[22]. Supply to this chip is fed back from the regulated output of 3.3V from the xPlained board.

7.3.1 Communication

FT-232 USB-Serial chip, mentioned earlier is now integrated on the PCB. The IC has proven to work like a charm, and has been very useful when debugging data from the xPlained board or change setting through the Terminal menu which will be presented in chapter 7!!!!!!!XxX Minimizing the use of external components and cable that can lead to errors or faulty operation, the FT-232 is provided on the board. Only a small part of the FT-232 feature are used, so the routing is kept to a minimum. The Voltage from the USB is not bypassed as there has not been any test yet it the board is capable of be sourced via USB and the FT232 chip still will work properly. The I/O lines of the FT-232 have 3.3V. The device can be powered directly from the battery through pin4, but to avoid spending extra unneccesary energy from the battery, the device is powered from USB, and will only operate when needed. Pin 17 - Pin 4.

Schematics are a fusion of the Sparkfun FT232 dongle and the circuity presented by the producer. The device is powered from a separate USB port and is isolated in that way that only TX/RX from controller are connected to the IC. This mainly fore reducing power consumption by not let the Ic be powered from the board. Since it would be stupid to run the IC in idle when not used. A idea have been to interface the USB connector to the charger with the USB connector to FT232 interface. However, due to lack of testing of stability merging these together and the cost and fear of damaging the FT232 chip these circuits have been left separated from each other.

7.3.2 Performance and Test

Powering up the PCB close to successful, but there are a few issues and deviations to consider. The problem with the layout arises when securing the xPlained1284P board to the power board. Placement of the USB connectors are unfortunate, the placements of the communication sockets lead to unnecessary long wires.

A practical test of the charger IC show a deviation in measured and calculated values. A multimeter is coupled in series between the battery and the power board, while swap the jumper between the two programming resistors, input current to the battery is measured. The rest of the circuit is at the time being not subjected to any load except the buck-boost regulator. From table 2 the measured and theoretical values becomes:

Resistor	Measured	Theoretical
2.2KΩ	614mA	454mA
10KΩ	139mA	100mA

Table 1: Output Current LiPo-Charger

Unfortunately, upon testing a simple inattentive moment forgetting to switch from current to voltage when measuring led to a short-circuit of the LiPO IC and the whole chip fried.

The FT-232 works like a charm, and no error are found during testing. The only change to be done is on the computer where a new driver for this FT232 chip has to be installed. No exceptions compare to the Sparkfun breakout of the FT-232 can be found.

TSP61200 boost regulator proves to be hard to function properly. This is mainly due to the package size and the problem is attached to hardware. The drawback and cause of failure is the tight space between each pin making it real hard to both solder and position correctly. Usually a re-flow oven is a better tool for soldering these packages, but the only tool available at the moment are fine soldering iron and paste. From picture below 7, the TSP61200 is compared to the tip of a 0.5mm Pentel P205 pencil. The output becomes unintentional and weak connection of pins, which further result in incorrect operation and instability. Failure of getting this IC to work in the circuit, the solution is to re-design the circuit and use a larger, easy to handle component.

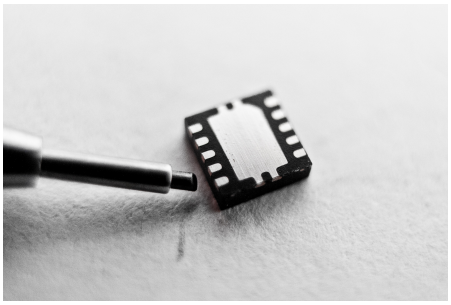


Figure 7: TSP61200 Package Size

7.4 Powerboard v2 - Uruz

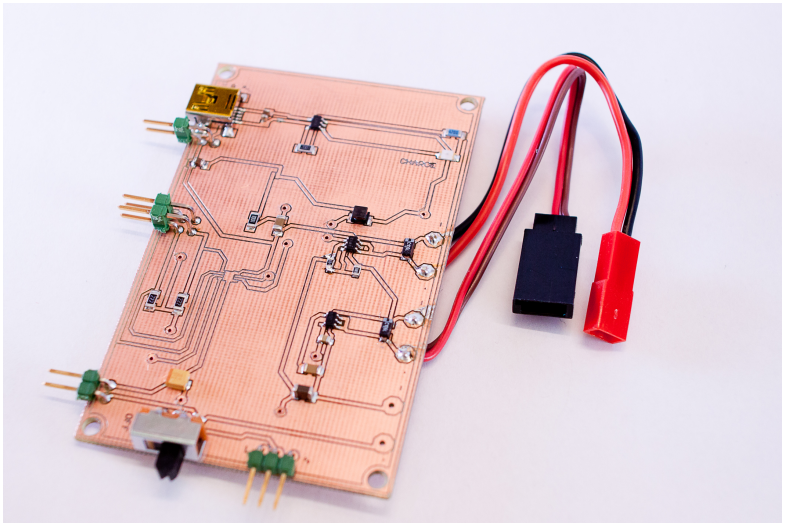


Figure 8: Power Board PCB

A redesign substituting the TSP61200 is done, see Appendix B 33. Two suited candidates are found, both having pros and cons. These are the MCP1252[26] Low noise, Posirive-Regulated Charge Pump and the MCP1640B[27] Synchrounous Boost Regulator. A short comparison chart gives the following values:

The 6-Lead SOT23 package of the MCP1640 has less pins but need more external inductor component compared to the 8-lead MSOP package of the MCP1252. MCP1640B offers higher current output, but the component itself drains more current.

To be on the safe side and not knowing how much current the final unit need, the selection falls on MCP1640B. Schematic is drawn according to FIGURE 6-2 [27, ,page 19]. The resistor values between V_{out}, V_{FB} and GND are found by:

$$R_{TOP} = R_{BOT} \times (\frac{V_{out}}{V_{FB}} - 1) \tag{19}$$

Parameter	MCP1640B	MCP1252
Output VoltageΩ	2V to 5.5V	1.5V to 5.5V
Output CurrentΩ	350mA	120mA
Quiescent CurrentΩ	200µA	60µA
AccuracyΩ	$\pm 3\% V_{out}$	±2.5%
Temperature RangeΩ	-40 to +125 °C	-40 to +85 °C

Table 2: Output Current LiPo-Charger

using $V_{out} = 5V, V_{FB} = 1.21V, R_{BOt=330k\Omega}$. R_{TOP} then becomes 1033kΩ the closes valid value for a resistor is 1MΩ. A 10µH inductor is used, and capacitors selected by TABLE 5-1 in [27]

7.4.1 Adding NiHM battery

Introduced to the circuit is the LDO regulator for use with NiHM batteries. The main reason for this is that substituting the TSP61200 with the XXX, makes it not possible to use the shut-down mechanism as mentioned above

To provide the datalogger with battery power during testing and experiments, a NiHM battery is quite durable, rechargeable and more robust compared to the LiPO One could use a widely used and common LM7805 5V regulator, however the datasheet [24] show that the voltage should be 7V or above to ensure a stable power supply. Since the NiHM battery is of 6V and measuring by multimeter gives 6.42V when fully charged the option is to get a Low-Dropout 5V Regulator. LP2985L is selected. This is a micropower 150mA Low-Noise Ultra Low-Dropout regulator. It is able to output 5V $\pm 3.5\% V_{NOM}$ and a drop-out of typical 120mV at 50mA current load [25]. Using only 15µA while running, makes this is a perfect candidate.

7.4.2 Other changes

As currently redesigning the MCP73811 is replaced with the MCP73831. The rating of these are more or less the same with the exception that the latest is specifically design for LiPO application, that be algorithms and performance.

The FT-232 is removed from this board, so that the board is mainly acting as a power supply/power management. It will be added to the sister-board made in parallel with this new power-board. MAX17041 chip is kept, as this now is crucial after the exchange of the LiPO boost regulator. This chip must be used to avoid over-discharge of the LiPO battery so it is not destroyed during operation or worse may cause fire.

7.5 Communication board - Anzuz

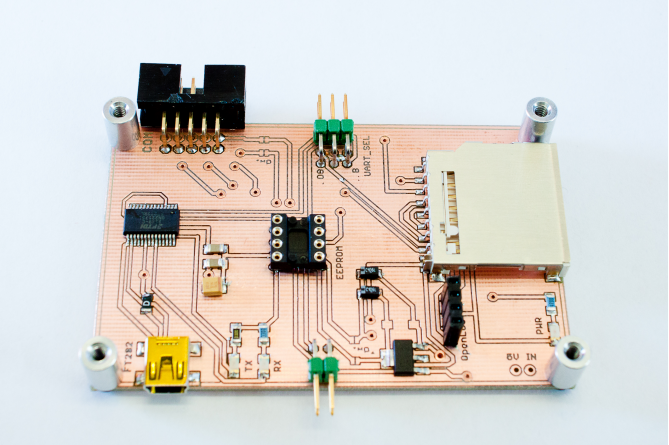


Figure 9: Communication board

Communication or storage is the essence of the datalogger. If there does not exist any option to communicate or store the data gathered from the sensors, the system is useless. The purpose of this board is to gather all the communication options at one level both for easy overview and clear interface. Here, openLOG, SD-card, FT232, EEPROM, UART RX/TX and I2C bus is made available for easy access. Pull-ups are added to the I2C bus according to the I2C design specification[45] A prototype offering a variety of interfaces may prove useful in further development of the system. By making this PCB, cabling is down to a minimum, minimizing errors caused by wrongly connect wires and other issues related to hardware wiring. The schematic is provided in Appendix B 34

7.5.1 EEPROM

An EEPROM is added to gain the possibility to store configuration data, calibration values and more. Here, the Microchip 24FC1025(1024K bit) is mounted in a 8-DIP socket making swapping to other EEPROM easy. The EEPROM is compatible with the I2C bus and can be cascadable for up to four devices. A standby current consumption of only 100nA and read current of 450µA makes it suited for low power device. The 24FC is chosen as it offers 1MHz clock frequency. But the 24LC might be appropriate as for now the I2C bus is running on 100kHZ. A long lifetime is offered by 1 million erase/write cycles. The A2 pin is wired to Vcc as required in the specification[28], while the A0=1 and A1=0 set in hardware.

7.5.2 SD-SHIELD

As there exist no library for the SD-shield used in this circuit, the component has to be drawn in Eagle. This is done using the mechanical layout[29] provided from the designer company. SD card has to be provided with a SPI bus interface, this is done through the communication socket and is taken directly from the J1 jumper on the xPlained1284P board. Using a SD-card instead of the OpenLOG device is preferable, as it would be easier to control the dataflow and design the system.

7.5.3 On-board Regulator

-3v3 regulator instead of supply from xPlained board A 3V3 fixed NCP1117LP Low-Dropout Voltage regulator is provided to the board. This is mainly presented as an option to sourcing the circuit through the xPlained1284P board's J1 header. The idea is to be capable to source the communication board through the powerboard if needed, and if a lot of components are added, this 3V3 regulated powersupply can provide the needed power instead of powering devices through xPlained1284P. However, one should avoid using this regulator if there are no need for the regulator. The reason is that the Quiescent Current is 500 µA, and it would be a waste of power if using it without the need.

7.5.4 OpeLOG Socket

For simplicity, a shield is placed to connect the OpenLOG breakout board without use of wires. By doing this, both reduced excecc wiring and the option of securing the breakout firmly to the board.

7.5.5 LEDs

In addition some LEDs are provided for user indication. The communication board is given a Green power led to indicate power in connected to the board, and some LEDs on the FT-232 circuit is added for visual indication of TX/RX communication.

7.5.6 Jumpers Connection

Last, some header-pins are provided for connecting other hardware like the I2C outlet, and for status signal from the SD-Shield like card insert and write protect status. These two signal can be routed to available free GPIO on xPlained1284P board and used for SD-card status. A 2x3 pin header makes it possible to route the TX/RX connection from the 1284P controller to either the FT232 or OpenLOG hardware.

7.6 Total System

As for now the datalogger is separated into four pieces, five if the battery is accounted for.

7.7 Proposed Redesign

Proposal for a compact solution is created, based on earlier circuitry and experience. By merging the communication and power featuures togheter removing some of the excess hardware. The 3D-model of the board can be seen down below, schematics are provided in appendix B35

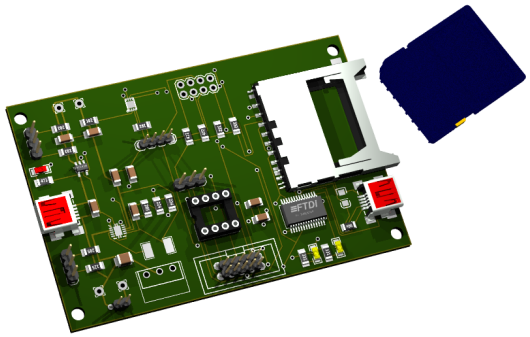


Figure 10: Communication board

8 System Software

8.1 AVR Studio 5.1

AVR Studio 5.1 from Atmel is used as development tool. Atmel have included a lot of support for their development/demo kits. Flashing of the xPlained1284P board is done through this software using JTAGICE MkII for tool. To add float number support to the compiler additional libraries have to be included. This is done as follows:

GOTO:

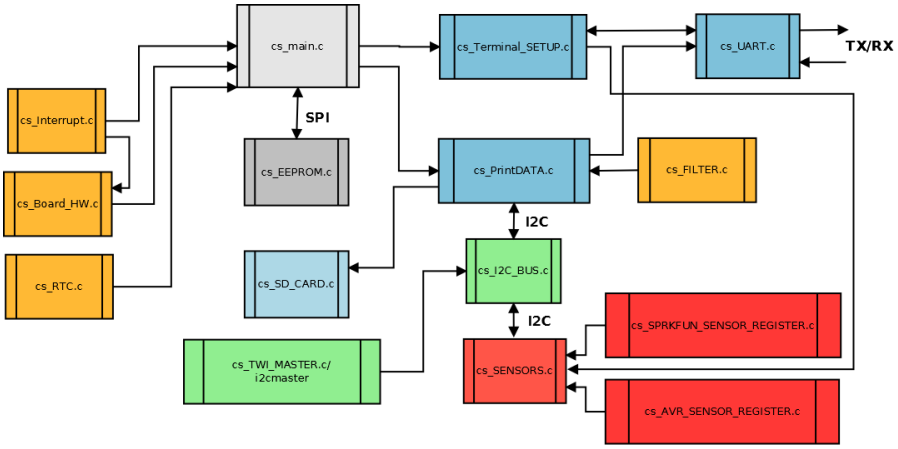
Project Properties → Toolchain → AVR/GNU C Linker → Libraries
In *Libraries (-Wl, -l)*; add the following: libprintf_flt.a and libm.a

Pjoject Properties → Toolchain → AVR/GNU C-Linker → Miscellaneous
In *Other Linker Flags:* ; type in: -Wl,-u,vfprintf -lprintf_flt

This have to be done so that the float numbers can be displayed through the *printf()* function. It is related to the triogonometric functions used in th micro-controller software, or derivation of g-force.

8.2 Software Overview

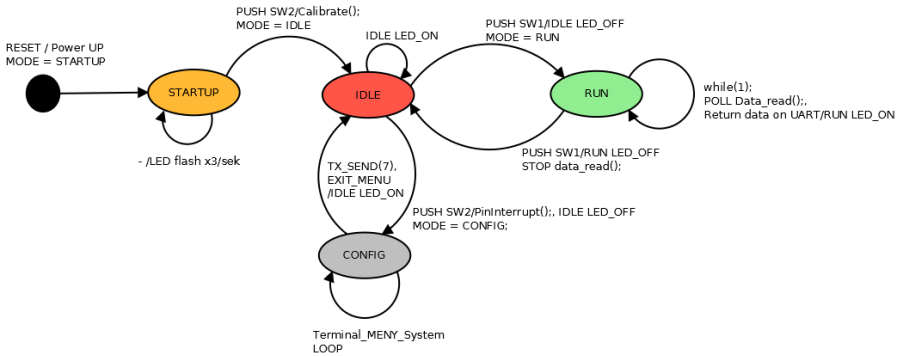
Developing software, care has been taken to separate the methods and functions in a most logical manner. During programming an approach of maintaining a transparency and ease of use between parts of the software is crucial for flexibility and a clear functional use. For this reason the software have been divided into groups like *Sensor, Bus, Communication, Interrupt, Hardware I/O etc.*.



Generating software and debugging to get it working with all the device has taken a considerably amount of time comparing to hardware design. The main reason for this is the challenge of creating a useful and simple interface to all the sensors making it user friendly and flexible during testing. It is crucial to read the sensors datasheet thoroughly, and the quality of the documentation varies between the producers. Debugging of register manipulation and bit-wise operation has taken at least three times more than expected. This is because issues regarding configuring and reading from sensors must be tested during operation to confirm if the data-output is sensible. Some faults have only revealed themselves during post-processing of RAW data. This adds to the total software development time, as corrections must be made and new test must be preformed.

