WORKSHEETS ON DRAGANFLYER X-PRO

Group 937a	Group 937b
Jakob Bjørn	Ole Binderup
Morten Kjærgaard	Jesper Haukrogh
Martin Sørensen	Sigurgeir Gislason

JANUARY 3, 2007

This collection of worksheets documents the work of 06gr937a and 06gr937b on the Draganflyer X-Pro (X-Pro). The work is an extension of the 8'th semester project done by members in both groups.
*The worksheets are arranged in order of occurrence and the worksheets highlighted with a * indicates the documents with highest relevance of the continues of the work on the X-Pro.*

After all the worksheets, a summary of what has been accomplished, incomplete tasks and two project proposals for the 10'th semester are located.

TABLE OF CONTENTS

I * Mechanical vibrations **9**

1 Mechanical Vibrations **10**

1.1 Problem description 10
1.2 Suggested Improvements 10
1.3 Improvements 11
1.4 Vibration Tests 11
1.5 Conclusion 13
1.6 Email from Hoffmann 14

II Hardware architecture and interfaces **15**

1 Hardware architecture **16**

1.1 Overall connection chart 16
1.2 Module description 17

III Setup of the gumstix™, robostix™ and DHM **23**

1 Setup of the gumstix™, robostix™ and DHM **24**

1.1 System description 24
1.2 Installing Buildroot on the Development Host Machine (DHM) 25
1.3 robostix™ cross-complier 25
1.4 Booting up the robostix™ 26

IV Power drive for the X-Pro **27**

1 Power drives in general **28**

1.1 Power transistors 28
1.2 Gate drive circuit 30

2 Current power drive **32**

3 New power drive **34**

3.1 Gate driver IC 34
3.2 Switching device 35
3.3 Protection circuits 35
3.4 Switching frequency 36
3.5 Test and verication of new power drive 37

4 Interface to radio receiver **40**

V Preliminary test record for investigating magnetic field in likely magnetometer placements on the X-Pro during operation **42**

1 Analysis **43**
1.1 Disturbances 43
1.2 Test equipment 43
1.3 Test cases 44

2 Test results **46**
2.1 Test case 1 - Rotation about ${}^{m}\hat{Z}$ 47
2.2 Test case 2 - Rotation about ${}^{m}\hat{X}$ 48
2.3 Test case 3 - Rotation about ${}^{m}\hat{Y}$ 49
2.4 Test case 4 - Start-up sequence 50
2.5 Test case 5 - Throttle 51
2.6 Test case 6 - Yaw 52
2.7 Test case 7 - Pitch 53
2.8 Test case 8 - Roll 54
2.9 Test case 9 - Iron block 55
2.10 Test case 10 - Yaw 2 56

3 Conclusion **57**

VI First contact with the SRF08 Ultra sonic range finder. **58**

1 The sensor **59**

2 Interface **60**

2.1 Media, physical layer 60
2.2 Data Link Layer 60
2.3 Presentation Layer 63

VII Economical budget for the X-Pro 65

1 Economical budget 66

VIII Analysis of the power consumption by the additional electronics added to the Draganflyer X-Pro 67

1 Power consumption 68
1.1 Introduction 68
1.2 Specifications 68

IX * Weight and flight time 69

1 Expected flight time performance 71
1.1 Total weight estimate 71
1.2 Thrust needed in hover 72
1.3 Hover time 73

2 Considerations on optimal construction 74
2.1 Weight 74

X * GPS module for the Draganflyer X-Pro 76

1 Introduction 77

2 Hardware setup 78

3 Serial communication 79
3.1 Protocol specification 79
3.2 Module configuration 80

4 Navigation **83**
4.1 Extraction of the position estimate 86
4.2 Implementation documentation 88

XI * Magnetometer **91**

1 Sensor specification **92**

2 Interface **93**
2.1 Physical layer / Media 93
2.2 Datalink layer 94
2.3 Presentation layer 95

3 First contact **98**

XII * robostix™ interface and SW design. **101**

1 robostix™ main process **103**
1.1 robostix™ supervisor design 103

2 robostix™ hardware interface **107**
2.1 General description of the robostix™'s pin set up 107
2.2 Input/Output (I/O) pin assignment 107

3 robostix™ driver design **112**
3.1 Assignment of timers 112
3.2 Motor driver 116
3.3 Magnetometer interface 117
3.4 Tachometer interface 120
3.5 Remote Control (R/C) interface 122
3.6 Inertial Measurement Unit (IMU) interface 126

XIII * gumstix™ and robostix™ communication **130**

1 Analysis **131**

1.1 Data for transmission 131
1.2 Inter-Integrated Circuit (I^2C) bus connection 132

2 Design **137**
2.1 Package contents 137
2.2 Complete package structure 140

XIV * Printed Circuit Board (PCB) design on the X-Pro **143**

1 PCB design on the X-Pro **144**
1.1 Hardware driver design 144
1.2 PCB design and layout 144
1.3 Additions made to the PCB 148

XV * Considerations on EMI and EMC and changes to design **149**

1 Problem areas **150**

2 The accommodations **151**

XVI Design of manual control **153**

1 Design of manual control **154**
1.1 Incoming signals 154
1.2 Mapping of steering signals to rotor Thrust 155
1.3 Range of T_{max} 159
1.4 Block Diagram of system 160

XVII * Closure of the 9'th semester **162**

1 Current project status **163**
1.1 Preanalyses 163
1.2 Rebuilding of X-Pro hardware 164
1.3 robostix^{TM} SW 165

1.4 gumstix^{TM} SW 166
1.5 Ground station 168

2 Incomplete tasks **169**

3 Plan for the 10'th semester **170**
3.1 Estimation group 170
3.2 Control group 170
3.3 Discussion of areas relevant for both groups 170

Abbreviations **172**

WORKSHEET I: * MECHANICAL VIBRATIONS

14/9-2006

From our last semesters project, the Draganflyer X-Pro (X-Pro) was concluded to vibrate too much mechanically, causing the linear accelerometer ($\pm 2G$) to saturate. The vibration was measured to be about 10 times greater along the z-axis compared to the vibrations measured in the xy-plane [ABG+06]. The cause of these vibrations and suggestions of improvements are the topics of this worksheet. Excessive mechanical vibrations has also been problem of groups doing similar projects as mentioned in 1.6 on page 14.

CONTENTS OF WORKSHEET I

1 Mechanical Vibrations **10**

1.1 Problem description 10
1.2 Suggested Improvements 10
1.3 Improvements 11
1.4 Vibration Tests 11
1.5 Conclusion 13
1.6 Email from Hoffmann 14

MECHANICAL VIBRATIONS 1

1.1 PROBLEM DESCRIPTION

The X-Pro is exposed to mechanical vibrations, the source of the mechanical vibrations is located at the motors/rotors. The rotorblades and motor rotate around their axis, the rotation speed makes them more sensitive for mechanical vibration, as they deviate from being perfectly balanced. The vibrations transmit along the rotor-arms and to the chassis where the sensors are located. The Draganflyer was shipped with nylon mounting-screws, for the purpose of connecting various parts of the Draganflyer. It has been observed that these do not hold the various body parts as tightly together as they originally did. Excessive mechanical vibrations and general wearing-down might have changed the nylon screws form slightly.

1.2 SUGGESTED IMPROVEMENTS

To decrease the vibrations of the X-Pro structural changes are suggested in the following. The suggested improvements/changes are furthermore discussed to determine if the changes will decrease the vibrations. From [Rao04] it is known that the vibration frequencies of a mechanical system can be increased by either increase the overall stiffness of the system and/or decrease the mass of the vibrating part. With an increase in the natural frequency the amplitude of the vibrations should be decreased.

Idea #1: Replace the original nylon-screws, at the bottom and the ones that are used to connect the rotor arms to the chassis, with screws that allows to tighten the construction more.
Expected result: By tightening the arms to the chassis the overall stiffness of the system should be increased, which thereby should increase the natural frequency of the combined system.

Idea #2: Mount a supporting string between the motor arms in order to restrict the vibrating movements, or even mount light "beams", as illustrated in Figure 1.1.
Expected result: By connecting the four arms to each other it should decrease the vibrations in the x, and y-plane. Furthermore because the way the arms of the X-Pro are mounted a string between the arms would decrease the vibrations in the z-plane as well.

Idea #3: Isolate the sensors by a mass damping system, by inserting a damping material between the place where the sensors are mounted and the source of the vibrations,

i.e. the motors/rotors.

Expected result: It is expected that by inserting the damping material on the X-Pro is should decrease the amount of disturbing vibrations the sensors are exposed to.

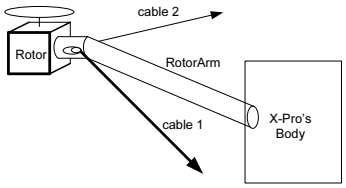

FIGURE 1.1: *Rotor arm attached with strings.*

1.3 IMPROVEMENTS

In the previous section a number of ideas to minimize the disturbances, the X-Pro is exposed to, are suggested. These ideas are discussed in the following.

Idea #1: The nylon-screws have been exchanged with metal screws, which tightened the construction of the X-Pro significant.

Idea #2: A cord has been drawn between the four arms where the motor and rotor are connected. The cord was drawn as drawn at that position to minimize the possible movement each motor/rotor can move, see Figure 1.1.

Idea #3: If a damping material should be placed on the X-Pro it would change the dynamics of the X-Pro and in addition to this be hard to get the sensors isolated completely from the rest of the X-Pro which vibrates.

1.4 VIBRATION TESTS

The measurements were performed where an accelerometer was placed on top of the battery in the CM. The accelerometer's range is $\pm 10G$, and it measures $14\,\mathrm{mV}/G$. The first measurement mentioned was taken before any restricting modifications were applied to the X-Pro, as shown on the graph in Figure 1.2. The frequency seems to be quite excessive, due to model parts each vibrating on it's own frequency and the G-effect is approximately shown to be $\pm 6G$.

The later measurement mentioned was taken after restricting modifications were applied to the Draganflyer, as shown on graph in Figure 1.3. The frequency seems to be

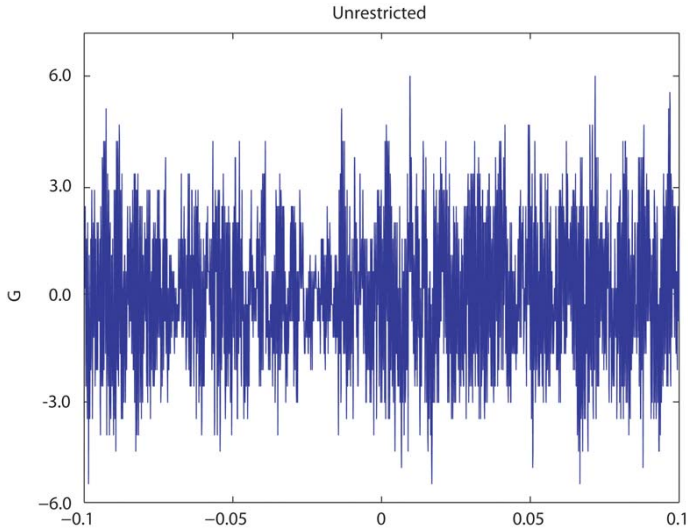

FIGURE 1.2: *Measurements from unrestricted X-Pro.*

lower as there is a mechanical vibration of a single system and the G-effect is approximately shown to be $\pm 0.5G$.

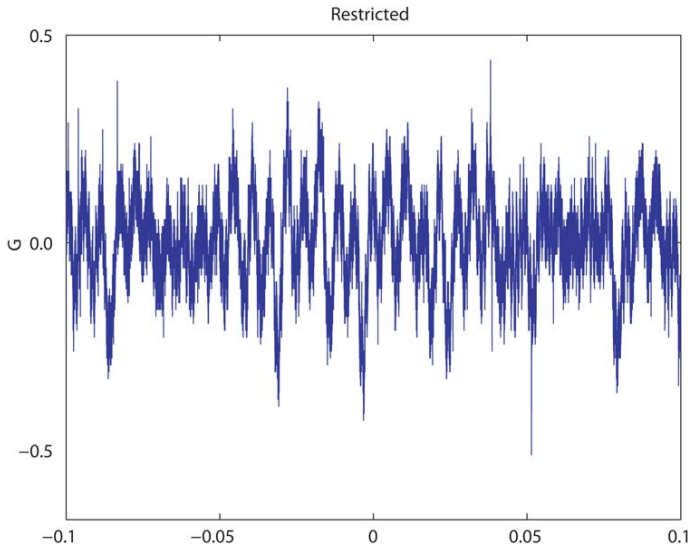

FIGURE 1.3: *Measurements from restricted X-Pro.*

To find out whether these improvements were enough to keep the linear accelerometers, a test with the originally used sensor was conducted. The test results can be seen in Figure 1.4. It is not possible to see actual measurements at $-2G$, as the accelerometer has an internal 5^{th} order Elliptic filter [O-N]. The figure reveals a vibration of $\pm 0.5G$, but also, that the sensor still saturates, as the mean value is increased when increasing the throttle.