8.3 State Machine

A state-machine is used to jump between the set of states on the unit. For the time being there are 4 states, *STARTUP, IDLE, CONFIG, RUN*. The different states are physical indicated by LEDs mounted on xPlained1284P board, and jumping between states are preformed by use of buttons available on the board.



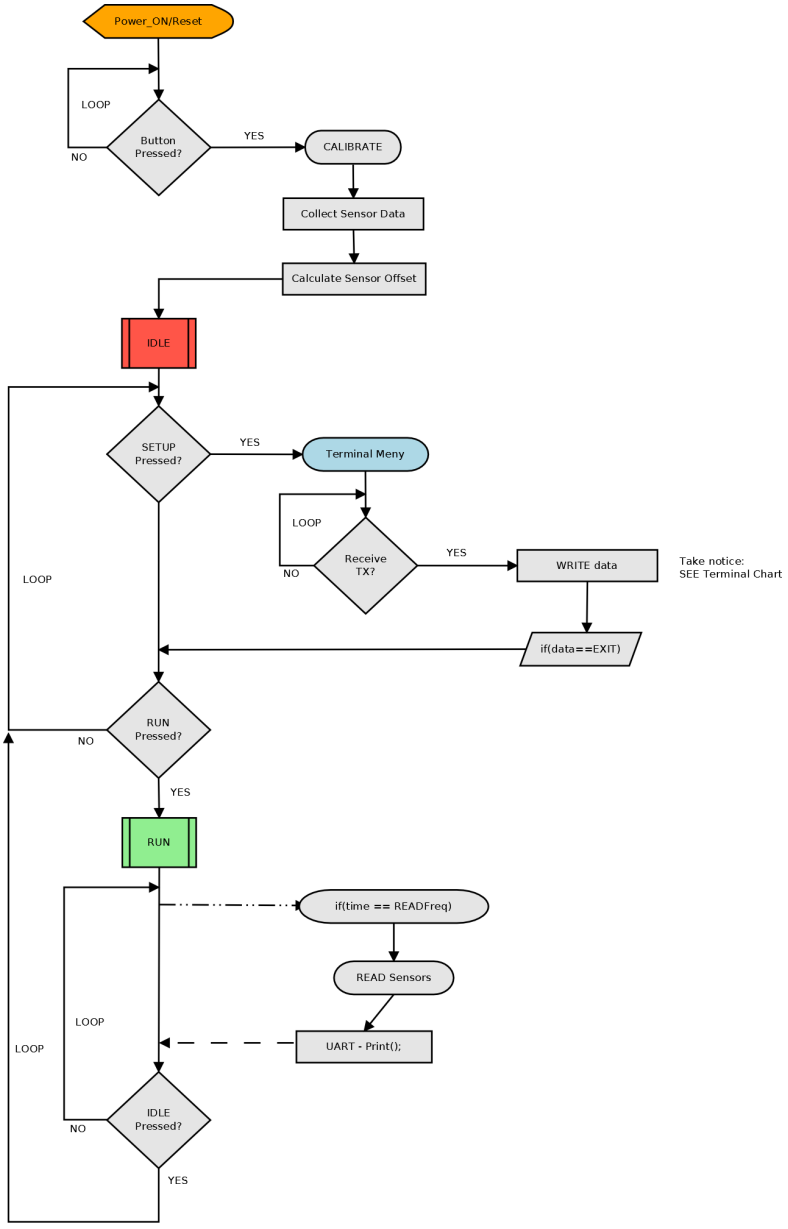
8.4 Initialization

All the initialization is done at the top of the main routine. At this moment all the required or wanted functionality has to added when flashing the controller. The reason not to make initialization of the parts of the system possible through terminal commando is that for a fully developed solution it would not be necessary to change the basic functionality or sensors used. Initialization starts all the needed parts of the system which need to interact. This separation makes it easy to change or deactivate functionality. This makes debugging easy as software parts can be deactivated to eliminate errors from other parts of the system.

STARTUP	During start-up the device must be secured to the desired position, and placed firmly in place. After power up, pushing the start-button will set the unit in STARTUP-mode. 1 Second after pushing the button the sensors will collect 50 data readings over a time period, which then again is processed and averaged. These data are stored for zero-point adjustment ensuring the sensor offset bias are removed.
IDLE	As the name says the unit is idle. The sensors are put to idle, port-pins changes states to reduce the power consumption. In IDLE the unit can be putted into RUN- or CONFIG-mode.
RUN	From default, all the sensors will be read and displayed on the UART. The run mode is by default simply polling data from the sensors at a fixed frequency. RUN mode will be affected by the changes done in CONFIG-mode.
CONFIG	Through terminal11.2 the unit can be reconfigured over serial/USART communication. A simple menu is displayed in terminal and gives access to change the default values on the sensors. Polling frequency, display option, and filter can be set.

8.4.1 main.c

Here all the initialization of the systems software parts are done. All of this is done at the top of the *main()* routine. State-machine and the Interrupt Service Routines are implemented in this file.



8.4.2 cs_UART.c

USART is used for communication from the micro controller to peripheral hardware like the USB-Serial or SD-card reader. To ensure the correct operation and output on UART the UBBR registers have to be sett accordingly to the formula given in the 1284P datasheet. Here the UART is used in Asynchronous Normal Mode (U2Xn=0).[1]. The

[1]Be sure to set the CKDIV8 = ”0” or the frequency of the internal RC oscillator is divided by eight giving frequency of only 1MHz

correct value is obtain by the following equation:

$$UBRRn = \frac{f_{ocs}}{BAUD \cdot 16ul} - 1 \tag{20}$$

The initialization then becomes,

```
1  void uart_init(uint32_t UART_BAUD_RATE)
2  {
3      /*Set baud rate */
4      UBRRH = (uint8_t)(UBRR(UART_BAUD_RATE,F_OCS)>>8);
5      UBRRL = (uint8_t)UBRR(UART_BAUD_RATE,F_OCS);
6      /*Enable receiver and transmitter */
7      UCSRB = (1<<RXEN)|(1<<TXEN);
8      /* Set frame format: 8data, 1stop bit */
9      UCSRC = (1<<USBS1)|(3<<UCSZ00);
10     // Binding printf with uart_send and receive
11     fdevopen(uart_send, uart_receive);
12 }
```

Here we decided to use 1-stop bit and a 8-bit data format. This is highly logic using a 8-bit microcontroller and all the sensor registers are 8-bit. The TX/RX is obtained using the TDX0 and RDX0 which can be found on the J1 port on xPlained1284P board. An effort is made trying to use the TDX1,RDX1 as will be mentioned in the hardware section. But due to hardware difficulties the solution became using the RX0 TX0 for UART communication. From the ATMEGA1284P documentation, the Table 19-11 [8] show that using the on-board oscillator of $f_{cpu} = 11.059200MHz$ @ a baudrate of 115200 bps should give an error of 0%.

The maximum error on the UART channel is said to be half a bit duration(0.5). From the initialization of the UART the maximum packet time can be found by: $1startbit + 8databits + 1stopbit = 10bits$. The error is sorted out with the following calculation $\frac{0.5}{10} \times 1000\% = 5\%$.

8.4.3 cs_I2C_BUS

The general operation of the last project[40] I2C routine is re-written, but it is still based on the cs_TWI_MASTER.c and i2cmaster.c files provided by *Peter Fleury* based on the Atmel Application Note 300. SPI compared to I2C has a few advantage when considering speed and simplicity. But some of the sensors selected5 does not support SPI interface. Another problem is getting sensors that offers same clock flank and polarity. This is set by the manufacturer, and there are no standardization. A seperate chip-select pin is also needed for each sensor. Because of this, the I2C bus interface is a preferred selection. I2C use the same routine for all the sensors, making it easier to introduce new sensors into the system. The addressing compared to the use of GPIO for /cs is also a huge gain in this matter. Our test setup use as many as 5 sensors, needing 5 extra pins for chip-select. A solution could been using a few pins connected to a MUX,

but that would add the cost of the device, increase complexity, and introduce one more component that could fail.

Initialization is simply done by setting the following registers:

```
1  void i2c_init(void)
2  {
3      /* initialize TWI clock: 100 kHz clock, TWPS = 0 => prescaler = 1 */
4      TWSR = 0;                           /* no prescaler */
5      TWBR = ((F_CPU/SCL_CLOCK)-16)/2;  /* must be > 10 for stable operation */
6  }/* i2c_init */
```

The read/write protocol are found in cs_I2C_BUS.c and are set according to the specifications given in the sensors documentation. The communication protocol is provided in table3,4, 5 in the Data Format section below.

8.4.4 cs_Board_HW

This .c file control the user buttons and LEDs. To swap between the different states in the state-machine, the buttons included on the xPlained board are utilized. These must be activated through software and this is done with the cs_Enable_Button() function:

```
1  #define LIGHT_PORT   PORTB
2  #define LIGHT_DDR    DDRB
3  #define SWITCH_PORT  PORTB
4  #define SWITCH_DDR   DDRB
5  #define SWITCH_PIN   PINB
6
7  #define SW0 2
8  #define SW1 1
9  #define LED0 0
10 #define LED1 3
11
12 void cs_enable_button(void)
13 {
14     LIGHT_DDR |= (1<<LED0) | (1<<LED1);
15 //Deactivate SETUP_LED
16     PORTB       |= (1<<LED1);
17     SWITCH_DDR  |=  (0<<SW1);
18     SWITCH_PORT |=  (1<<SW1);
19 }
```

Included in this file is also mechanism for LED control and blinking.

8.4.5 cs_Interrupt.c

Interrupts are here used for sensing a change on I/O pins like push of a button or sensing an*Interrupt* from a sensor. The timers interrupt is used to control timing mechanism

like reading frequency of sensors, time-stamp events or adjust the flash control for LEDs. To get these working they are activated like:

```
1  void cs_enable_PinInterrupt(){
2      EIMSK |= (1<<INT1);
3      PCICR |= (1<<PCIE1);
4      PCMSK1 |= (1<<PCINT9)|(1<<PCINT10);
5      //(1<<PCINT8);
6  }
7
8  void cs_enable_TimerInterrupt(){
9      //TIMSK1 |= (1<<TOIE1);
10     TCCR1B = 0x05; //Prescaler clk/64;
11 }
```

All of the sensors offer new_data_event interrupt on pin. So the sensors can offer reading when a new data is ready.

8.4.6 cs_SENSORS.c

All the code for the sensors used are included in this file. The sensors are set to a default value when initializations is called. Registers entries are created in the cs_AVR_SENSOR_REGISTER.h and cs_SPARKFUN_SENSOR.h files.

```
1  /*********************************************************************
2  /* SPARKFUN 6-DOF SENSOR BOARD ↩
                                                                       */
3  /*********************************************************************/
4  void cs_ADXL345_init(){
5      //Set the power register, Set the sensor i measure mode and no sleep options
6      cs_Sensor_write(ADXL345_POWER_CTL, ADXL345_MEASURE_ON, ADXL345_WRITE);
7      //Set the data format/sensitivity, set to 10-bit mode
8      cs_Sensor_write(ADXL345_DATA_FORMAT, ADXL345_RANGE_4G, ADXL345_WRITE);
9      //Set the bandwidt and output data rate; by default 0x0A(100Hz)
10     cs_Sensor_write(ADXL345_BW_RATE, ADXL345_SRATE_200, ADXL345_WRITE);
11 }
12
13 void cs_IGU3200_init(){
14     //Sample rate divider, 0x10 is selected by random only for testing! Approx 5↩
           ms per sample
15     cs_Sensor_write(ITG3200_SMPL_RATE_DIV,0x47,ITG3200_WRITE);
16     //FS_SEL MUST be set to 03h for proper operation!, DLPF_CFG set to 0 (↩
           Filter_BW 256HZ, Sample Rate 8kHz)
17     cs_Sensor_write(ITG3200_RANGE_LP_FILTER,0x18,ITG3200_WRITE);
18 }
19
20
21 /*********************************************************************
22 /* ATMEL 9-DOF SENSOR BOARD ↩
                                                                       */
23 /*********************************************************************/
24 void cs_KXTF9_init(){
25     cs_KXTF9_Stop();
```

```
26      cs_KXTF9_DATA_REG_WRITE(ODR100);                //Set data output rate(baud)↩
            100Hz for the low-pass filtered acceleration
27      cs_KXTF9_CNTR_REG_WRITE(0,KXTF9_RANGE_2G);      //No Interrupt on new data, ↩
            G-range: 4G
28      cs_KXTF9_Start();
29  }
30
31  void cs_IMU3000_init(){
32      cs_Sensor_write(IMU3000_DLPF_FS, 0x12, IMU3000_WRITE); //00001011 100de/sek,↩
            96Hz
33      cs_Sensor_write(IMU3000_USER_CTRL, 0x08, IMU3000_WRITE);
34  }
35
36  void cs_HMC5883L_init(){
37      cs_HMC5883L_CONF_REG_A(HMC5883L_AVG_SMPL_0,HMC5883L_BAUD_75HZ,↩
            HMC5883L_MEASURE_Normal);
38      //cs_Sensor_write(HMC5883L_CONF_REG_B,0xA0,HMC5883L_WRITE);
39      cs_Sensor_write(HMC5883L_MODE_REG,0x00,HMC5883L_WRITE);
40  }
```

8.4.7 IMU_init()

IMU3000 has a special feature allowing a 3rd part Sensor to be read through the gyroscope. The values from the 3rd party accelerometer are stored to the IMU3000 register, and the IMU3000 can be configured to output all of the gyro and accelerometer data in a single read operation.[19] This feature can be further utilized by enabling InvenSense on-board sensor fusion feature.

```
1   void cs_IMU_init(){
2       //Sample rate 1Khz, Filter bandwidth 42Hz, Gyro range 500d/s
3       cs_Sensor_write(IMU3000_DLPF_FS, 0x0B, IMU3000_WRITE);
4       //Set accelerometer register data address
5       cs_Sensor_write(IMU3000_AUX_BURST_ADDR, 0x06, IMU3000_WRITE);
6       //Set the accelerometer i2c slave address
7       cs_Sensor_write(IMU3000_AUX_SLV_ADDR, KXTF9_DEVID,IMU3000_WRITE);
8       //Enable pass-through mode
9       cs_Sensor_write(IMU3000_USER_CTRL, 0x08, IMU3000_WRITE);
10      //Set accelerometer cntr register in measure-mode
11      cs_Sensor_write(KXTF9_CTRL_REG1, 0xC0,KXTF9_WRITE); //Set this to 0x80 do ↩
            disable 12-bit resolution and use 8-bit, G-range will be set to 2G for ↩
            both values
12      //Cancel pass through to accelerometer to enable gyro handle the reading
13      cs_Sensor_write(IMU3000_USER_CTRL, 0x28, IMU3000_WRITE);
14  }
```

8.5 I2C over SPI

Argumentation and a study of the communication buses available can be further investigated in[40]. But in this thesis it simply is a question of interface than best performance. As there are several sensor devices that has a unique address, it is easy to use the I2C bus.

8.6 Output data

One of the requirement is to read the data using the best resolution possible. The sensors selected outputs the data in 10-, 12- and 16-bits fashion. Since the sensors stores these data on 8-bis registers it is important to put these bits correctly together if data are to be valid. Multiple registers can be read upon one request over I2C, this works in the way that one send the device address followed by a read request. Then the data register with the lower adress is read. As long the sensor receive an ACK after the register is read, the sensor will increment to the next data register and let you read from this. This will continue until a NACK is send, terminating the READ session. When combining data it is important to keep track of which are high and low register, and which axis the registers represents. Following bit-shifting is necessary to get a correct readout.

```
1 ADXL345_READ = (int8_t)i2c_rawdata_HIGH[i]<<8 | (uint8_t)i2c_data_LOW[i+1];
2 ITG3200_READ = (int8_t)i2c_rawdata_HIGH[i]<<8 | (uint8_t)i2c_data_LOW[i+1];
3
4 KXTF9_READ    = (int8_t)i2c_rawdata_HIGH[i+1]<<4 | (uint8_t)i2c_data_LOW[i]>>4; ↩
      //LSB-first
5 IMU3000_READ  = (int8_t)i2c_rawdata_HIGH[i]<<8 | (uint8_t)i2c_data_LOW[i+1];
6 HMC5883L_READ = (int8_t)i2c_rawdata_HIGH[i]<<8 | (uint8_t)i2c_data_LOW[i+1];
```

MASTER	S	AD+W		RA		DATA		P
SLAVE			ACK		ACK		ACK	

Table 3: Single-Byte WRITE Sequence

MASTER	S	AD+W		RA		S	AD+R		NACK	P
SLAVE			ACK		ACK			ACK		

Table 4: Single-Byte READ Sequence

MASTER	S	AD+W		RA		S	AD+R			ACK		NACK	P
SLAVE			ACK		ACK			ACK	DATA		DATA		

Table 5: Burst READ Sequence

8.7 Default Settings

For simplicity during initialization and start-up, all off the sensors are set to values that are both preferred, and from a practical point values that seem to be useful from a practical point of view. Bandwidths are set to rating that would not conflict the samplings rate used during testing. It is preferable to have the settings stored in EEPROM but due to time-limitation, this deature is not created yet.