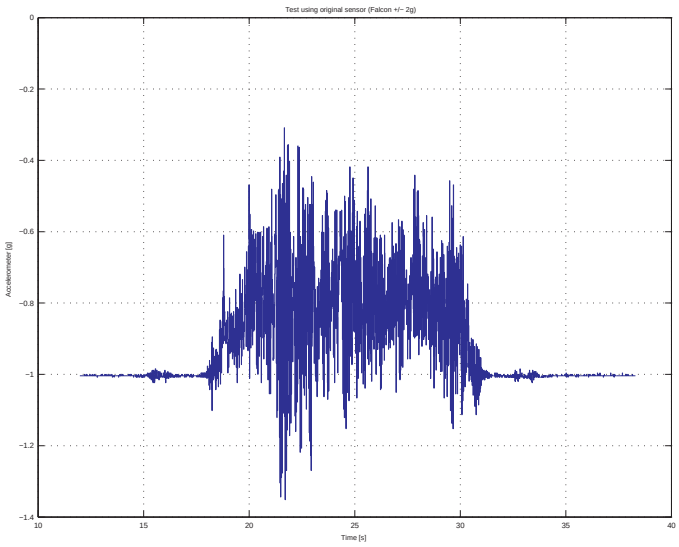

FIGURE 1.4: *Measurements using Falcon IMU z-axis accelerometer.*

1.5 CONCLUSION

The nylon-screws were replaced with metal-screws and the added support-string combined, showed to reduce the G-effect due to mechanical vibration significant. The trade-off was that, the metal mounting-screws increased the weight with approximately $25gr$.

The result measured from the restricted X-Pro indicated that there was mechanical vibration in the area of $\pm 0.5G$ which showed to be a great improvement from what it was before. The gravity force adds $1G$ to the equation, leaving a $0.5G$ "buffer area" in best case scenario, before saturation starts to influence the measurements. This is illustrated in Figure 1.5. Based upon the measurements from last semester and this semester's test using the original accelerometer, it is concluded that $0.5G$ "buffer area" does not lie within acceptable range of the $\pm 2G$ accelerometer, and therefore we suggest the use of no less than $\pm 5G$ accelerometer.

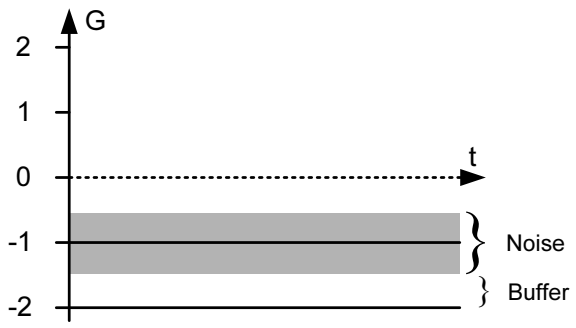

FIGURE 1.5: *Acceleration noise disturbance at steady-state.*

1.6 EMAIL FROM HOFFMANN

Date: Mon, 18 Sep 2006 09:33:04 -0700 From: "Gabe Hoffmann" <gabeh@stanford.edu> Sender: gabe.hoffmann@gmail.com To: "Sigurgeir Gislason" <sigurgeirg@hotmail.com> Subject: Re: Request regarding Draganflyer

Hi Siggy,

We're happy to hear about your project.

With the two vehicles that we use, we have not encountered this problem as severely as you describe. However, a group at MIT, in John How's lab, tried to use the same vehicle as you, and ran into a similar problem. The X-Pro seems to vibrate excessively, compared to the smaller Draganflyer aircraft. We currently make our own, using outrunner brushless motors. They don't have any gears or pulleys, so they have relatively low vibration.

We did have worse vibrations in an early model. We solved that problem by making the frame stiffer. That kept the vibrational modes of the booms, on which the motors were mounted, from being excited. That cut the vibration in half, roughly. To do this, we connected points near the motors with carbon fiber rods.

Finally, you can try using a different IMU, with +/- 5g or 10g. That's clearly not the preferable solution, but we currently have +/-5g, and that works nicely.

A final note is that excessive vibration can corrupt the rate gyros in the IMU as well, so watch for inaccuracies in those gyros associated with vibration. (the rate gyros use tuning forks essentially to measure rate). It's always nicer not to have excessive vibrations. Isolating a light weight IMU is difficult, though, and all of our attempts to do so failed.

Cheers, Gabe

On 9/17/06, Sigurgeir Gislason <sigurgeirg@hotmail.com> wrote:

Hi Gabe,

I am a student at Aalborg University, Denmark, studying M.Sc. taking a part in a group re-constructing a Quadrotor model for an autonomous flight as our masters project. The model we work with is Draganflyer X-Pro. We have constructed a rough dynamical model of it and have measured on it and come across the problem that manifests in mechanical vibrations that cause the accellerometers on our IMU to saturate +/-(2g).

The questions we'd like to ask you/your group are: Have you come across this problem? How have you decreased the effects of the mechanical vibrations on the accellerometers?

with regards, Sigurgeir (Siggy) Gislason group 06gr937 sgis03@control.aau.dk

WORKSHEET II: HARDWARE ARCHITECTURE AND INTERFACES

29/9-2006

CONTENTS OF WORKSHEET II

1 Hardware architecture **16**

1.1 Overall connection chart . . . 16

1.2 Module description . . . 17

1.2.1 Robostix . . . 17

1.2.2 Gumstix . . . 18

1.2.3 Wifistix . . . 19

1.2.4 GPS module . . . 19

1.2.5 SODAR (Ultra sonic range finder) . . . 20

1.2.6 Inertial Measurement Unit (IMU) . . . 20

1.2.7 Magnetometer . . . 21

1.2.8 Rotor speed sensor . . . 22

1.2.9 Radio receiver . . . 22

1.2.10 Safety interrupt module . . . 22

HARDWARE ARCHITECTURE 1

The following description concerning the connection of all hardware interfaces. First the overall system will be depicted and each component listed with their respective features. Later each interface will be described. The idea of this document is that is to be a ad hoc worksheet, that will be the reference in the implementation.

1.1 OVERALL CONNECTION CHART

The base in the configuration is the gumstix^{TM} solution which consist of the Robostix, the Gumstix and the Wifistix. The sensors, actuators, communication and security equipment are connected to this processor mainstay.

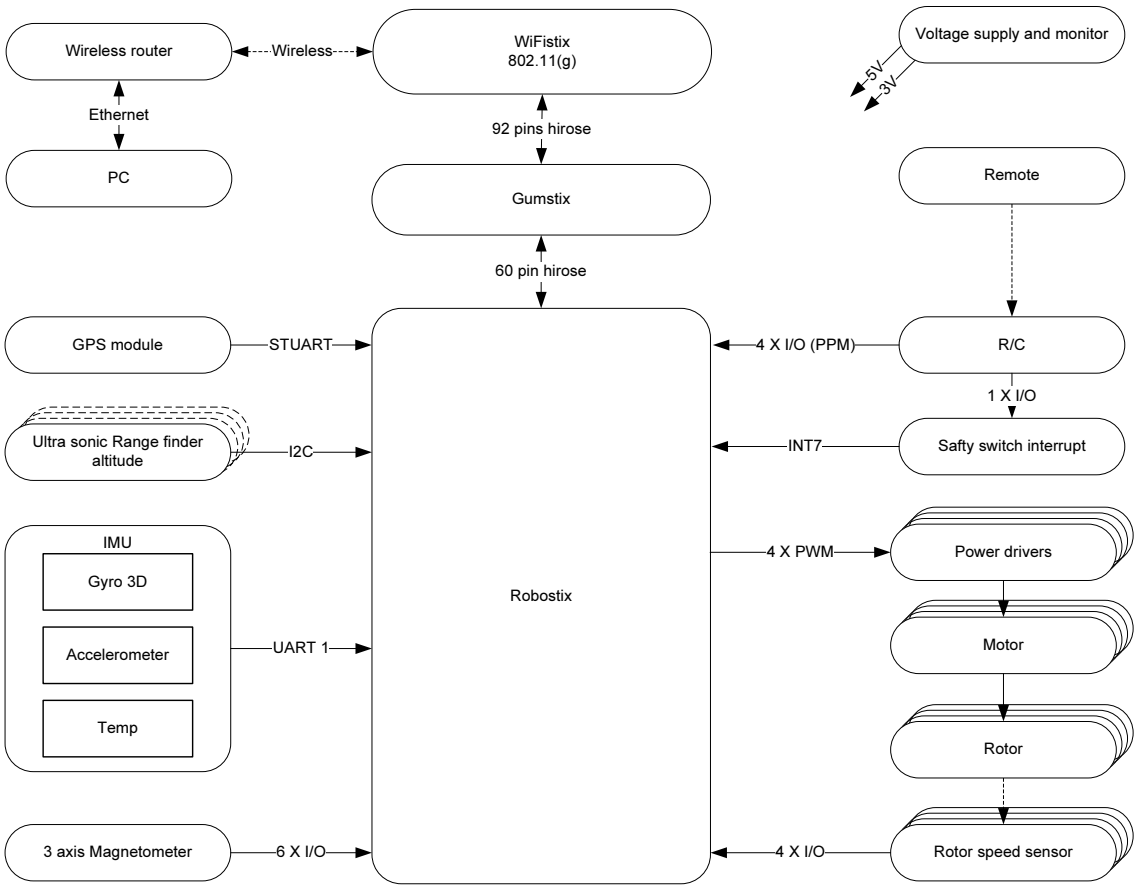

FIGURE 1.1: *Hardware communication chart.*

1.2 MODULE DESCRIPTION

In this section the main features regarding the individual modules are listed.

1.2.1 Robostix

The Robostix board is provides power for gumstix and wifistix. The input voltage should be between 4.5V to 5.5V. This voltage level is then converted to 3V before it is connected to the 60pin hirose attached to the gumstix. The main features of the robostix are listed in the following:

CPU and memory

- Max 16 Mhz operating clock
- 128k bytes of reprogrammable flash
- 4k bytes of Electrically Erasable Programmable Read Only Memory (EEPROM)
- 4k bytes of Static Random Access Memory (SRAM)
- 32 $\times$ 8 general purpose working registers

Peripheral features

- 2 $\times$ 8bit Pulse Width Modulation (PWM) channels
- 6 $\times$ 2-16bit PWM channels
- 8 $\times$ 10bit Analog/Digital Converter (ADC)
- 2 $\times$ Universal Asynchronous Receiver Transmitter (UART) (connect to Atmega)
- Inter-Integrated Circuit (I^2C) bus (two-wire serial interface)
- 2 $\times$ external interrupts
- min 16 $\times$ General Purpose Input/Output (GPIO)

Programming

- Serial Peripheral Interface (SPI)
- Joint Test Action Group (JTAG) for debugging

- Watchdog timer
- Software selectable clock frequency

Relay to gumstix

- STUART (up to 230kbaud)
- FFUART (up to 230kbaud)

1.2.2 Gumstix

The main computing power is stationed on the gumstix motherboard. The board is of the type connex 400xm which then include a 92-pin bus header (connects to the wifistix) and a 60 pins hirose (connects to the robostix).

CPU and memory

- 400MHz powerPC
- 64MB SDRAM standard (100MHz)
- 16MB Flash RAM

Peripheral features

- STUART (rates up to 230baud)
- FFUART normally used by console can be used for other communication if configured (rates up to 230baud)
- I^2C bus

Programming

- Linux kernel 2.6.10 in flash
- u-boot - bootloader
- possible to extend JTAG features

1.2.3 Wifistix

Fully configurable wireless board, following the 802.11(g) standard.

- Rates up to 54 Mb/s
- Transmit power 3mW - 79mW
- Passive rubber antenna
- Power supply through 92 pins hirose
- Communication range is at least 10 m, and the bandwidth will degrade on the distance. In the final software should the issue of a lost connection be handled. newly added

1.2.4 GPS module

The purpose of the GPS module is to provide a global absolute position estimate. The chosen module is an all in one solution from Ublox called SAM-LS.

Estimate features

- 4Hz position update rate
- 2.5m Circular Error Probability (CEP) (50% of the time)
- 2.0m CEP with Differential Global Positioning System (DGPS) / Satellite Based Augmentation System (SBAS) (50% of the time depending on accuracy of correction data)
- 5.0m Spherical Error Probability (SEP) (50% of the time)
- 3.0m SEP with DGPS / SBAS (50% of the time depending on accuracy of correction data)

Hardware features

- 16 channel Global Positioning System (GPS) receiver
- 8192 simultaneous time-frequency search bins
- ANTARIS Positioning Engine

- ATR0600 RF front-end IC
- 512kB FLASH memory
- DGPS and SBAS (WAAS, EGNOS) support
- Operating Voltage 2.7 to 3.3 V
- Connect to gumstix through approx 15cm Flat Flex Cable (FFC)

1.2.5 SODAR (Ultra sonic range finder)

The SOund Detection And Ranging (SODAR) is a sensor that in principle works as a sonar. The chosen sensor is a Devantech SRF08 and has the following features. With this sensor it is chosen to estimate the distance to the ground, hence only one unit is needed.

- Range 3cm - 6m (realistic 3cm - 1.5m)
- 5V supply voltage
- I^2C bus standard with programmable bus address
- Variable gain on the received signal (used for range changing)

more information can be found on: `http://www.robotstorehk.com/srf08tech.pdf`

1.2.6 IMU

The IMU will provide angular rates and linear acceleration, these sensor types are influenced by temperature, hence many IMU come with temperature compensated output. In the following list is other features connected to this module.