Defaults values are:

Sensor	Sensitivity	Bandwidth	Gain
ADXL345	2G	100Hz	NA
ITG3200	1000 deg/sek	100Hz	NA

Table 6: Default Sensor Values

8.8 Filter Implementation

A short implementation in software of the Moving Average and Low-pass Filter from Chapter?? are presented here.

8.8.1 Moving Average

```
int16_t cs_ACC_MOVAVG_Filter(int16_t *avg,int16_t measure, int size){
int sum = 0;
int average = 0;
    for(int i = size;i>0;i--){

    avg[i]=avg[i-1];
    }
    avg[0]=measure;
    for(int i=0;i<size;i++){
    sum = sum+avg[i];
    }
    average = sum/size;
    return average;
}
```

8.8.2 Low Pass

This is the implementation of the Low-Pass filter in software.

```
float filter = (2* halfT) / (Timeconstant * 2 * halfT);
ax = filter*acc\_x + (1- filter) * ax;
ay = filter*acc\_y + (1- filter) * ay;
az = filter*acc\_z + (1- filter) * az;
```

, where Timeconstant = 1/freq so for example 1/42 for 42 Hz low pass.

8.9 Integrating SD-Card to Microcontroller

By using the SPI interface, a SD memory card can be added to the circuit. It would be preferable to use a FAT32 file formate, as stored data can easily be accessed by inserting the memory card into a computer. Now one could have access to all the stored

data without any custom software. As a starting point, the portable "FatFS"library implementation developed by Elm Chan(2010) is used. Modifications are necessary to do to make it work on the platform selected in this assignment. Some code lines must be changed, and timer interrupt is needed.

A good guide how to perform these changes can be read from: http://www.basementcode.com/avr/sd_fatfs/fatfs This page is used as a guide for implementing the framework for the FatFS library.

Functions to perform operations on a memory card are provided by the documentation of the "FatFS" library. Example of use are:

```
1 //Mount device
2 f_mount(0, &myfat);
3 //Open/create file
4 f_open(&fil_obj, ``logfile.txt'', FA_WRITE | FA_OPEN_ALWAYS);
5 //Write to file
6 f_printf(&fil_obj, ' \%d',(int)sensor_value);
7 //Close file
8 f_close(&fil_obj);
```

9 Calculation and Calibration

9.1 Calculation of resultant acceleration

The magnitude for a single-axis accelerometer can be derived as $acc\_max = \sqrt{x^2} = x$

For a multi-axis device it becomes: $acc\_max = \sqrt{x_n^2 \cdots x_i^2}$

Force Calculation $Force\_gravity = -g * \cos(angle)$ (depended on starting axis of accelerometer)

if you reverse the equation you can calculate the angle by knowing the detected force: $\cos(\frac{sensor\_value * conversion\_constant}{-g})^{-1} = angle$

9.2 Angle calculation

The angle can be measured using both sensors by using a distinct technique. To measure the angle using the gyroscope, the signal has to be integrated. This is because the gyroscope is giving the angular rate, so a simple way to get the angle is have the angle rate multiplied by the time $angle = angle + \omega * dt$.

To measure the angle using the accelerometer, one have to sense the gravity on each axis of the accelerometer. The projection of gravity accelerating on each direction of the sensor give us an idea about the angle. A simple calculation is given by

$$angel_acc = \arctan \frac{A_i}{\sqrt{A_j^2 + A_k^2}} \tag{21}$$

A better angle calculation can be made by using both sensor properties. Gyro data will continuous increase as it experience drift, acceleration data are quite sensitive and change a lot in smal time steps. A complementary filter can be used to merge these signals togheter. The filter can be written as:

$$filteredangle = HPF*(filtered_angle+\omega*dt)+LPF*(angel_acc), where HPF+LPF = 1 \tag{22}$$

9.3 LOW pass filter

The low pass filter implementation on time-discrete form is based on a exponential-weighted moving average.

$$y_1 = x_0 y_t = \alpha x_t + (1 - \alpha) y_{i-1} \tag{23}$$

With the smoothing factor $0 < \alpha < 1$. This imply in the limiting case when $\alpha = 1$ the output is the same as the original measurement. So the closer the value is to 1 the greater weight is given to the recent changes in data while a value close to zero will give less responsive to recent changes.

9.4 Moving Average Filter

The simple moving average (SMA) is used to filter data output. This is done by taking the unweighed mean of the previous n data points. The code is based on the following formula:

Assume a sample space $\{a_1, \ldots, a_n\}$ The arithmetic mean A is defined via the equation.

$$S := \frac{1}{n} \sum_{i=1}^{n} a_i \tag{24}$$

The result is a moving average lags behind the latest data point by half of the sample width.

9.5 Calculate Force

RAW data read from the sensor must be translated into understandable readable data for the user. A good measurement is G-force. The derivation of force can be calculated by the formula:

$$A_g = A_r aw \times \frac{Range(g)}{Resolution(bits)} \tag{25}$$

9.6 Centripetal Acceleration

$$\frac{\Delta v}{\Delta t} = a = \frac{v^2}{r} \tag{26}$$

For a velocity of x m/s and radius x m, the centripetal acceleration is x m/s2.

9.7 Centripetal Force

The force called "center seeking" force is the centripetal force. This is the force directed to the center of the curvature of the path. Motion in a curved path represents accelerated motion.

$$F_{cntripital} = m \times \frac{v^2}{r} \tag{27}$$

9.8 Tilt-measurement

Tilt measurement can be done by using the result from the RAW data from the accelerometer converted to g-force measurement. Putting these measurement into the trigonometric formula.

$$\Theta = \arctan \sqrt{\frac{A^2}{B^2 + Z^2}} \tag{28}$$

Here the x-,z-axes and the y-,z-axes are used to derive tilt. Replacing the A and B with X and Y in the formula will give the angle. The orientation can be found by checking the quadrant which the Z-X-axis or Z-Y-axis lies in.

9.9 Calibrating Accelerometer

9.10 Calibration

The calibration should compensate for individual accelerometer offset, scale factor and misalignment to the orthogonal reference. Before calibration it is important to check that:

- The data output is stable and repeatable
- The conversion is correct

Thrusting the datasheet for correction and calibration is not reliable. Even if it promise exact sensitivity, in doesn't know how the sensor will be placed in the circuit.

Real calibration will give better results, but take some work to do.There exist a variety of calibration methods, but here I will discuss the one used in this task.

9.11 Offset correction

The effect of an offset on one of the axis will result in bad angle measurement. How large this effect can be can be shown by a simple example. Assume that the y-axis has

no offset and is pointing directly upward measuring $1g$. The x-axis is pointing along a horizontal axis at 0 °. However, the x-axis has a offset of $0.1g$. Based on simple a simple trigonometry equation

$$\theta = \arctan(\frac{A_x}{A_y}) \tag{29}$$

The resulted calculated angle would be 5.7105°, introducing an error of 5.7° in the measurement.

9.12 Sensitivity Mismatch

The main error component due to accelerometer sensitivity in a dual-axis inclination sensing application is when a difference in sensitivity exists between the axes of interest. If we again look at the two-axis example above, but now assume the solution is used with perfect offset trim on both x-,and y-axis. But in this example a sensitivity mismatch of 10% is introduced on the y-axis. The sensitivity on the x-axis is perfect. In a 1g field the x-axis will report the correct value of 1g, while the y-axis with sensitivity mismatch will report $1.1g$. Sensitivity mismatch varies over the entire range of rotation, and makes it difficult to compensate for the error after the inclination calculation. An attempt to skew the y-axis by varying the sensitivity will only introduce greater error.

9.13 Basic calibration Technique

As shown, combining both the sensitivity mismatch and offset error can give large errors in the result. This is not acceptable for inclination measurement and must be dealt with. With a simple calibration technique these errors can be suppressed to an acceptable limit. To accommodate this the sensitivity and offset should be calibrated, and this calibration should then be used to correct the inclination measurement.

When the accelerometer uses the raw values without compensation the output is similar as follow:

$$A_{out} = A_{off} + (GAIN \times A_Actual) \tag{30}$$

Implementation of these correction is done by the expressed formulas:

$$A_{out}[g] = A_{off} + (Gain \times A_{read}) \tag{31}$$

$$A_{off} == \frac{1}{2} \times (A_{+1g} + A_{-1g})(offset) \tag{32}$$

$$Gain[g] = \frac{1}{2} \times (\frac{A_{+1g} - A_{-1g}}{1g})(Sensitivety) \tag{33}$$

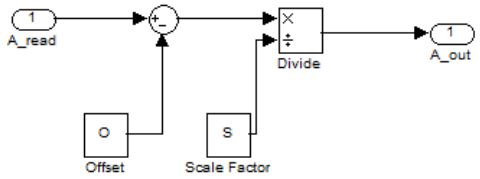


Figure 11: Model of the Offset and Sensitivity Correction

$$A_{actual}[g] = \frac{A_{out} - A_{off}}{1g} \tag{34}$$

The values are derived by measuring the positive and negative values along the X-, Y- and Z axes. A state machine according to the Freescale Application Note [9, Figure 3, Flow chart] is implemented to calculate the offset parameters.

9.14 Adjusting Compass Parameters

The compass can be programmed to operate at different gains setting. To get a more accurate reading based on the gain of the sensor and the deviation between the actual and true north some simple calculations can be performed.

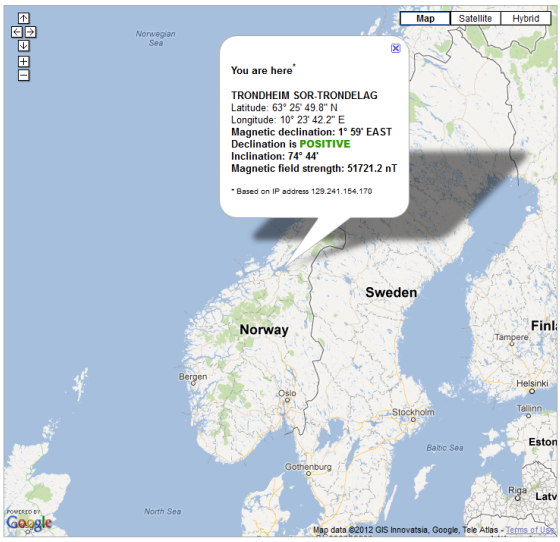
From the datasheet of the HMC5883L[**?**, Configuration Register B] a table shows gain compared to the digital resolution. Resolving to the Table 9 of the components datasheet and information of the deviation at the sensors location,better accuracy of the bearing can be achieved.

Default settings of the HMC5883L magnetometer during testing is $\pm 1.3Ga$ or 1090 LSB/Gauss giving a resolution of 0.92 $\frac{mG}{LSB}$.

The magnetic field around the earth is not perfect, and changes in ground and the core of the earth affect the readings. As known, the magnetic field will also change over time. This is what is called magnetic declination, and will effect the compass reading. The declination is resorted by using the webpage: http://magnetic-declination.com/, by using a online calculated the angular degrees and arc minutes are resolved to radians.

Heading can be calculated by:

$$Heading = 2 \times \arctan(\frac{\sqrt{x^2 + y^2} - x}{y}) = atan2(y, x) \tag{35}$$



$$headingDegrees = Heading \times \frac{180}{\pi} \tag{36}$$

10 Processing GUI and Terminal communication

10.1 Attitude Sensor Fusion

With an accelerometer and a Gyroscope you may create an attitude sensor fusion based on these chips. A lot of stuff can be found by searching IMU and MARG sensors. The implementation of this combination is seen in the prossesing exaple code for IMU. The implementation on this page has two processing program, one on chip version where all the math is done on-chip and one where math is done in processing. The firs version is prefferable as we want to read the raw parameter of the board and do the sensor fusion on the computer.

GUI is made in Processing IDE

The serial has to be match to the device, check Tools-¿Serial Port to get how to work this.

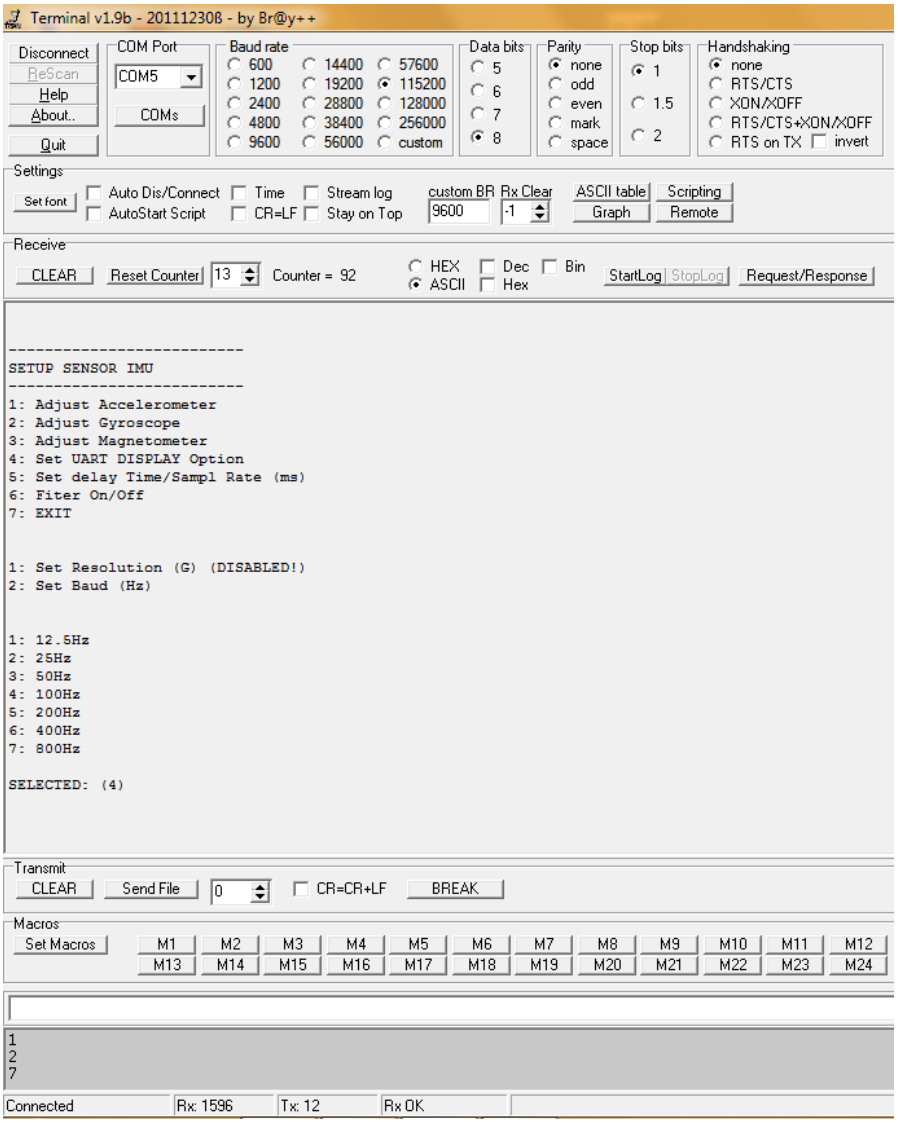
10.2 Terminal communication

Information received from the sensors by the microcontroller can be transfer to a Computer Terminal through the FT232 hardware. The output on the UART is then converted to a serial output that the computer can interpret. Terminal is a great tool for debugging software, as the sensor reading can be displayed on the computer screen. Here, the Brays terminal is used. Additional information can be found in[**?**].

A great feature is using the terminal for debugging, but now there will be added functionality to configure the datalogger through terminal. A menu system is created for the datalogger, and is made available when pressing the SETUP button when running the datalogger in IDLE mode as described in chapter XXX. When entering SETUP the following screen will be presented.

By sending numbers through over the TX line from the terminal, the device can be configured to the users preference.

Now, the device can easily be reconfigured to suit the need for the user or during testing. It is not necessary to connect the mainboard to a JTAG and flash the controller to change setup, instead changes can be done quick and effective through a laptop and terminal-window.



The terminal program is a switch-state logic using the *scanf()* function to receive commandos from the computer and change the desired values. The cs_Terminal.c file uses the methods from cs_Sensors.h and combine these with the values predefined in one of the REGISTER entries files for the sensor boards.