Angular rate

- Angular Rate Range: $\pm 200^\circ$/sec
- Resolution: 0.30°/sec
- Bandwidth: 100Hz
- Filtering: 5^{th} Order Elliptic Low Pass Filter
- Start-up Time: <1 sec

Acceleration

- Acceleration Range: $\pm$10g
- Resolution: 4mg
- Bandwidth: 100Hz
- Filtering: 5TH Order Elliptic Low Pass Filter

Digital section

- Sampling Rate: 1, 10, 20, 50, 75 & 100Hz (User Selectable)
- ADC Resolution: 10bits
- Interface: RS-232 Serial speed 4800 - 115.2 baud (N81 - User Selectable)

Electrical

- Input Voltage: 5V (ś5%)
- Current: 89mA
- Power: 445mW

Physical

- Mass: 11.5 g
- Dimensions: 63.50 mm X 31.75 mm X 15.62 mm

1.2.7 Magnetometer

The magnetometer will provide the direction of the magnetic field, hopefully produced by the Earth. The chosen magnetometer is of the type Micromag3, and is a 3-axis magnetic field sensing module. Following are the features of the module listed:

- Voltage: 5 VDC after modification.
- Part size: 25.44 mm X 25.1 mm X 19 mm
- Field measurement range $\pm$1100 μT ($\pm$ 11 Gauss)
- Resolution: 0.015 μT (0.00015 Gauss)
- SPI protocol for interfacing to a processor.

1.2.8 Rotor speed sensor

The basis of the sensor consist of a hall effect sensor that detects a passing magnet that is mounted on the gear wheel of each rotor. When the magnet is near the hall sensor the signal out changes from 5V to 0V. This has proved to be a simple and reliable sensor for estimating the angular velocity of the rotor blades. The construction made by the 8th semester group, mounted one magnet on each gear. To make an even more precise estimate another magnet could be mounted on the opposite side of the gear.

1.2.9 Radio receiver

The radio receiver is to be used when the Draganflyer X-Pro (X-Pro) is manually flown. This will come in handy as a safety feature in the development of an autonomous X-Pro.

1.2.10 Safety interrupt module

The purpose of this module is to provide the possibility of changing between two different modes during flight using the Remote control. The idea is that the module generates an interrupt to the robostix if it is needed to take the X-Pro out of an autonomous control that has become hazardous, or to give back the control to the autonomous controller. This will come in handy in the integration process to ensure the security of X-pro and surrounding personnel and hardware. The switch E on the Remote Control (R/C) is intended to be used together with this feature.

WORKSHEET III: SETUP OF THE GUMSTIX^{TM}, ROBOSTIX^{TM} AND DHM

22/9-2006

In order to develop and compile software to the gumstix^{TM} and robostix^{TM} the right develop environment is required. The first chapter will concern system description and communication between the systems. The second chapter concerns the software required to develop software later on.

CONTENTS OF WORKSHEET III

1 Setup of the gumstix^{TM}, robostix^{TM} and DHM **24**

1.1 System description 24

1.2 Installing Buildroot on the Development Host Machine (DHM) 25

1.3 robostix^{TM} cross-complier 25

1.4 Booting up the robostix^{TM} 26

SETUP OF THE GUMSTIX™, ROBOSTIX™ AND DHM

1

1.1 SYSTEM DESCRIPTION

This chapter deals with the system description and the communication between the DHM and the gumstix™.

Table 1.1 shows the three platforms in the system.

Technical infomation	DHM	gumstix™	robostix™
CPU type	Intel Pentium© 4	Intel XScale© PXA255	ATMega128
CPU speed	2.8 GHz	400 MHz	16 MHz
Memory	512 MB RAM	64 MB RAM	4 KB SRAM
Harddisk space	16 GB	16 MB Flash	128 KB Flash
Operation System (OS)	Ubuntu 6.06.1	Linux 2.6-18gum	None

TABLE 1.1: *System description of the different components in the Development environment.*

The DHM is a faster machine compared to the gumstix™, which makes cross-compiling a lot faster. The gumstix™ is a "host" machine seen from the robostix™. This makes it possible to let the gumstix™ run hard jobs and let the robostix™ manage the simple things.

Figure 1.1 shows the communication between Lab-net, the DHM and the gumstix™.

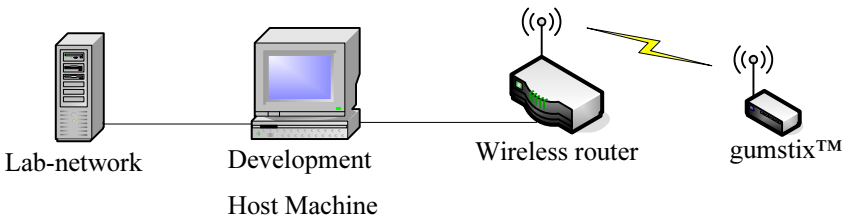

FIGURE 1.1: *The set up of the communication between the Lab-net, the DHM and the gumstix™.*

In order to have a relatively secure connection to the gumstix™, using a wireless router, and to be able to develop software on the DHM from a remote location the DHM must be connected to the Lab-net using one of the two Local Area Network (LAN) cards installed. The DHM is also connected to the wireless router's Wide Area Network (WAN) port making the DHM will be the intermediate link between the Lab-net and the gumstix™.

The DHMs connection to the wireless router's WAN port is set up by the DHM having a Dynamic Host Configuration Protocol (DHCP) server running on last LAN cards. All this makes it possible to make remote software development on the DHM from anywhere within the university network.

1.2 INSTALLING BUILDROOT ON THE DHM

In order to develop and compile software to the gumstix™ and robostix™ the following homepages are used to set up the compile environment

http://docwiki.gumstix.org/Buildroot_on_Ubuntu:
This describes which packages is needed to be installed on the DHM to make Buildroot compile on Ubuntu.

http://docwiki.gumstix.org/Buildroot:
This describes how to get Buildroot and the set up. To get Buildroot Subversion is used (in short svn). The current network setting on the DHM didn't allow Subversion to be downloaded, due to the university's proxy settings. The solution to this problem were to use one of the university servers to get the repository. After that copy the repository to the DHM.

http://docwiki.gumstix.org/Robostix:
This describes the basics about the robostix™ such as how to get the gumstix™ and robostix™ working together.

1.3 ROBOSTIX™ CROSS-COMPLIER

When designing and developing software the right developing tools are required. Therefore from the FAQ of the gumstix™ homepage there are descriptions on different development tools for the robostix™. The first url below is found on the FAQ.

http://docwiki.gumstix.org/Robostix:
This url describes the basics of the robostix™ and the modifications required to make the gumstix™ and robostix™ function properly when combined into one unit. The url also describes how to develop programs to the robostix™ and how to download the program to the robostix™ using the gumstix™.

http://docwiki.gumstix.org/Robostix_avr_gcc:
Toolchain for developing software for the robostix™. The url describes how to

install either a pre-build version of Toolchain or build it from source.