10.3 Processing

Directly taken from the Prossesing.org[41] web page, it states that: "Processing is an open source programming language and environment for people who want to create images, animations, and interactions". It was intended to serve as a software sketchbook,

_ >		
1: Adjust Accelerometer		
	1: Set Resolution(G)	
		1:2G, 2:4G, 3:8G
	2:Set Baud (Hz)	
		1:15.5Hz, 2:25Hz, 3:50Hz, 4:100Hz, 5:200Hz, 6:400Hz, 7:800Hz
2: Adjust Gyroscope		
	1: Set RANGE (deg/s)	
		1:250deg/s, 2:500deg/s, 3:100deg/s, 4:2000deg/s
	2: Set Low_Pass_Filter (Hz)	
		1:256Hz, 2:188Hz, 3:98Hz, 4:42Hz, 5:20Hz, 6:10Hz, 7:5Hz
3: Adjust Magnetometer		
	1: Set Averaged Samples	
		1:1-Sample, 2:2-Samples, 3:4-Samples 4:8-Samples
	2: Set Baud (Hz)	
		1:0.75Hz, 2:1,5Hz, 3:3Hz, 4:7.5Hz, 5:15Hz, 6:30Hz, 7:75Hz
	3: Set Gain (Gauss)	
		1:1370/0.88(LSB/Ga), 2:1090/1.3(LSB/Ga), 3:820/1.9(LSB/Ga), 4:660/2.5(LSB/Ga), 5:440/4.0(LSB/Ga), 6:390/4.7(LSB/Ga), 7:330/5.66(LSB/Ga), 8:230/8.1(LSB/Ga)
4: Set UART DISPLAY option		
	1:Accelerometer data, 2:Gyroscope data, 3:Magnetometer data, 4:X,Y,Z-axis ACC+GYRO, 5:X, Y, Z-axis comma separated, 6:X,Y,Z Legend separated Data, 7:PROCESSING GUI DataFormat, 8:G-force Out (Terminal)	
5: Set Samplings Rate	Set time milliseconds(ms)	
6:Filter ON/OFF	1:ON, 2:OFF	
7:EXIT		RETURN TO IDLE

Table 7: MENY system in Terminal-mode

but has evolved into a tool for generating finished professional work. It is free for download and open source. The processing has a huge pile of libraries to support sound, video, computer vision and more.

Cargosafe found it interesting to develop a tool both for demonstration purposes and visualisation. Processing software12 is here used to receive real-time data from the datalogger and manipulate them in the software. A sketch is written, opening a communication channel to the device and interpret the data over the FT-232 connection. The data are received as comma-separated stream of data ended with the ($\backslash n$) new_line commando.

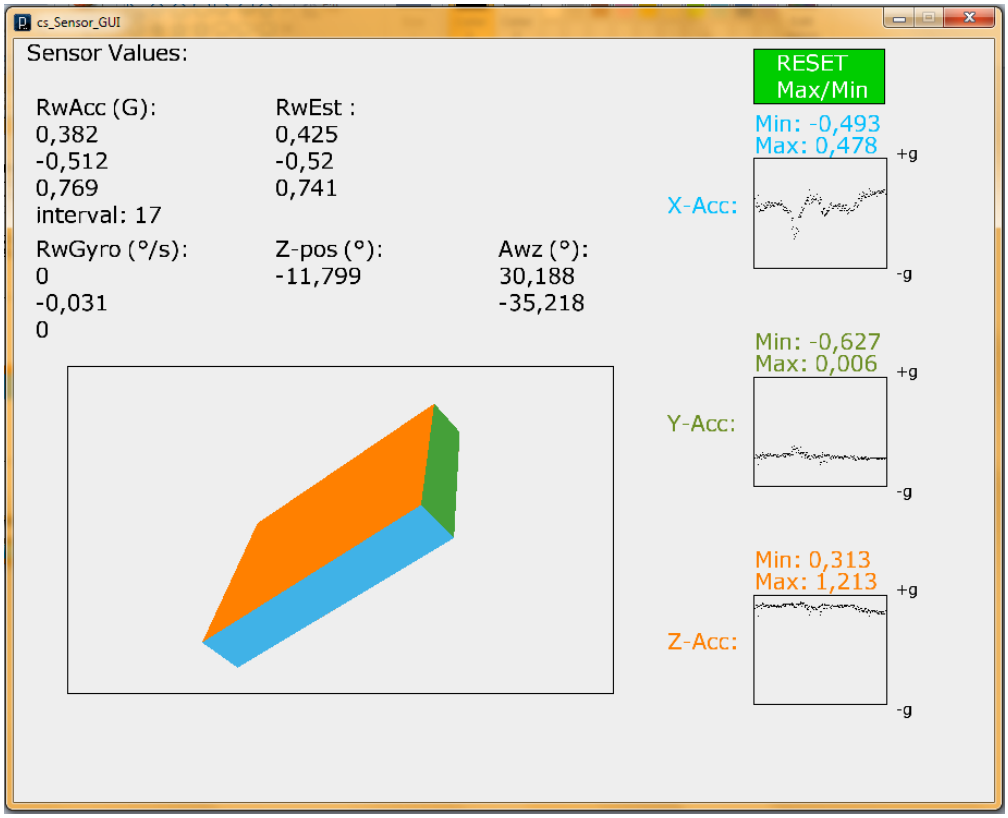


Figure 12: Processing GUI

All the data entering the processing-sketch are RAW values from the sensors. To reduce the amount of calculations and load on the microcontroller, all of the computation is done in the designed program. An attempt is made to do a sensor fusion between the accelerometer and the gyroscope. For formula deductions and implementation are based on[42] and 10

```
currentTime = millis();
interval = currentTime - lastTime;
lastTime = currentTime;

for(w=0;w<=1;w++){
    tmpf = Gyro[w];                          //get current gyro rate in deg/s
    tmpf *= interval / 1000.0f;                       //get angle change in deg

    Awz[w] = atan2(RwEst[w],RwEst[2]) * (180 / PI);   //get angle and convert to degrees
    Awz[w] += tmpf;             //get updated angle according to gyro movement
}
//estimate sign of RzGyro by looking in what qudrant the angle Axz is,
//RzGyro is pozitive if  Axz in range -90 ..90 => cos(Awz) >= 0
signRzGyro = ( cos(Awz[0] * PI / 180) >=0 ) ? 1 : -1;

//reverse calculation of RwGyro from Awz angles
for(w=0;w<=1;w++){
    RwGyro[0] = sin(Awz[0] * PI / 180);
    RwGyro[0] /= sqrt( 1 + squared(cos(Awz[0] * PI / 180))
                  * squared(tan(Awz[1] * PI / 180)) );
    RwGyro[1] = sin(Awz[1] * PI / 180);
    RwGyro[1] /= sqrt( 1 + squared(cos(Awz[1] * PI / 180)) * squared(tan(Awz[0] * PI / 180)) );
}
RwGyro[2] = signRzGyro * sqrt(1 - squared(RwGyro[0]) - squared(RwGyro[1]));
}
//combine Accelerometer and gyro readings
for(w=0;w<=2;w++) RwEst[w] = ((RwAcc[w] + wGyro * RwGyro[w]) / (1 + wGyro)*biasEST);
```

11 Tests and Results

Testing the datalogger is important to ensure the quality and performance of the device. Both configuration using the Sparkfun Sensor board and the Atmel Sensor xPlained1284P are used during the test-phase. At the time being the hardware is proven stable after a scrupulous selection of components and design. A presentation of data and methods regarding accuracy and stability of each sensor from lab test could be displayed, but procedure used for sensitivity mismatch, offset error calibration and noise performance are well described in [40]. Similar procedures can be applied to gyroscope or the magnetometer for improving performance. The magnetometer possess a self-test feature that is useful ensure correct operation of the device. Gyroscope should be positioned on a rate-table, and controlled for correct operation and scaled for mismatch, offset and misalignment error. However, a accurate rate-table has not been available, so the sensitivity calibration can not be done and of this reason set aside in this thesis. Only offset error is accounted for, and scaling of the gyroscope output for correct value.

Some key-data regarding sensor performance will be presented for comparison, but the reader is advised to read [40] regarding performance testing and results. Test results from the Sparkfun 6DOF device will not be displayed in this section. The reason for this is that during last-minute testing of performance, the sensor is short-circuited by a mistake, making it useless. As the unit has to be shipped from USA, a decision is made only to test the datalogger with the ATAVRSBIN2 board.

All data collected during these tests are post-processed in Matlab, as calculation made on RAW values by the microcontroller would produce the same result as manipulating RAW data off-line on a computer.

11.1 Test setup

For checking for correct operation and during calibration, only the xPlained1284P secured to either the Sparkfun Sensorboard or Atmel Sensor Xplained are used. It is not before field testing the power board and communication board are fixed to the unit. The main reason for this is that these PCBs were not yet produced, tested and verified while creating the software. It is also easier to work only have the xPlained1284P board connected to sensorboard in addition to JTAG and power supplied from a computers USB port.

The power to the system is 5V, either supplied from battery or USB power from computer. UART is experimented with, and mainly set to 38,200 baud. But during field testing the baud is set to 115200, which is the highest baud-rate the OpenLOG can

handle. I2C is the only used communication bus between the controller and sensors. Buttons placed on the xPlained1284P board are used as described in 9.3 to switch between the states. Brays terminal, Stamp Plot, Processing GUI and OpenLOG are used for reading data from the device. All the software is initialized according to chapter 9. The data formates are change during testing to suit the software used to interpret the information.

11.2 Noise Performance

In the 5 chapter is is stated that the KXTF9 seemed to have issues regarding noise. A better picture of this is presented in the following plot13,

The KXTF9 is a seriously noisy device (at least mine is). Heavy filtering is required. It needs some output offset trimming (about 8%). It seems to be fairly usable.

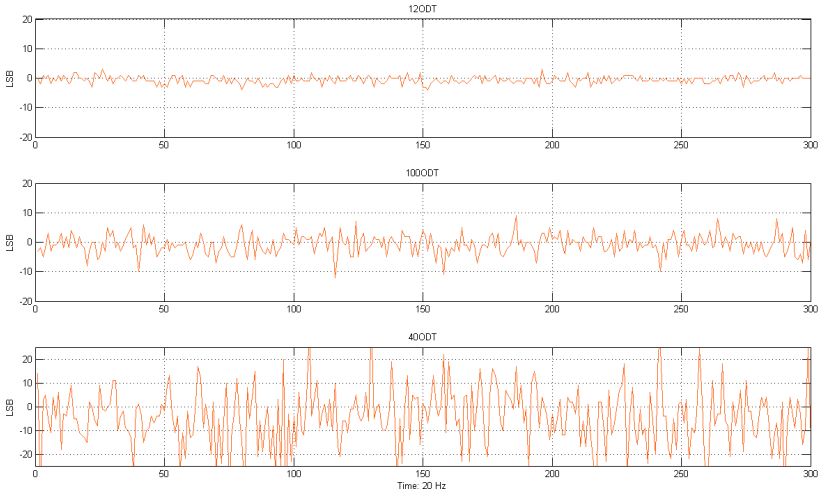


Figure 13: Noise performance of the KXTF9 Accelerometer

By lowering filter frequency of the device, the amount of noise becomes considerably better. But this is destroying the sampling speed. Since one of the requirement is to be capable to have a 20-50Hz samplings rate in the system, it is not possible to use the filter producing less noise. Multiple readout of the same value occur when using a low bandwidth.

11.2.1 Glitch on OpenLOG

OpenLOG is a very useful device for storing data to a memory-card, however there are a few drawbacks. In the OpenLOG specification, it is reported that logs contain dropped

characters at baud rates above 9600bps. As explained in the units documentation, at 9600bps results in 1.04ms per byte. OpenLOG uses a 512byte receive buffer, so it can buffer about 500ms of characters. We have that $1000ms/960byte = 1.04msperbyte$, in other words it will take 532 ms before the buffer is overrun when writing continuous. Since the unit can buffer 500ms, this will never happend and it is therefore the OpsnLOG successfully receive all characters at a baud of 9600bps. This result becomes clear during performance testing of the device storing data from all of the sensors. Trying to store the data at 50Hz show to be error prone. This result can be seen in AppendixA 30.

A test is putted through using a samplings rate of 50Hz reading accelerometer, gyroscope and magnetometer. The data are comma separated, and a counter is incremented every execution to control the difference between the real length of the dataset and the length of the received dataset. Data sampling is preformed over a 5 minute period of time, using a stop-watch on a HTC-desireS timing. The stop-watch and unit are started at exact same moment. A fault tolerance of $\pm 2sec$ is accepted.

Briefly locking at the log-file, it becomes immediately clear that data-packages are lost. First of all, the length of the file is 14921 lines, but should have been 15138. A short conversion of data-points to time gives:

$$Totaltime = (\frac{15138}{50Hz})/60[sek] = 5.046[min]$$

$$0.46[min] \times 60[sek] = 2.76[sek]$$

This gives the total time of 5 minutes and 2.8 seconds which is close to what expected.

Difference between the expected time and the output data time is $15138 - 14921 = 217$ or 3.61 seconds. In other words, 3.61 seconds of data is lost during the 5 minute test or about 1.205% of the information. The overflow occur at random intervals making it hard to create a mechanism to prevent this buffer issue.

The first 5000 values in the log-file are checked for invalid data. Position of where data are contaminated, and the difference between each event is checked. This resulted in 20 event where data is lost. By taking the length between each loss averaged on the 20 events the mean value becomes 4.99sek between each invalid sets.

So every 5 second the system is subjected to information loss. Even if there is a 20ms pause between each data burst, the OpenLOG hardware can not manage size of information.

A new test is performed. Counting is still preformed at each read execution and the frequency is reduced to 20 Hz. From the log-file no information loss or skewing of time can be found during a 5 minutes test at 20Hz producing 6102 lines. In other word no package loss were found when using a 20Hz sample rate.

11.3 Magnetometer performance

Two small test are putted through to check the magnetometer sensor performance. The first is to control that the setup and algorithm used to derive compass heading is correct, the second one is checking how serious the impact of metallic objects are to the magnetometer sensor.

11.3.1 Compass heading

To test that the compass output the correct heading, a simple test is performed.

ExecutionFrom the plot down below the compass is rotated three times clockwise direction before it is rotated one time anti-clockwise. The purpose of this test is to check that the compass is working correctly through its entire range, and as shown the heading varies from 0° to 360°. To know this information is essential during field-tests.

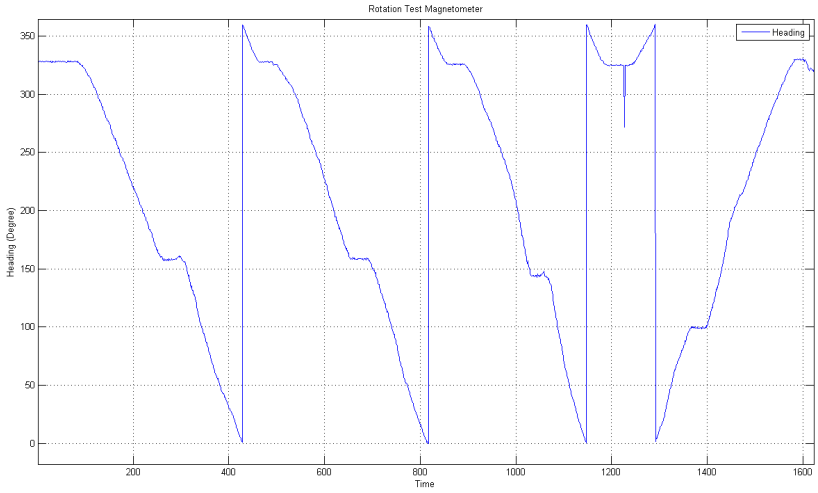


Figure 14: Rotating Compass Sensor

Result The plot shows that turning the unit lying flat on a surface produce magnetic heading from 0-360 °. The small bumps on the graph is due to the sensor is rotated by hand, and it is not possible to keep a steady motion as the hand must be shifted when rotating. The spike around time=1300, is caused by unintended shaking. Nevertheless, it seems that compass heading is working correctly.