1.4 BOOTING UP THE ROBOSTIX^{TM}

The state of the robostix^{TM} is per default reset. In order to get the robostix^{TM} out of the default state at boot up the booting procedure of the gumstix^{TM} must contain the robostix^{TM}. This is described on the following url's

`http://docwiki.gumstix.org/Robostix_power:`
On this homepage the description on how to power up the robostix^{TM} if in reset mode.

`http://docwiki.gumstix.org/Robostix_uboot:`
This link describes the changes add to the u-boot environment. u-boot is the boot-loader the gumstix^{TM} system uses. The changes i strait forward. Changes is carried out on the system and it are working properly.

WORKSHEET IV: POWER DRIVE FOR THE DRAGANFLYER X-PRO (X-PRO)

14/9-2006

In order to fully augment the X-Pro into a complete autonomous platform, new motor power drives must be designed. The first section will concern general important issues when designing a DC motor power drive. When the proper knowledge has been aquired, the current X-Pro power drives will be reverse engineered in order to get a picture of how extensively the original design was made. This will in turn enable a discussion of possible improvements towards the new drives.

CONTENTS OF WORKSHEET IV

1 Power drives in general **28**

1.1 Power transistors 28

1.1.1 Protection of transistors 29

1.2 Gate drive circuit 30

1.2.1 Protection of gate drive 30

2 Current power drive **32**

2.0.2 Evaluation of current power drive 33

3 New power drive **34**

3.1 Gate driver IC 34

3.2 Switching device 35

3.3 Protection circuits 35

3.3.1 Turn-on snubber 35

3.3.2 Turn-off snubber 35

3.4 Switching frequency 36

3.5 Test and verication of new power drive 37

4 Interface to radio receiver **40**

4.0.1 Radio receiver $\Rightarrow$ Safety interrupt module 40

POWER DRIVES IN GENERAL 1

Power electronics are usually based on the switching of various semiconductor devices, [Vit95, p. 1]. In relation to motor drives, the most common devices are the so called power transistors which can provide sufficient power. A typical modulation used to control the switching of the power transistors is the Pulse Width Modulation (PWM) signal. Usually a controlling IC generates the PWM signal but often it cannot itself provide enough current to secure fast switching of the transistors. In order to switch the gates of a power transistor a dedicated driver IC capable of high current output is inserted into the circuit.

When high voltage and current spikes occur caused by e.g. fast switching, the mentioned sub circuits can be damaged an so protection of these must be considered. The diagram shown in Figure 1.1 depicts a general power drive configuration for a DC motor.

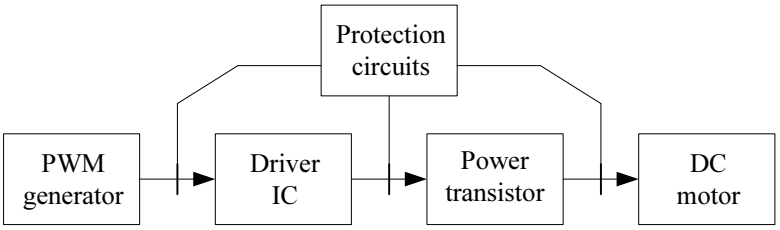

FIGURE 1.1: *Overview of a common DC motor power drive configuration.*

The remaining worksheet will concern the design of the three center blocks, driver IC, power transistors and protection circuits.

1.1 POWER TRANSISTORS

There are generally five types of power transistors, namely BJTs, MOSFETs, SITs, IGBTs and COOLMOS. There are several concerns when choosing a switch transistor, [Ras04, p. 16-19]. The key concerns are listed in the following:

Voltage ratings:

The switch must be able to withstand forward and reverse repetitive peak voltages (dv/dt). These can e.g. occur during the rapid turn-on turn-off of the switch and also when connecting the main supply. Furthermore the on-state forward voltage drop must be low tending to zero.

Current ratings:

A low off-state leakage current tending to zero is wanted and an ability to tolerate

repetitive and non-repetitive peak currents ($^{di}/_{dt}$). These peak currents occur when the current rises rapidly due to high frequent switching.

Switching losses:
When the forward current rises the forward voltage falls and vice versa. This leads to situations where high current and voltage appear simultaneously. This results in power dissipation in the device. According to the latter, fast switching characteristics of the device are of the essence.

Thermal resistance:
The power dissipation in the device must be rapidly moved from the internal wafer as the thermal capacity of a power transistor usually is too low to safely remove the heat. A high maximum junction temperature in the chosen device is however wanted.

1.1.1 Protection of transistors

The before mentioned key concerns are important to keep in mind when investigating the datasheets towards choosing the appropriate switch. However even in carefully designed power circuits with high tolerant switches, voltage and current transients and short circuit fault conditions may occur, [Ras04, p. 791]. Generally the power devices must be protected externally from the following:

- Thermal runaway
- High $^{dv}/_{dt}$ and $^{di}/_{dt}$
- Reverse recovery transients
- Supply transients
- Fault conditions

Addressing the first item, thermal runaway is always prevented by the use of some kind of heat sink. Calculations can be made to figure out how big a heat sink is needed in the given situation or if any heat sink is needed at all.

During switching, high $^{dv}/_{dt}$ and $^{di}/_{dt}$ can stress the device significantly. In order to prevent this, switching aid circuits or so called *snubbers* can be used. Among various snubber circuits, two are of the more common kind. The first is called a turn off snubber which basically limits the $^{dv}/_{dt}$. The second is called a turn on snubber which limits the

$^{di}/_{dt}$. Both snubber circuits help to keep the device within its Safe Operating Area (SOA), [Vit95, p. 43-46].

Usually a switch has a controlled forward current flow, and some devices like the MOSFET also have an integral reverse diode for reverse current transients. If for instance a BJT is used, an external diode leading potential harmful back-currents around the switch must be used, [Vit95, p. 50-51].

The last two items can be dealt with by introducing components such as an inductor to take care of a possible on-supply transient and a fuse in faulty conditions such as short circuits for instance.

1.2 GATE DRIVE CIRCUIT

To get the power transistors working properly, a gate driver is needed. Therefore it is important to gain knowledge of the gate characteristics in order to construct the correct gate driver circuit. The goal is to achieve high switching speed which means faster turn-on and turn-off time and high efficiency, which reduces the power loss in the transistor. This can be obtained by optimizing the gate driver circuit by applying various components to the default set up.

Protection of the gate driver circuit towards the transistor and the signal to the driver is important. Two protection topologies will be described, namely Pulse Transformers and Optocouplers. The need for one of these protection topologies are to protect the signal generator from peaks form either voltage or current.

1.2.1 Protection of gate drive

Pulse Transformers:

Pulse transformers are constructed with one primary winding and one or more secondary windings. This gives the ability to have multiple gate signals as an example parallel-connected transistors. When using a small switching frequency the transformer would saturate and its output would be distorted. The transformer gives the galvanic separation which is required.[Ras04, p. 769]

Optocouplers:

The optocoupler contains a Infrared Light-Emitting Diode (ILED) and a silicon photo-transistor. The ILED transmits the received signal to the photo-transistor which gives the galvanic separation from the signal generator and the gate driver. This also limits the turn-on and turn-off times due to the rise and fall times of the

ILED and photo-transistor.[Ras04, p. 769]

CURRENT POWER DRIVE 2

After a coarse reverse engineering process on the current power drives with focus put on the switch arrangement and safety circuitry, the following circuit diagram in Figure 2.1, has been derived.

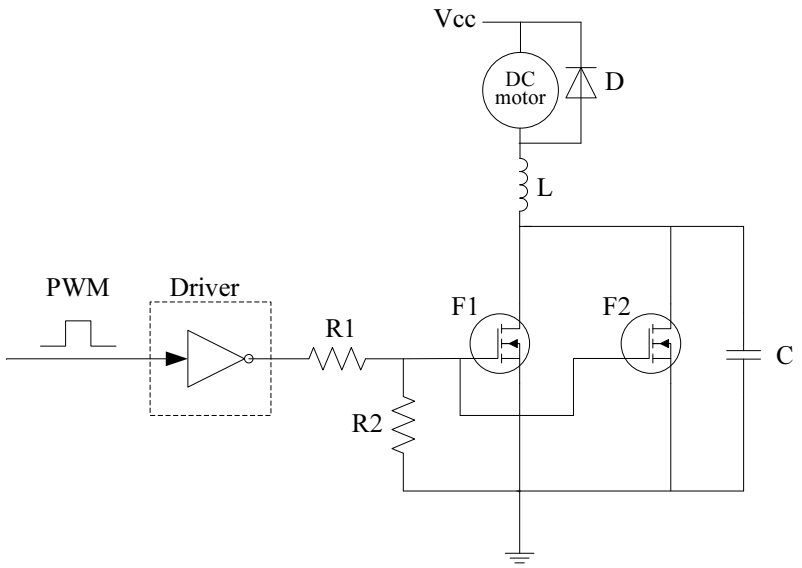

FIGURE 2.1: *Current power drive reverse engineered.*

As the Figure illustrates, two identical MOSFET switches has been used in a parallel setup presumably to divide the power dissipation between them. From the left the control PWM signal is applied to an opamp driver which is capable of driving the FETS. Resistances $R1$ and $R2$ serve as voltage dividers so that both transistors have the exact same working point. If this is not the case one MOSFET e.g. $F1$ could be forced to withstand a higher load than $F2$, potentially leading to a thermal runaway unaccounted for. The capacitor C seems to work as a low cost version of a turn off snubber according to Figure 1.26 in [Vit95, p. 43]. A turn on snubber also seems to be implemented in terms of the inductor L according to the configuration shown in Figure 1.29b in [Vit95, p.46]. The diode D is certain to serve as protection of inductive kickback commonly produced by the motor. The kickbacks are continuously led back into the inductor of the motor which is able to withstand the transients. If this configuration was not implemented the kickbacks would stress the FETs and most likely damage them.

The switching frequency is measured with an oscilloscope to be $180\,\mathrm{Hz}$

2.0.2 Evaluation of current power drive

This section is not aimed at degrading the current X-Pro power drive. It is merely aimed at discussing which features that could be potentially reused, left out or even enhanced.

The driver IC contains an opamp to provide sufficient current to drive the gate. A thumb rule indicates that the more current you drive your gate with the faster turn on and off times you get. This in turn increases the efficiency thereby lowering the heat dissipation in the switches. As can be seen from Figure 2.1, both MOSFETs are driven by one opamp. A new design could augment the driver IC block to contain two outputs, one for each FET in order to ensure sufficient current. This would however increase the component count and thereby weight.

One way to get around the latter could also be to use only one MOSFET. This would result in only one gate to be driven and both circuit complexity and component weight/count would be decreased. This would however set high demands for the single MOSFET in terms of especially current ratings.

Regarding the turn on snubber L, it seems as it can be left out. The reason is that in the ON state the two inductors, L and the motor inductance L_m are in series. Inductors add up in series like resistors do so unless L is quite large compared to L_m which is unlikely, then L can be omitted. In the new power drive, L_m could be used as the turn on snubber.

The turn off snubber could be augmented with a diode and resistor according to [Vit95, p. 42-44]. This could enhance the safety feature.

As mentioned before in section 1.2, the signal generator and all other connected electronics on the left side of the gate driver should be protected from possible voltage or current transients. This is however mostly common if more than one voltage supply is used introducing different ground planes. This must be the reason as to why the current power drive is implemented without e.g. an optocoupler. If a separate battery is to be installed for the low power part of the X-Pro, an optocoupler should be heavily considered in the power drive implementation.

NEW POWER DRIVE 3

With the knowledge perceived in the previous sections, the new power drives can be designed. There has been put weight on minimizing the mass of the power circuit, on protecting the switch elements and protecting the low power end towards various transients. Furthermore efficiency of the power circuit is highly regarded. The new design is depicted in Figure 3.1

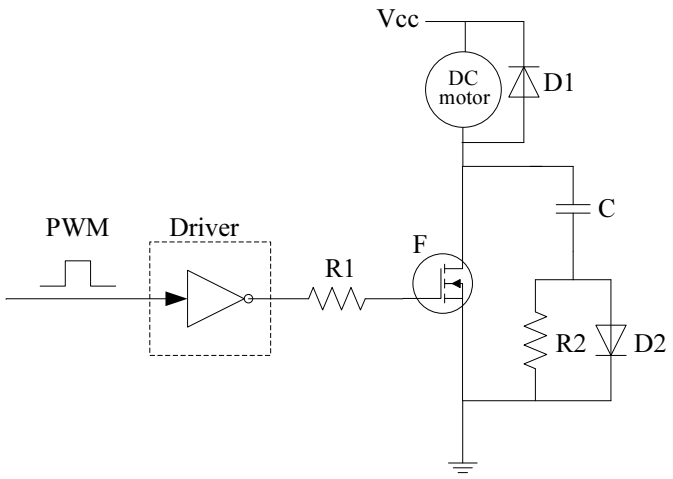

FIGURE 3.1: *The new X-Pro power drive circuit.*

In the following sections each part of the circuit will be described.