The post-processing Matlab script used for correcting the compass is created as follows:

```
1 %MATLAB SCRIPT
2 %Resolution from datasheet
```

```
Scale = 0.92;
x_mag_scale = x_mag*Scale;
y_mag_scale = y_mag*Scale;
heading = atan2(y_mag_scale,x_mag_scale);

%Declination calculated using Magnetic Declination information
%and resolve this to radians. Set it to milli-radians
declinationangle = 34.62/1000;
heading = (heading+declinationangle);
%Correct for reversed signs
for R =1:time
    if(heading(R)<0)
       heading(R) = heading(R)+(2*pi);
    end

    if(heading(R) > 2*pi)
        heading(R) = heading(R)+(2*pi);
    end
end
%Convert radians to degrees
headingDegree = heading * (180/pi);
headingDegree = smooth(headingDegree,10,'moving');
```

A 10-step moving average filter is used to smooth the compass-values. The only problem using the formulas presented in10.14, is that it does not account for tilt. So the more the compass is tilted the larger the error will become. A 3-axis magnetometer is usually capable of sensing in three directions. But that does not mean that it is tilt compensated, so one really do not know its orientation when the reading comes in. An accelerometer is needed to sense the orientation and combine this to the magnetometer reading. It is necessary to formulate a tilt-compensation algorithm. Adding accelerometer inclination reading one can utilize this to compensate compass tilting. An example for implementation can be found at[44]. But a drawback of this algorithm is that the data will only be valid up to an angle of 40°. More complicated algorithms must be used to get a better result.

11.3.2 Metallic interference

Compasses are sensitive to interference from magnetic materials in close proximity. This is shown in the plot 15. The HMC5883L compass is exposed to a metal.

ExecutionHere, a metal rod is kept at a distance and slowly moved close to the sensor, until it lies on top of the sensor. The metal rod is then moved away from the sensor. The movement is done from a 40cm distance to placing the rod on top of the sensor. This is repeated two times using a even slower motion on the second attempt. The sensor is kept complete still, lying on a wooden surface. Not at any time is the sensor moved, and all the data is sampled through cable and terminal to avoid antecedently moving the device out of its position.

Result From the result it is clearly seen that the metallic rod has a great impact on the sensor bringing it out of bounds and into saturation far away from its range. All

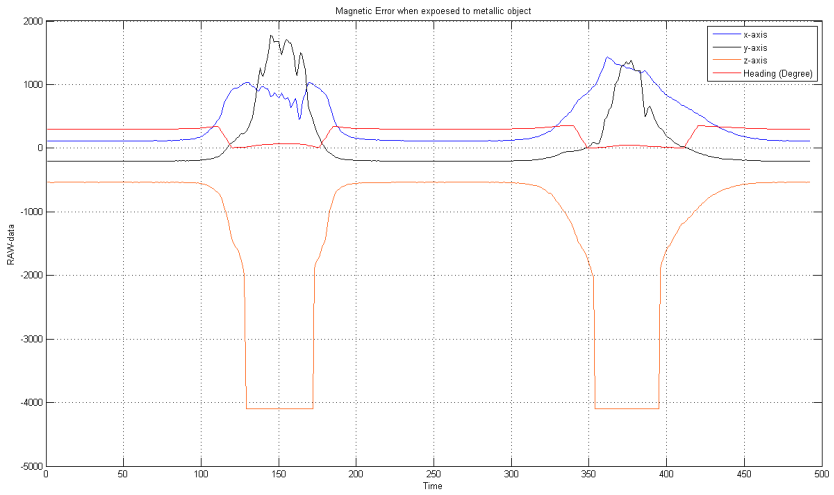


Figure 15: Compass Error when Exposed to Metal

the X-, Y-, and Z-axis are affected by the interference from the rod. Starting at 301° the heading slightly increases to 346 at time 110 before dropping to 7°, when removing the rod from the sensor one can see the opposite effect and the heading return back to it correct value when the rod is not inflicting the reading.

11.4 Practical Test 1 - Triangle Course

The first test on trailer, a open tarmac-covered parking lot is used for the test area avoiding obstacles and traffic. This would give a test where repetition can be made, on same condition without loosing speed or stop for traffic. The idea is to create the same test-scenario multiple times to control if any periodic information occur. Overview of the test area is taken from google map.

Execution The course displayed above is repeated three times, attempting to maintain as constant speed as possible following the same path. The speed is kept somewhere between 20-30 km/h. Datalogger is placed on a trailer as shown of the photo. The datalogger is set according to default values presented in 9, and with a samplings frequency of 40Hz. Even if there are risks of losing packages at this rate, 40Hz is set on request from Cargosafe. The reason is that the amount of package-loss is considerably lower compared to 50Hz sampling rate.

Direction of the course is anti-clockwise, like the above picture indicates. North is pointing directly upward. Logging with constant speed to the OpenLOG hardware the following result were produced.

Result As mentioned above, the frequency is set to 40Hz risking loosing data. Four

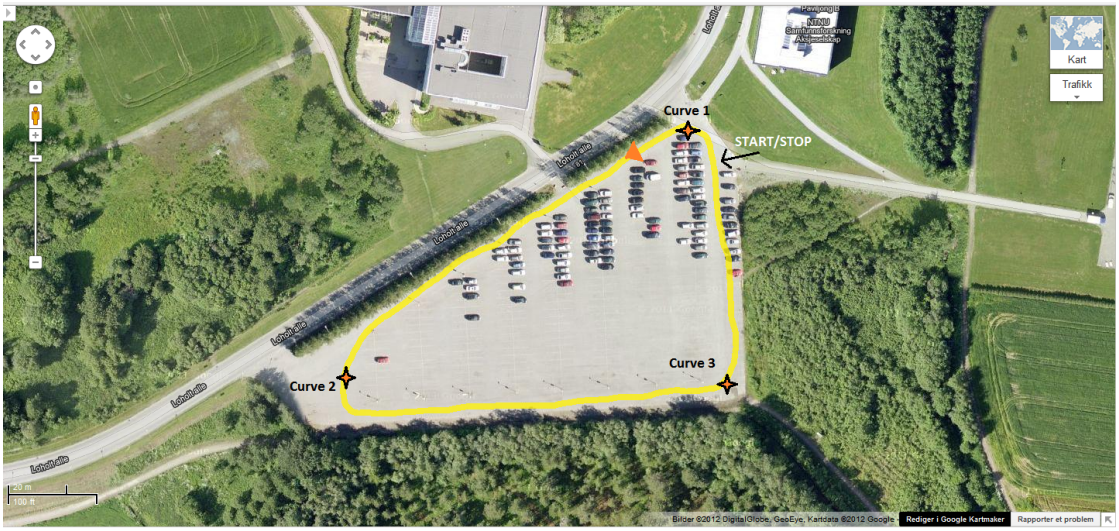


Figure 16: Test Course 1

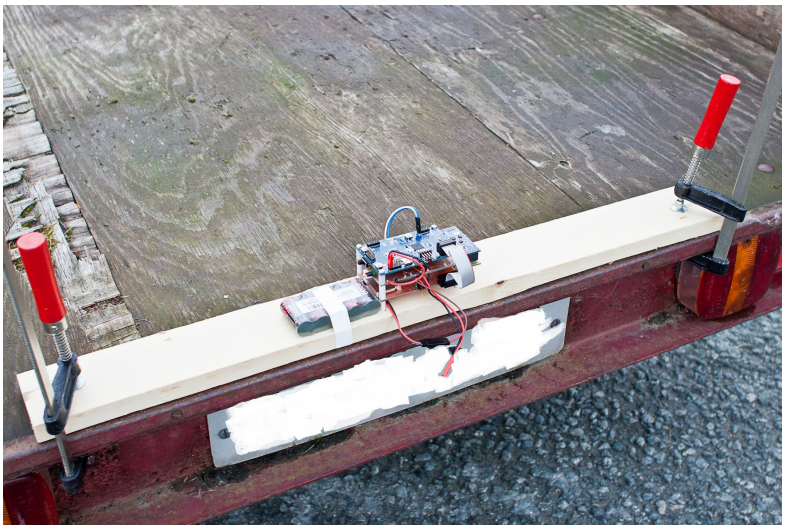


Figure 17: Datalogger on Trailer

packages are discovered lost. On average this becomes one package loss every 50.42 second8. This give a total loss of 0.05percent of the information which in this case is nothing.

Post-processing data, following result can be displayed. Data have to be filtered to some extent to get useful information from the dataset.

The roll angle in 19 is based upon the X-axis pointing directly to right on the trailer and decomposition of Z. Here the roll experienced by the trailer is revealed in every turn.

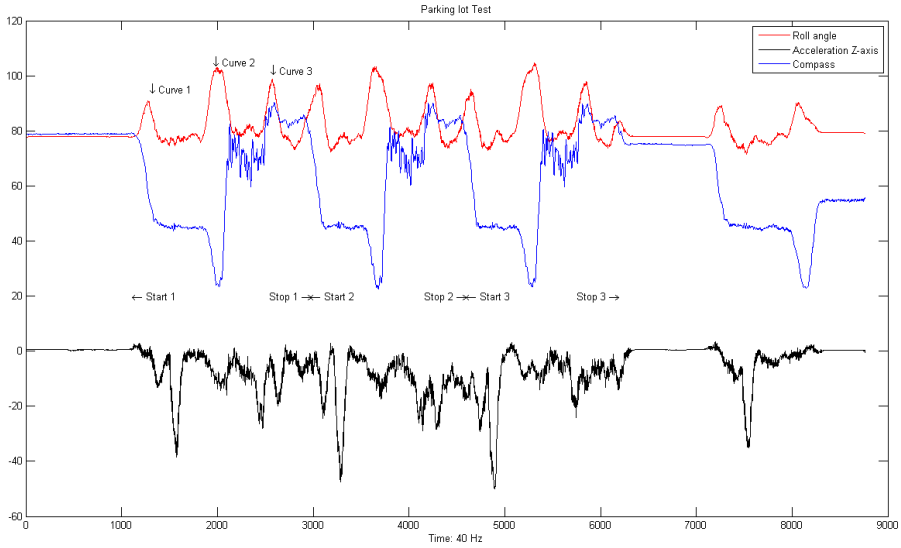


Figure 18: Test Course 1

Line Number	Difference
3250	3250
4067	817
7797	3730
8113	316

Table 8: Loss of data

The magnitude is in the same direction every single turn, which is logic as the vehicle is driving in the same direction around the course. From the compass direction we can see that the vehicle is turning in circular motions. The effect of not compensate for tilt becomes clearly between curve 2 and curve 3. The tarmac in the parking lot was very rough, especially between curve 2 and curve 3. The Z-axis is included to give an indication of this. A bump in the road is indicated periodically on the z-axis. For the accelerometer, the output is probably too sensitive, to produce useful interpretation of the roll angle a moving average filter of size 40 is used.

Comparing the plot to google map picture gives a better view of what is happening. We can see that after cornering curve 1, the compass heading is constant as expected from the distance between curve 1 and 2, cornering curve 2 we get into the roughest part of the tarmac, resulting in ocsillations then cornering curve 3 and the pattern repeats on the two next rounds.

The plot after STOP is just a result of repositioning before turning off the datalogger.

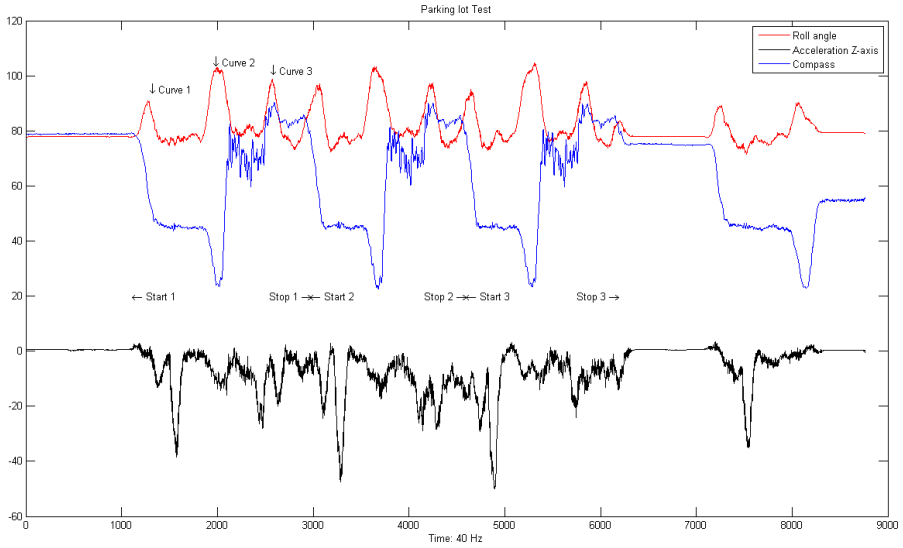


Figure 19: Test-1 Plot

11.5 Practical Test 2 - Slalom

A test controlling the response of slalom driving is put through. Settings are like before, the sensor tightly secured at the back of a trailer as described above. The settings are the same, and still using a samplings frequency of 40 Hz.

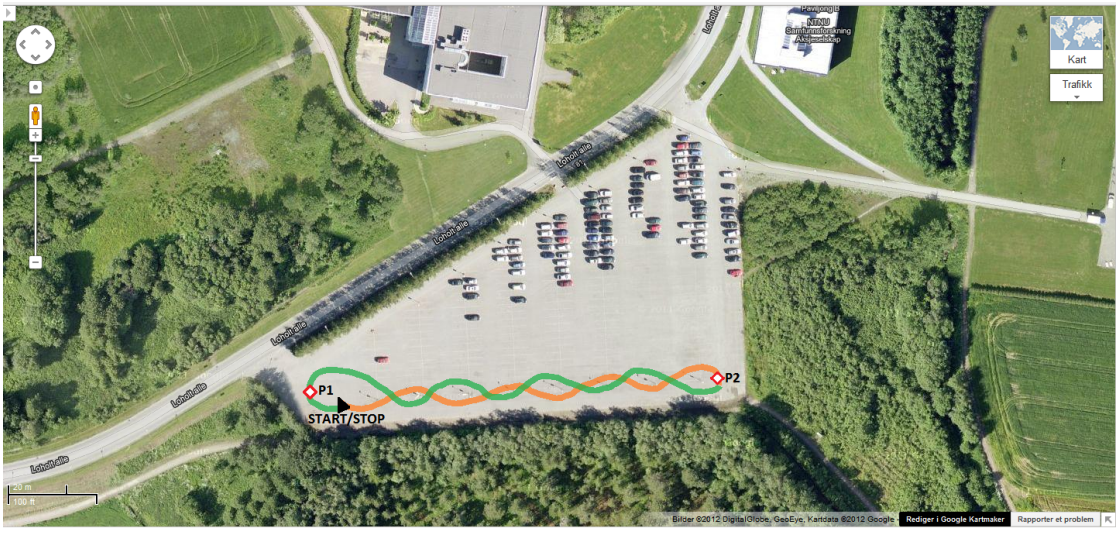


Figure 20: Test Course 2

Exection The course is simply driving slalom between light poles at a parking lot. The measurment start from rest, where the vehicle is accelerated to about 25km/h after doing the course twice the vehicle is stopped at the startingpoint. Utilizing the fixed length between each pole, we hope to receive data with fixed oscillation and repetitive pattern. The gyro should indicate angular rotation around z-axis. Accelerometer should indicate a strong acceleration along x-axis and smaller along the y-axis, since throwing the trailer from side to side should induce more acceleration across the trailer and less acceleration in vehicle direction because of constant speed. When accelerating or braking, we should see a spike on the Y-axis.

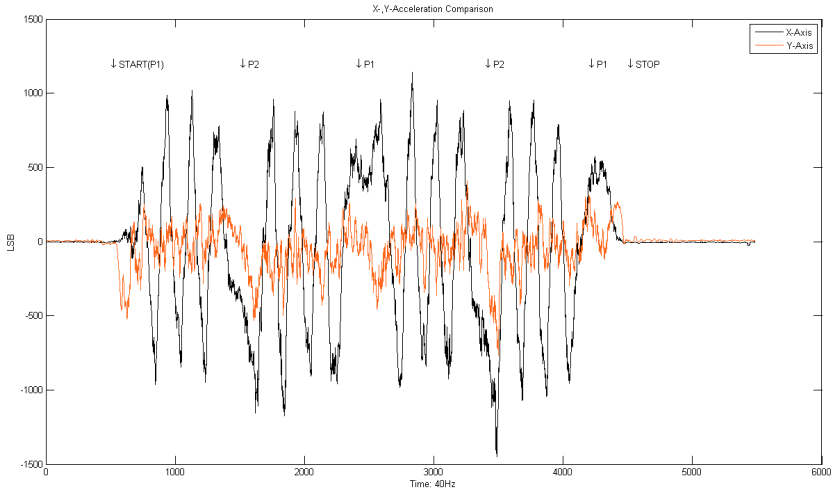


Figure 21: X-, Y- Acceleration

Result As expected the magnitude in X- direction from the plot21 is greater than the Y-direction. When starting we can see a acceleration along the Y-direction, as a result of increase speed when starting at zero velocity. The repetitive pattern of driving slalom is quite clear. Another thing to notice is that the acceleration is opposite when turning around P1 or P2. This gives the nice result that it is possible to differ between anti-clock and clockwise rotation in curves.

Here22, a comparison of the X-axis of the accelerometer and Z-axis Gyroscope is plotted. Once again the effect from driving side-to-side is similar for both the accelerometer and gyroscope. We see that when turning left the gyroscope rate increase due to change in angular velocity of the trailer. And at a right turn indicated by the accelerometer, the gyro rate shifts and starts to decrease, indication a right turn. Again, turning around point P1 and P2 the rate period increase as we are rotation through a whole circular rotation.

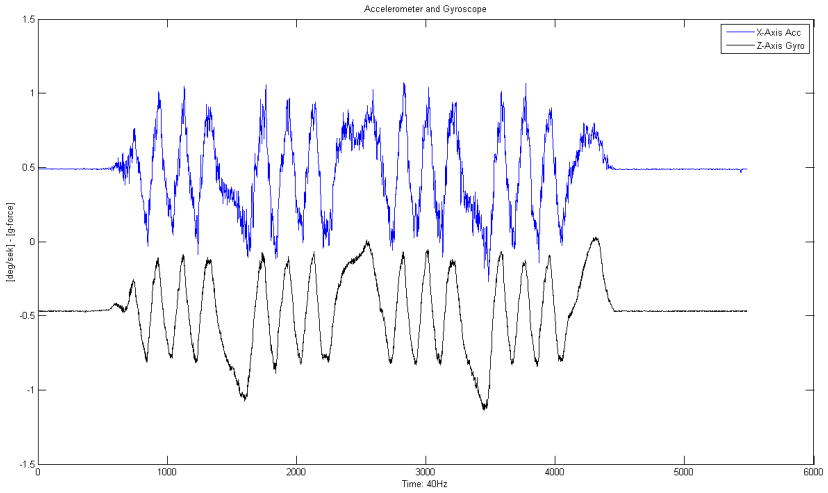


Figure 22: Z-Gyro vs X-Accelerometer

11.6 Practical Test 3 - Roundabout

For testing of heading, the compass can be used to determine this. The compass prove to be correct in a lab environment, therefore it will be interesting how the compass will preform in a practical test. This next test is preformed close to a roundabout. See bellow23.

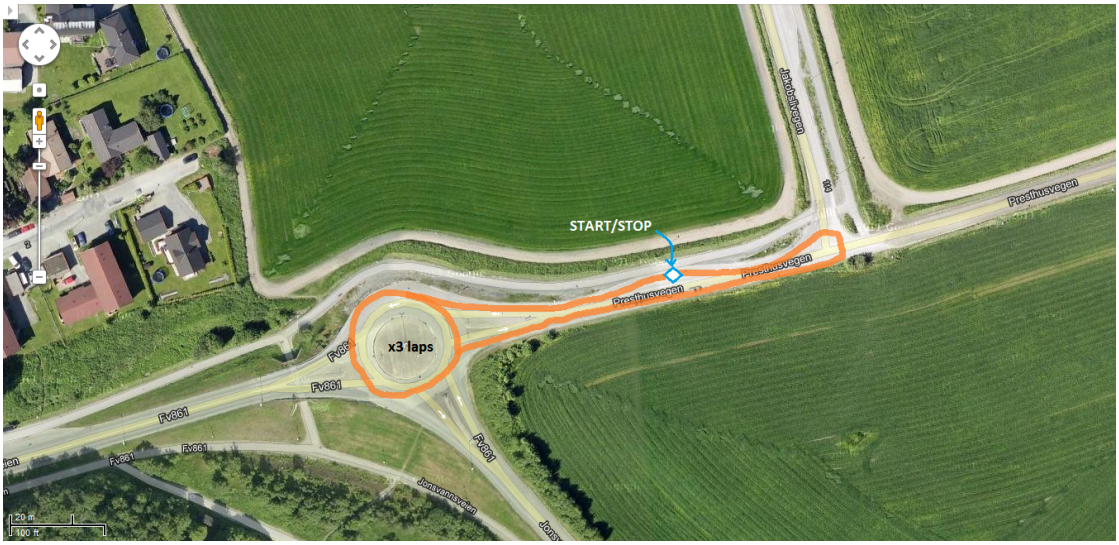


Figure 23: Test Course 3 - Roundabout

Execution Using a bus stop as a starting point, the car started from rest and is ac-

celerated towards the roundabout. The velocity is kept constant about 25km/h while driving 3 complete revolutions before exiting the roundabout. A single U-turn is made before stopping at the initial point.

Result From the magnetometer it can be clearly seen the result of driving in circles. Another result to notice, is the noise generated during the u-turn. Since there are no tilt compensation on the compass, the heading is affected by inclination caused by trailer moving side to side on the axles. The more it shake, the worse the error becomes and compass reading is corrupted.

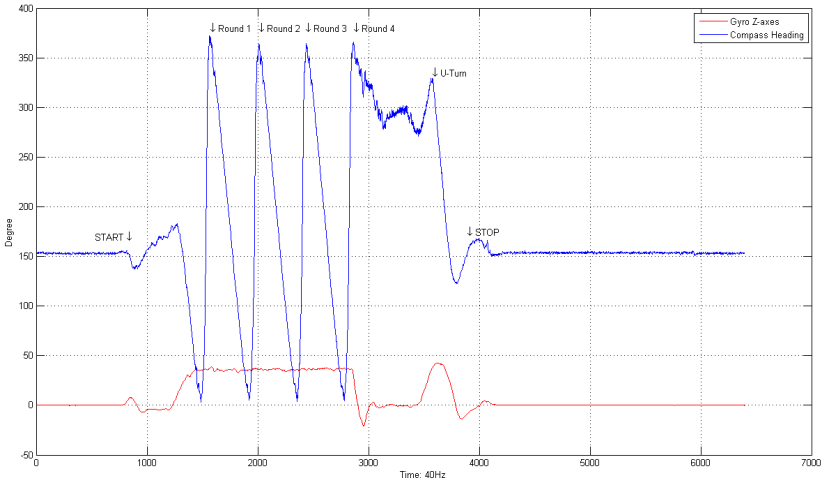


Figure 24: Test Course 3 - Roundabout

The measured times compared and positions in plot are presented in the table 9. Each round is counted when completing a revolution at the entrance.

Event	Time Measured	Position
START	20 [sek]	800
Round 1	29 [sek]	1160
Round 2	39 [sek]	1560
Round 3	50 [sek]	2000
U-turn	60 [sek]	2400
STOP	93 [sek]	3720

Table 9: Resort frequency

From the table above, we see that there is a good concurrence between the measured time of an event and the dataloggers time of an event.

During the entire roundabout, the gyro rate is kept constant. As the rotation speed is

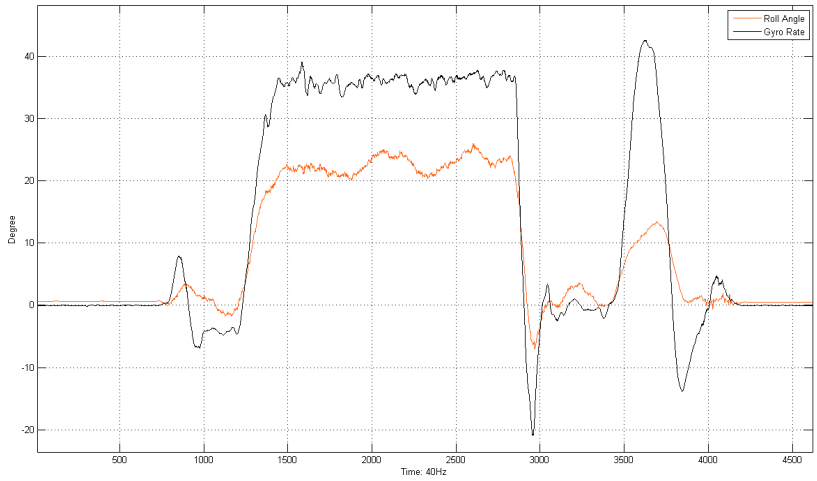


Figure 25: Test Course 3 - Roundabout

kept constant during all of the three rounds, this adds up to that the angular velocity is constant. The tilt angle of the trailer is fixed, and it seem logical assume that the force through the circular path should be constant as the velocity and radius is closed to fixed. Again, the oscillation from exiting the roundabout can be resolved, as for the U-Turn in the end where there is a large magnitude in the gyro measurement.

11.7 Practical Test 4 - Regular Road

Now, a ordinary test is putted through driving a short course with traffic. The vehicle is accelerating from rest, to a driving speed of approximately 60km/h. This test is preformed 3 times to check if data can be reproduced over a distance using the same location, applying the same driving conditions and keep as close as steady speed during the test.

Figure 26: Test Course 4 - Map of the course

11.7.1 Concurrency of time

Result from measured time and logging frequency from datalogger are compared. While driving this course, the time is captured using a separate stopwatch. This is done to simplify post-processing to get a rough estimate for where the data can be found at each way-point located on the map. A simple calculation is done to control that the measured time is according to the time of the samples. As earlier mentioned, the start/stop event can be located from accelerometer reading, by utilizing this property the precise moment when the vehicle is brought to a stop can be resolved. The following calculations are used for determine theoretical clock

$$(\frac{steps}{sampling frequency})/60[sec] = time \tag{37}$$

rearranged, this gives

$$out_put frequency = \frac{steps}{time \times 60} \tag{38}$$

The table below includes the values from log file, stopwatch and calculated time10.

	Number of sample	Estimated time	Measured time
Test Run 1	7947	3:18	3:17
Test Run 2	8157	3:23,9	3:24
Test Run 3	8174	3:24,4	3:24

Table 10: Resort frequency

From the above table10 it is clearly that the true frequency is 40Hz. Non noticeable skew in clock makes this a fairly precise and reliable system. Even the risk of package loss, it may seem that this does not affect the clock-performance.

11.7.2 Brief presentation of Events

A short view of some events are presented here. These are some example of data that can be interpret from the log-files from the datalogger.

First, presenting the gyro data output from Z-axis for all of the three attempts27, reveal that the data can be reproduced when maintaining the same conditions. The magnitude and time is close to even for all of the three attempts.

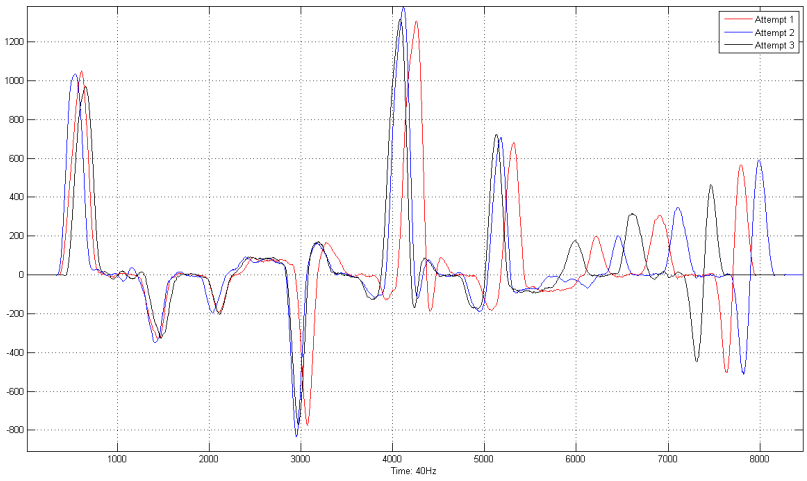


Figure 27: Test Course 4 - Regular driving

From the accelerometer in X direction, it is easy to see the result of the 90° turn at P326. This graph shows the result from the hard bend to turning at the turning point and yet again the impact from the hard bend heading back to start28.

During the second test-run a car came from the side-way marked as a star on the map26. The test vehicle had to be brought to a stop, because of duty to yield. This happens

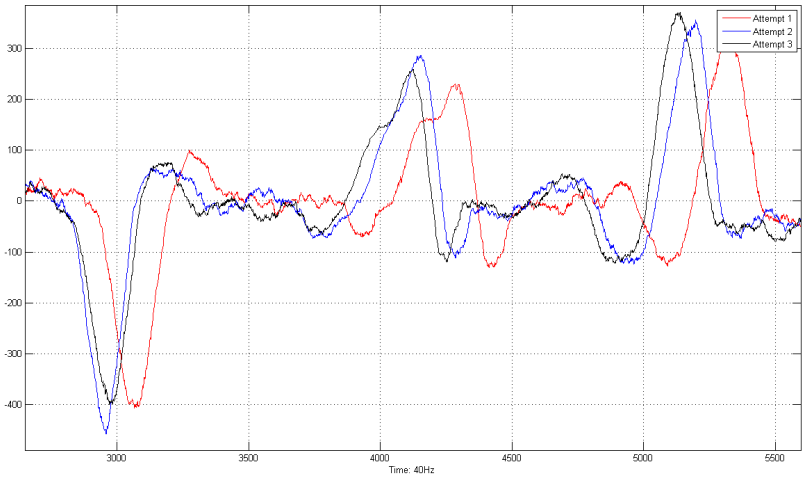


Figure 28: Test Course 4 - Regular driving

at 2minutes and 22seconds, or around sample 5680. From the plot29, we see a large deceleration on the Y-axis, indicating the brake event.

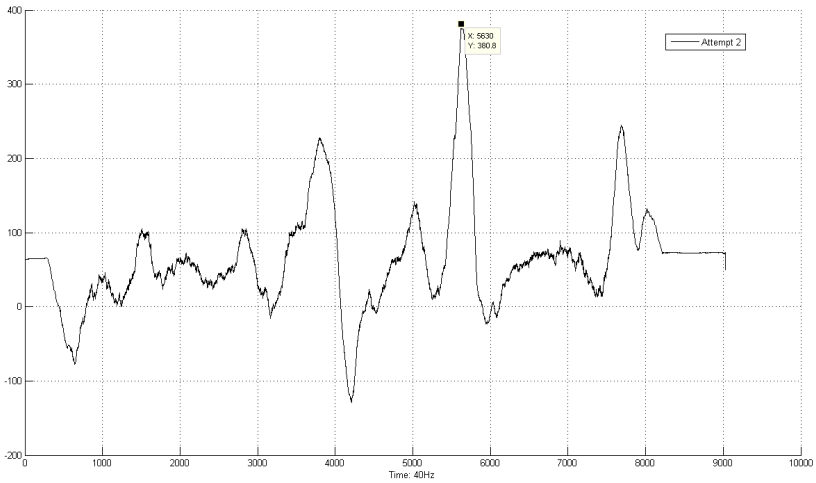


Figure 29: Test Course 4 - Regular driving

12 Discussion

The selection of off-the-shelf ready-made components have been successfully put together. All the hardware can communicate and interact without errors. There have been challenges due to the board controller on the xPlained1284P card. Workarounds have been necessary to overcome this, and alternative solutions have fixed the issues.

Specialization of hardware components intended for the data-logger system have proven useful, yet fragile. During testing, accidental frying of some of the components could easily been avoided by implementing short-circuit protection and isolate the copper layers on the boards. By separating the system into several hardware parts, testing and redesigning is made easy. The communication board can be utilized to develop software and testing for functionality without being connected to the other hardware. If an accident would occur with the power board, it can be swapped with a new one without the risk of rebuilding the entire circuit, and avoiding the cost of ordering expensive components used on the other boards.

However, a smaller design would be preferable. Even so, the device have been easy to place or secure to surface by screws. Having the capability for battery driven and local storing of sensor data have made it quite flexible.

Battery-operated devices are hard to design - The powe rconsumption is hard to keep to a minimum. And outermost care needed to be done to select the appropriate components giving good specification, correct operation and low consumption. Selecting battery technology add complexity as the handling of the types of battery demands different care to be taken.

Research and development takes a lot of time, usually at least 3x as much as planned. Designing, reconfiguration, testing, production and planning demand a huge effort.

From data log there were no indications of Wild point (spikes) or High variance (fluctating) signals. It seems that the sensors operate well with no frozen signals or signs of High derivative(sudden jump in flank).

13 Conclusion

This thesis have given a in depth knowledge and experience with electronics, MEMS sensors and problems related to orientation sensing and noise suppressing and realization of the package for a stand-alone data logger. This thesis has resulted in better understanding of electronics and printed circuit boards design but, as proven by the projects created, the knowledge gained on related topics from the master degree is quite relevant.

Practical implementation issues using ready-made breakout board and education kit are now well known and a better understanding off issues are made clear. Currently, I'm also intimate with MEMS sensors and can implement practical applications utilizing their functionality.

The main objective of this tesis has been to develop a datalogger tool with either available or own-designed hardware. This work covers a complete development process, from system requirements, to design/redesign, software implementation and testing. It has required a wide range of disciplines, both theoretical and practical regarding embedded design.

In this thesis several proposed solutions are presented and evaluated. One of them proved hard to realize in practice, as the there were issues with the size of component. The sensor platforms are easy to implement and swap using the same hardware interface and power supply. The challenge is to get the software running properly and error free on the same units, and slight modifications are needed when adding new sensors. However, the state-machine, communication and hardware related software are transparent, so introducing a new sensor only require sensor specific initialization and queries.

A Atmel xPlained card is used together with own developed PCB as the base when creating the test platform for the data-logger. It consist of the Atmel xPlained1284p, power board, communication and storage board, and two different IMU's board are used to measure raw data and sense orientation. This makes a fairly compact design.

Drivers and basic software functionality is implemented in C-code to present a software framework that is well known and can be utilized by Cargosafe. The framework can handle communication with external measurement units, let the user manipulate the sensors through a terminal interface. Processing GUI is made for presentation purposes, debugging and can be used for testing motion code on the raw data by having the motion algorithm on the computer instead of the device.

A data-logger and a few hardware solutions are presented, implemented and successfully tested on a real vehicle and trailer. Not all requirements are met, like the loss of packages. But, from testing, this loss is considered non existing compared to the

amount of data that is stored. Further investigations regarding sensor issues, due to magnetic disturbance and calibration must be preformed. As for now, this unit serves as a stand-alone data-collecting tool for motion sensors.

14 Future Work

This thesis open the way to many possible future developments mainly related to improvement and expanding the orientation sensing approach and practical and ease of usage of the Data-logger.

14.1 Orientation Sensing

Regarding the orientation sensing and angle calculations presented there are still much work to be done on the topic of enhance algorithm for motion sensing and calibration of the sensor. To keep workload low and not use too much tiime on accuracy of the sensors, a 6-way calibration technique is used when calibration the sensor. There exist a wast wariety like using advanced algorithms and maby a lazy approach by using the self test of the device.

14.2 Datalogger

Datalogger has proven to be a good tool for collection raw data measurement. The further available I/O makes this device capable of sourcing and expanding the units connected to the device, and can be used to store data from a larger amount of sensors.

14.3 Hardware

In future development it would be of interest in reducing the size of the unit, reducing both the space the data-logger holds, but also selecting components so the device can be incorporated easier into products and utilize the capacity. It is preferred to offer charger, battery maintenance, I/O and processor in parallel with memory and storage on one compact PCB. Before do so a better design making calculations to ensure components are within the required specification must be done.

14.4 Software

Following improvement of software to enhance the user experience should be done. All the set-up values should be recorded to the EEPROM, perhaps through a data-struct such that all the selection done in the data-logger through the terminal is stored. The problem is that when cutting power the unit has to be re-initialize with the desire values. This is because the setting are predefined at start-up and the device is set in a pre-defined

setting which is controlled by the programmer. It is frustrating have top connect the device to a laptop to configure the device on each and every power up. When field testing is is preferable to start from the last set-up or have th opportunity to select last configuration or default mode.

14.5 Processing GUI

A lot of features can be added to the processing GUI like battery-status, power consumption, magnetometer reading, gyro reading, more graphs, temperature from sensor and xPlained1284P board. Further could scrolling menus be made to control the device and let one configure the data-logger from the GUI. This would be a great way to quickly change parameters and see and confirm the result of the change. Filters could also be implemented giving online possibilities to manipulate the strength and range of filter directly. Offering real-time pre and post result could give the user a great tool for checking the response of the selected parameters.

14.6 DSP and sensor fusion

It would be of interest to do a sensor fusion on the accelerometer, gyroscope and magnetometer. The features of each of them can contribute to eliminate errors.

References

[1] Odd Arild Olsen, *Instrumenterings Teknikk*. TAPIR, NTNU, 1998.

[2] Paul A. TIPLER; Gene MOSCA, *Physics for Scientists and Engineers*. Freeman, 2004.

[3] Application Note, *AVR315: Using the TWI module as I2C master*. ATMEL, 2010.

[4] Application Note, *AVR309: Software Universal Serial Bus (USB)*. ATMEL, 2010.

[5] Application Note, *AVR151: Setup And Use of The SPI*. ATMEL, JULY 2008.

[6] Andrew, *USING SPI on an Avr*. Website, http://www.rocketnumbernine.com/2009/04/26/using-spi-on-an-avr-1/, 2009.

[7] Application Note *Tilt Sensing Using Linear Accelerometers*. Freescale Semiconductor, JUNE 2006

[8] ATMEGA 1284P *8-bit Atmel Microcontroller*. ATMEL, MAY 2011.

[9] Application Note, *Implementing Auto-Zero Calibration Technique for Accelerometers*. Freescale Semiconductors, MARS 2007.

[10] John CATSOULIS, *Designing Embedded Hardware*. O'Reilly, 2'nd edition.

[11] Bjoernar VIK, *Integrated Satellite and Inertial Navigation Systems*. Departmant of Engineering Cyberntetics.

[12] Brian W.Kernighan, Dennis M. Ritchie, *The C Programming Language*. 2'nd Edition.

[13] Article 10048, *USB vs SERIAL*. Knowledge base, http://totalphase.com

[14] *Accelerometers*. http://www.siliconfareast.com/accelerometers.htm.

[15] *Sensor Overview*. http://www.freescale.com/webapp/sps/site/overview.jsp?code=SNSMEMSOVERVIEW.

[16] *TWI Architecture*. http://www.avrbeginners.net/architecture/twi/twi.html.

[17] Dimensionengineering, *Switching regulators*. http://www.dimensionengineering.com/switchingregulators.htm.

[18] Wikipedia.org, *Proper Acceleration*. http://en.wikipedia.org/wiki/Proper_acceleration#cite_note-0.

[19] Diydrones.com, *Diydrones Forum* http://diydrones.com/forum/topics/atmel-inertial-two-atavrsbin2-arduino?id=705844%3ATopic%3A719308&page=1#comments

[20] Data Sheet *TPS 61200*. Texas Instruments, 2012.

[21] Data Sheet *MCP 73811*. Microchips Technology Inc, 2007.

[22] Data Sheet *MAX 17041*. MAXIM, 2010 April.

[23] User Guide *AVR364: Xplained Hardware User's Guide*. ATMEL, MAY 2011.

[24] Datasheet *LM7805*. Texas Instruments, 2004.

[25] Datasheet *LP2985 - ULDO Regulator*. Texas Instruments, 2011.

[26] Datasheet *MCP1252 - Low-Noise Charge Pump*. Microchip, 2002.

[27] Datasheet *MCP1640B - Synchronous Boost Regulator*. Microchip, 2010.

[28] Datasheet *24FC1025 - Serial Electrically Erasable PROM*. Microchip, 2006.

[29] Mechanical Drawing *SD Memory Card Connector Type C*. Multicomp, 2009, April.

[30] Technical note, *Silicone Designs*. `http://www.silicondesigns.com/tech.html`.

[31] Digital sensors, *Silicone Designs*. `http://www.silicondesigns.com/digital.html`.

[32] Digital sensors, *Bosch Sensortec*. `http://bosch-sensortec.com/content/language1/html/4377.htm`.

[33] Datasheet, *ADXL3xx Accelerometer*. Analog Devices, 2007.

[34] Data Sheet, *ITG3200 MEMS Digital Gyroscope*. InvenSense, 2010, Mars.

[35] Datasheet, *KXTF9 3-axis Digital Accelerometer*. Kionix, 2011,Jan.

[36] Datasheet, *IMU3000 Motion Processing Unit*. InvenSense, 2010,August.

[37] Datasheet, *HMC5883L Digital Compass IC*. Honeywell, 2012, April.

[38] Kjell Malvig, *Kompendium del 1 datastyring ttk4125*. Institutt for Teknisk Kybernetikk, 2007, Mars.

[39] Gordon McComb, *Robot Builder's Bonanza*. Tab Electronics, 2'nd Edition 2000

[40] Oyvind Lorgen, *G-force measurement with use of acceleration sensor*. Institute of Cybernetics, NTNU, Fall 2012 http://processing.org

[41] Processing, *Open Source Alternative to Proprietary Software Tools*. `www.processing.org`.

[42] Sensor fusion, *Starlinos*. `http://starlino.com/imu_guide.html`.

[43] Bosch Sensortec, *Digital, triaxial acceleration sensor*. **BMA150**, Data Sheet, MAY 2008.

[44] Tilt Compensating a Compass, *Love Electronics*. `https://www.loveelectronics.co.uk/Tutorials/13/tilt-compensated-compass-arduino-tutorial`.

[45] UM10204, *I2C-bus specification and user manual*. NXP, 2012.

[46] Eagle PCB Dessign Software, *CadSoft*. http://www.cadsoftusa.com/.

[47] Intro. to MEMS Gyroscopes, *Electroiq*. http://www.electroiq.com/articles/stm/2010/11/introduction-to-mems-gyroscopes.html

[48] Brje Forssell, *TTT4150 - Navigasjonssystemer, Kompendium*. Department of Electronics and Telecommunication.

[49] Bjoernar VIK, *Lecture Notes - Integrated Satellite and Inertial Navigation Systems*. Departmant of Engineering Cyberntetics.

[50] FAT Filesystem for uControllers, *Elm-Chan*. http://elm-chan.org/fsw/ff/en/appnote.html#license

[51] Tor ONSHUS, *Instrumenterings Systemer*. Departmant of Engineering Cyberntetics, 5.ed 2011.

[52] Magnus Andersen - Svein Hovland., *Lab Vinkelmling: TTK410 Kybernetikk Introduksjon*. Departmant of Engineering Cyberntetics, Spring 2005.

[53] DiyDrones, *Forum*. http://diydrones.com/forum/topics/atmel-inertial-two-atavrsbin2-arduino?id=705844%3ATopic%3A719308&page=2#comments

15

OpenLOG error data

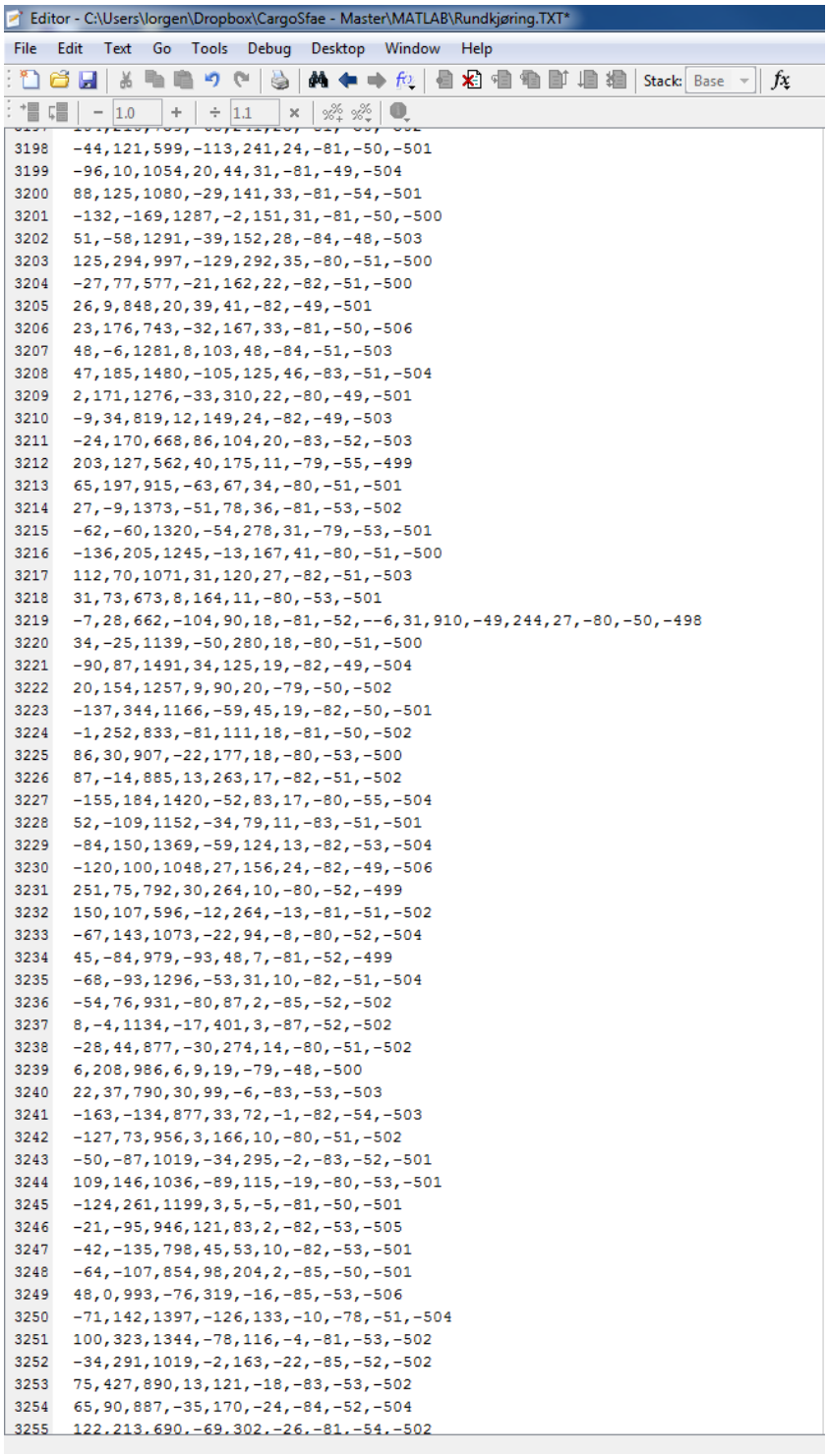


Figure 30: Corrupted data in OpenLog file

16 Appendix B - Schematics

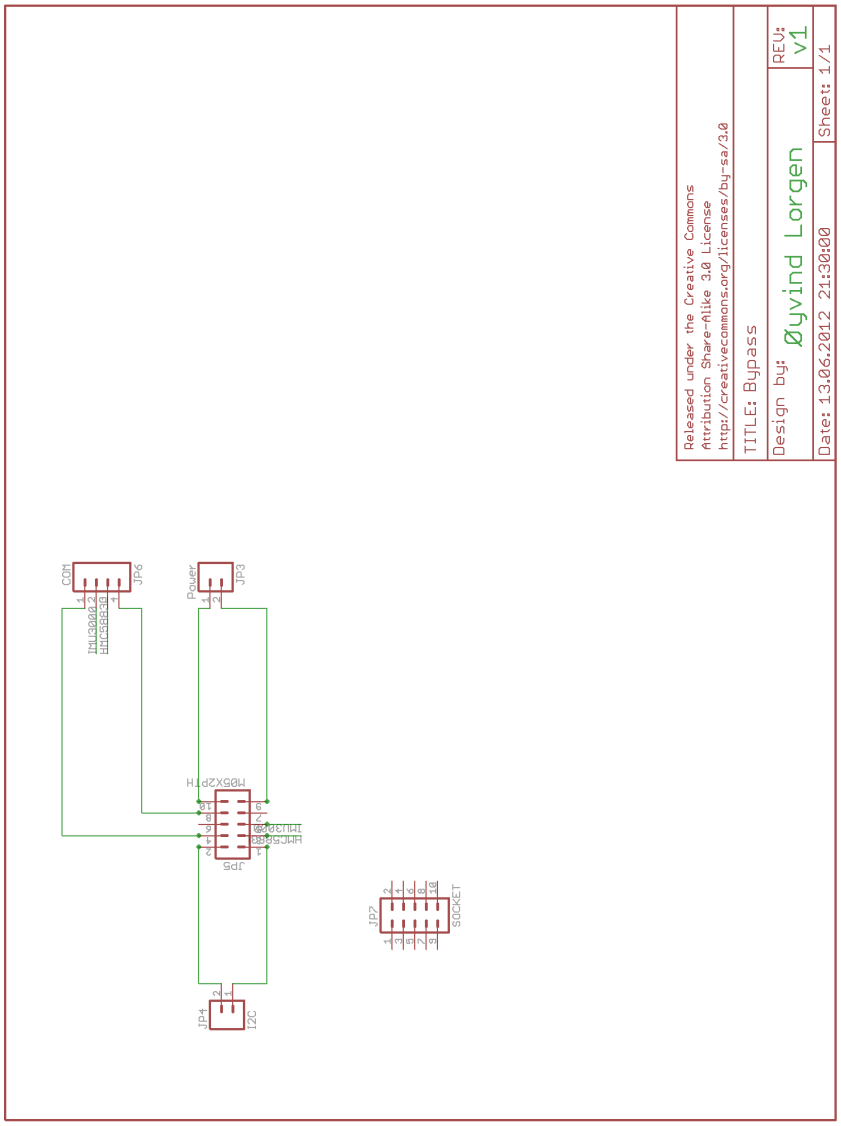


Figure 31: External socket for the ATAVRSBIN2

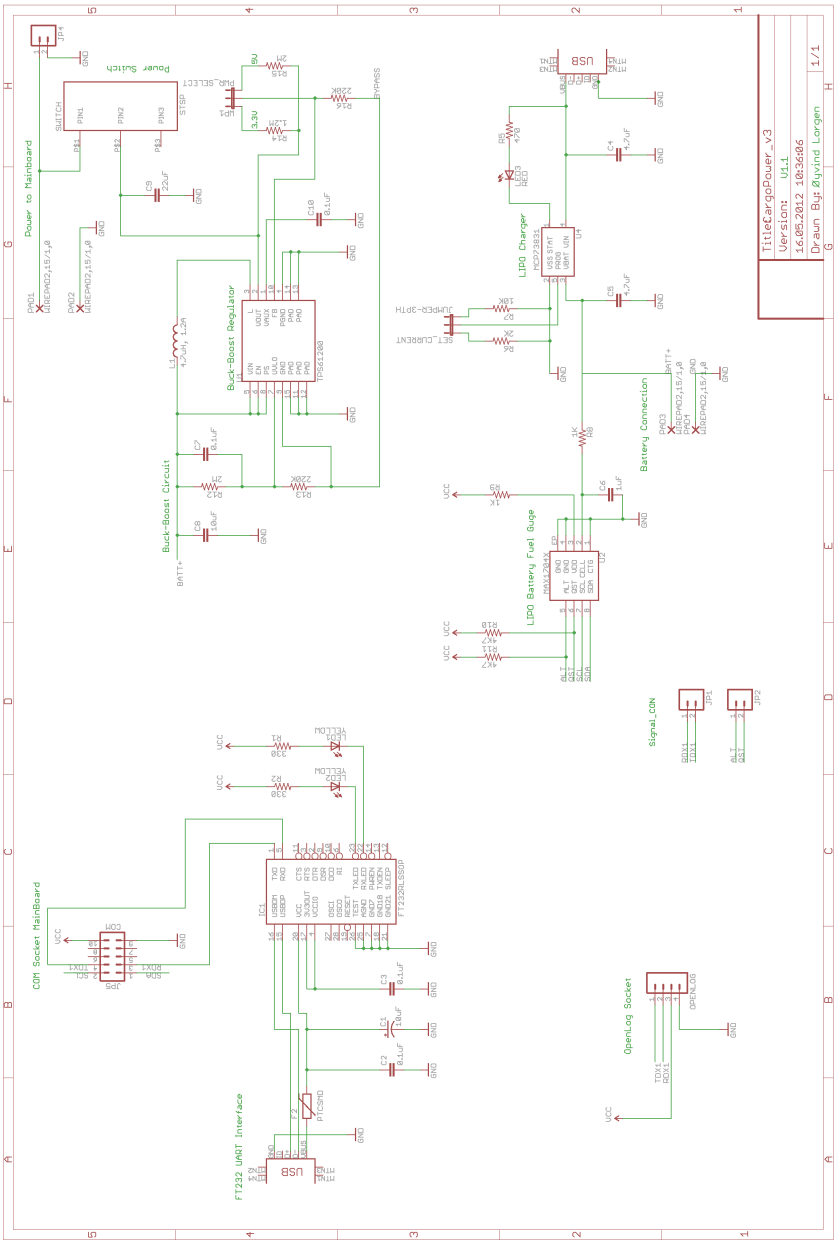


Figure 32: Power Board Schematic

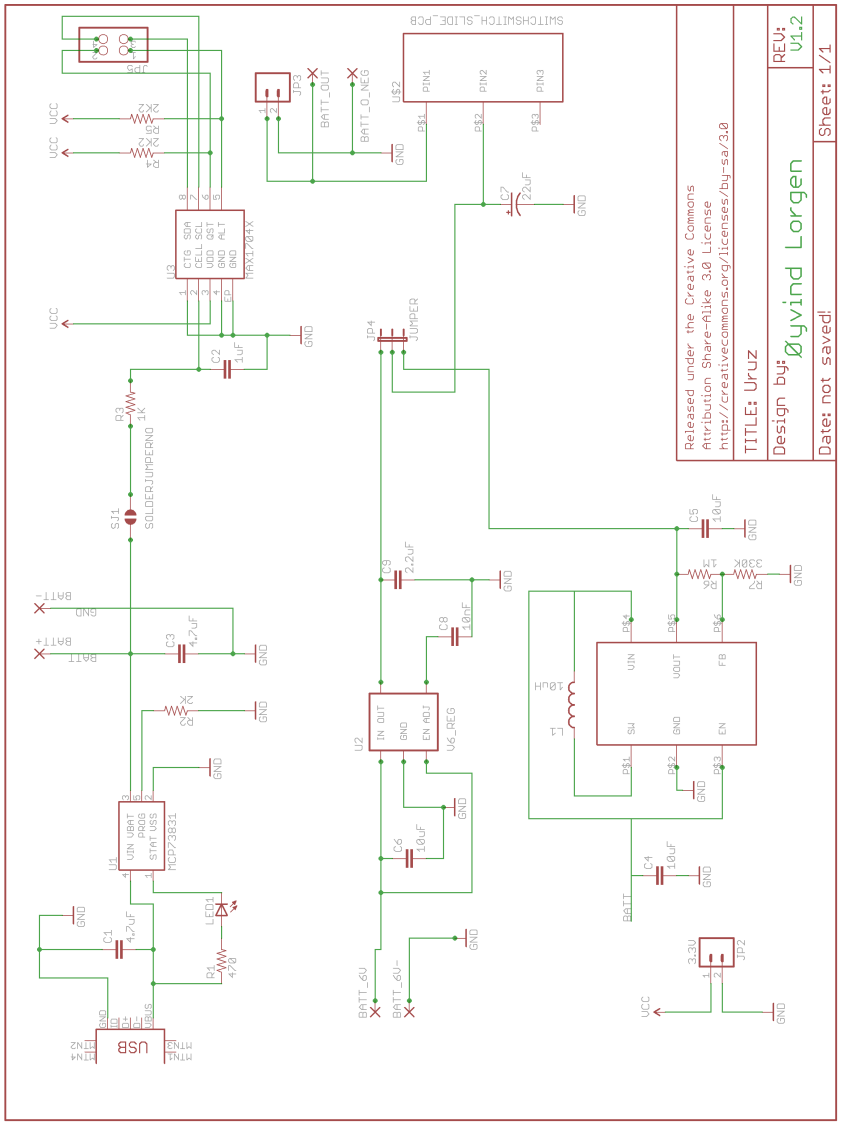


Figure 33: Power Board v2 Schematic

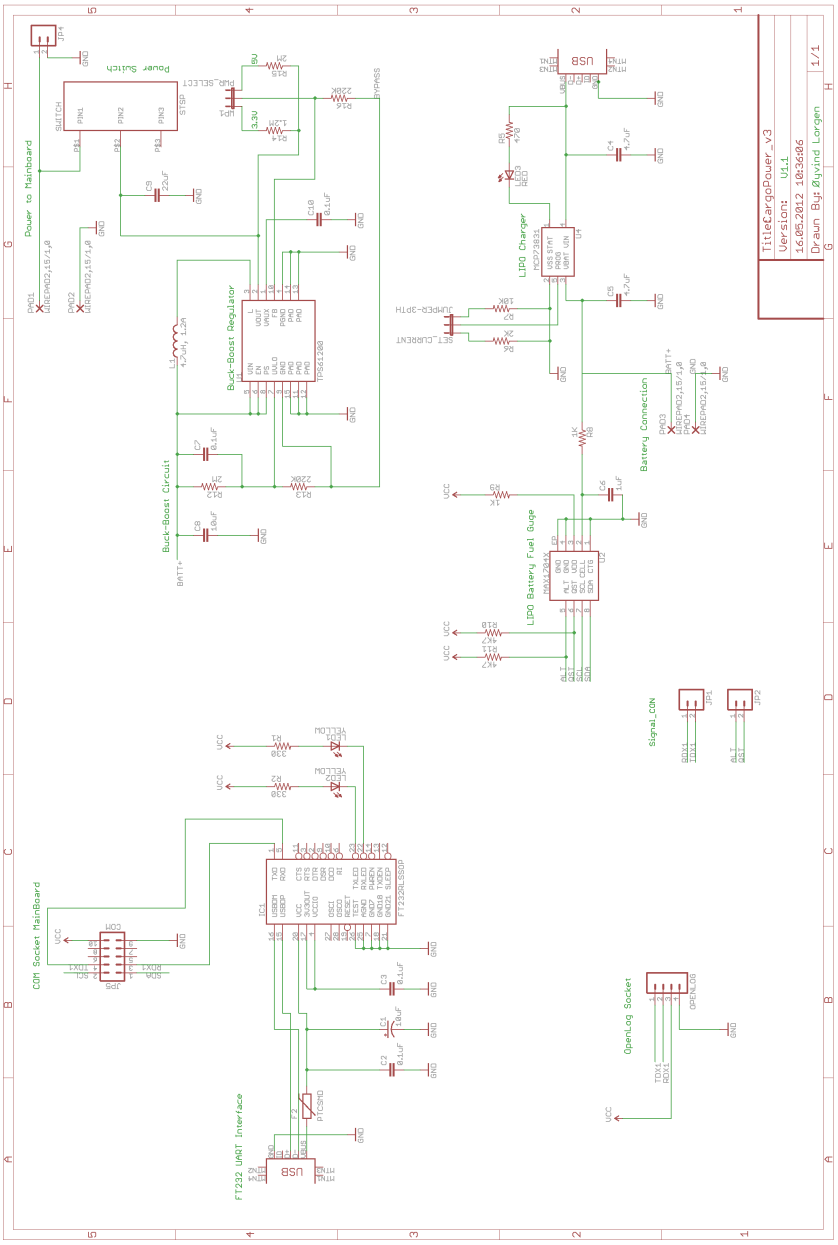


Figure 34: Communication Board Schematic

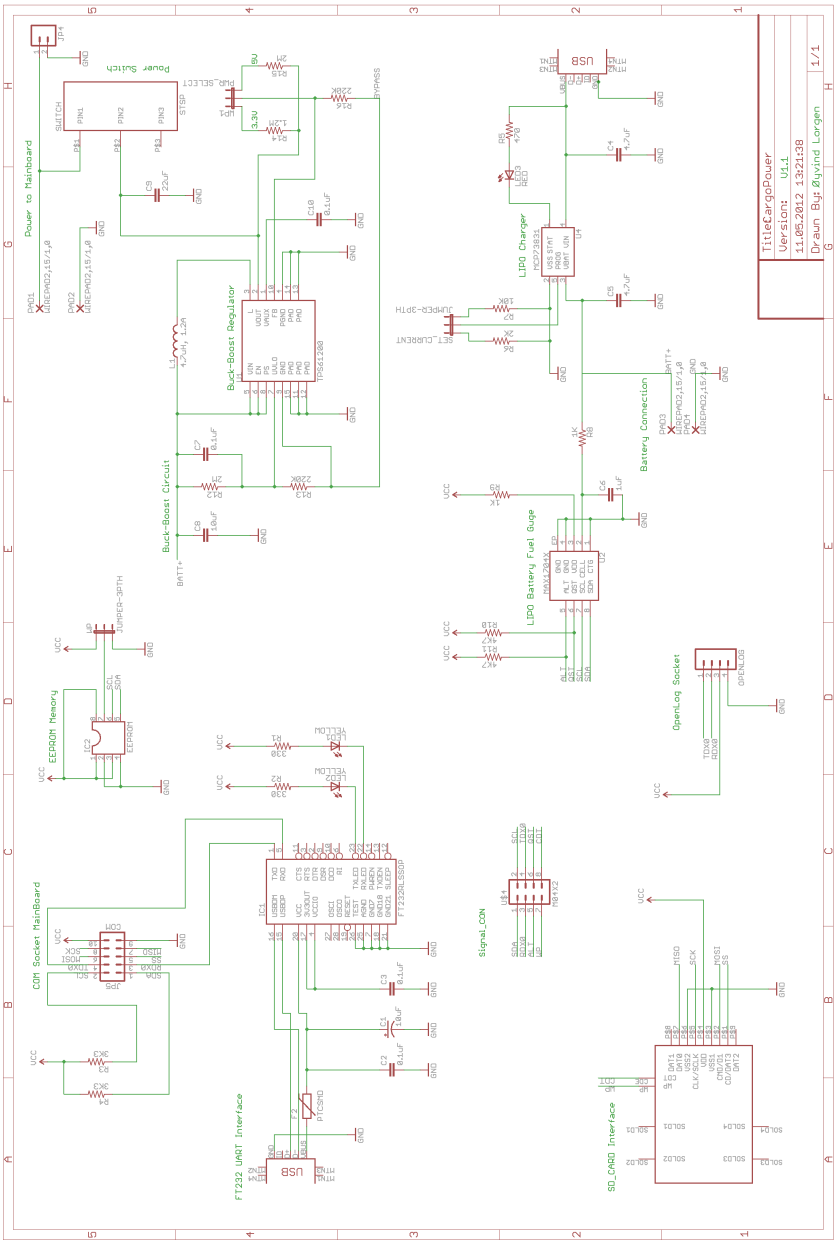


Figure 35: Proposed merged solution Schematic

17 Appendix C

17.1 Bypass - Gerber

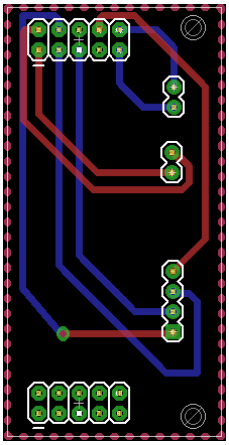


Figure 36: Gerber file

17.2 Powerboard - Gerber

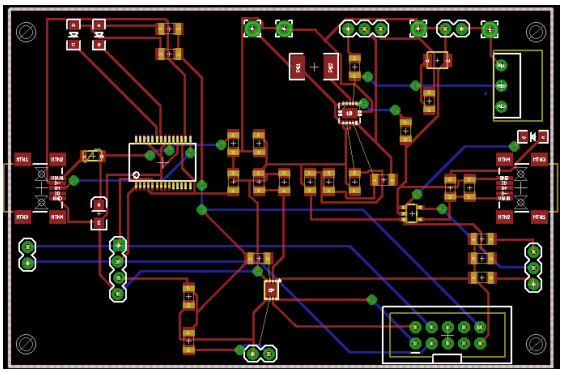


Figure 37: Gerber file

17.3 Uruz - 3D

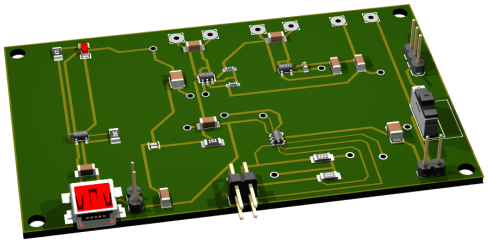


Figure 38: 3D Picture

17.4 Uruz - Gerber

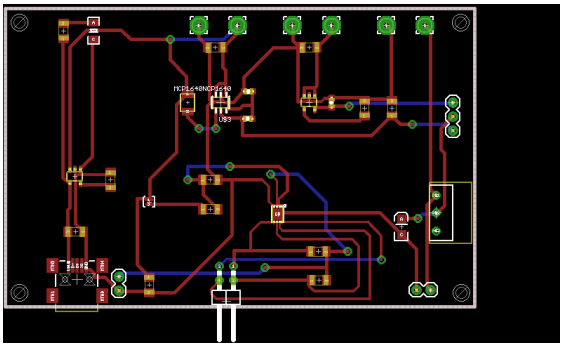


Figure 39: Gerber file

17.5 Anzus - 3D

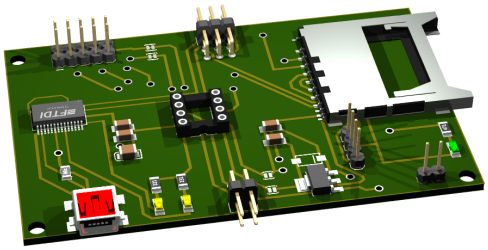


Figure 40: 3D Picture

17.6 Anzus - Gerber

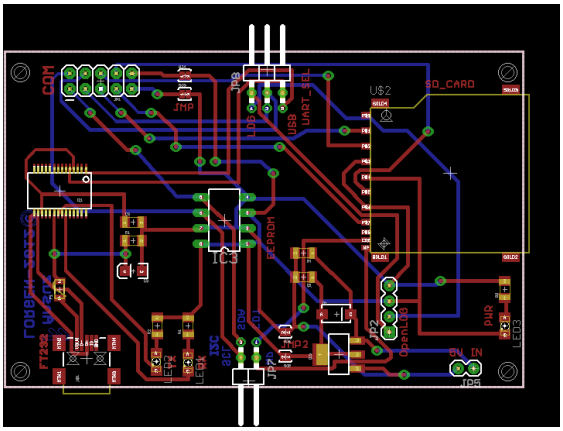


Figure 41: Gerber file