3.1 GATE DRIVER IC

The chosen gate driver is a ICL7667 with a dual output configuration. The datasheet can be found on the CD-ROM. As the circuit shown in Figure 3.1 is designed with only one switch, only two ICL7667 drivers are needed in the overall four power drive circuitry.

The driver output can provide up to $300\,\text{mA}$ and is manufactured specifically to drive MOSFETs. The high current output minimizes the time the switch has in the linear state, both when turning off and on.

A gate resistance $R1$ is recommended in the driver datasheet. The resistance should according to the datasheet lie in the region of $10\Omega - 30\Omega$ and is chosen to 18Ω. The small resistance reduces ringing but at the cost of a slightly increased rise and fall time.

3.2 SWITCHING DEVICE

According to the discussion in chapter 2.0.2, only one MOSFET SUP/SUB85N02-06 switch has been chosen in order to both minimize circuit complexity and weight. The datasheet can be found on the CD-ROM. The MOSFET can withstand 85 A continously, 240 A pulsed and has a maximum junction temperature of $175\,\mathrm{C}^\circ$ among some of its most important features. Because of its high current ratings only one SUP/SUB85N02-06 is necessary.

3.3 PROTECTION CIRCUITS

Recall Figure 1.1 on page 28 where protection circuits are added to every interface of the design. As mentioned in the gate driver section the pulse transformer or optocoupler can be implemented either between the PWM generator and driver IC or the driver IC and power transistor. The datasheet of the driver IC states that the driver is able to drive a standard pulse transformer. This would however introduce coils in terms of the transformer and thereby possibly Electro Magnetic Interference (EMI). An optocoupler would be the obvious choice however only if more than one battery supply is used. According to the latter no high power / low power separation is implemented in the new power drive at present time.

3.3.1 Turn-on snubber

In order to accommodate the frequent incoming back transients of the motor, a common fast recovery diode $D1$ is anti paralleled with the motor (ln6073). The motor inductance is $0.0013\,\mathrm{s} \cdot 0.209\Omega = 0.2717\,\mathrm{mH}$ [ABG$^+$06, p. 141] and thereby a great deal larger than the snubber inductance L from the original circuit measured to be $2\,\mathrm{uF}$. On this behalf the motor inductance is used as a turn on snubber.

3.3.2 Turn-off snubber

The turn off snubber circuit is augmented in contrast to the original power drive. The capacitor C, the resistor $R2$ and diode $D2$ are determined from the following, [Vit95, p. 44-45].

- According to the datasheet, the switch has a storage time of $0.8\mu\,\mathrm{s}$ and a fall time of $1\mu\,\mathrm{s}$.
- The current in the ON state is set to 15 A, which is a great deal more than the current

used in hover ($9 - 10$ A), [ABG+06, p. 156].

- The maximum value of V_{DS} should at all times stay below 20 V (datasheet).
- The peak discharge current of C should not exceed 6 A during turn ON. This value is approximated from [Vit95, p. 44].

The current through the switch is given by:

$$i = 15(1 - \frac{t}{t_f}) \tag{3.1}$$

and so the current though the capacitor is:

$$i_c = \frac{15t}{t_f} \tag{3.2}$$

The voltage across the capacitor at instant t will therefore be:

$$V_{DS} = \frac{1}{C}\int i_c dt = \frac{1}{C}\frac{15t^2}{2t_f} \tag{3.3}$$

This is equated to $V_{DS} = 20$ V at $t = t_f = 10^{-6}$ s and the snubber capacitor is calculated to $C = 0.375\mu$F. The resistance $R2$ is determined through $R2 = 20\,\text{V}/6\,\text{A} = 3.33\Omega$. The diode $D2$ is a fast recovery diode (ln4007). Now that the circuit is designed the only thing left is to determine the switching frequency.
The functionality of this circuit is as follows. When the switch is in the ON state the voltage across it and therefore the switching circuit is nearly zero. When the switch quickly turns to the OFF state, the capacitor-diode combination slows down the rate of rise of voltage across the switch. This happens because during this time the diode turns ON and the capacitor starts charging. During the OFF state the capacitor remains charging to the full switch blocking voltage. During the next turn ON the capacitor starts to discharge. The resistor is for the purpose of limiting the peak value of the discharge current through the transistor. Each time the transistor turns ON the total energy stored in the capacitor is dissipated in the resistor.

3.4 SWITCHING FREQUENCY

The immediate issue to address when determining the switching frequency would be the size of the mechanical time constant of the motor. According to [ABG+06, p. 143], the mechanical time constant with gear wheel is, $\tau_{em} = 0.03$ s. The time constant is assumed

larger than this because of the inertia added by the rotors, however $0.03\,\mathrm{s}$ will be used in the PWM frequency approximation. The mentioned time constant is measured at 63%, which means that it reaches 63%, $1/0.03 = 33.3\,\mathrm{Hz}$ times per second. If this number is multiplied with a factor 10 we arrive at a switching frequency of $333.3\,\mathrm{Hz}$. As the mechanical time constant is assumed greater because of the rotors, a frequency of $300\,\mathrm{Hz}$ is chosen.

As mentioned earlier the original switching circuitry was switched at $180\,\mathrm{Hz}$, which means that the new chosen frequency is almost twice as large. Theoretically the motor will see a more even signal than at the lower frequency and operate in a more stable mode. However a tradeoff with more heat dissipation in the switch will be apparent. If the switch can withstand this dissipation or not will be revealed in the following test section.

3.5 TEST AND VERICATION OF NEW POWER DRIVE

This section aims to test and verify that the new power drive can function as a reliable and efficient power source to the motors. The following two pictures in Figure 3.2 and Figure 3.3 show the test setup in the laboratory.

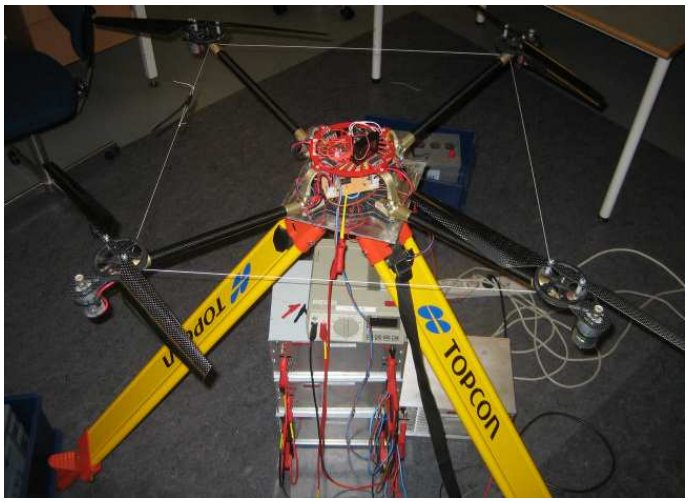

FIGURE 3.2: *Test of new power drive in laboratory.*

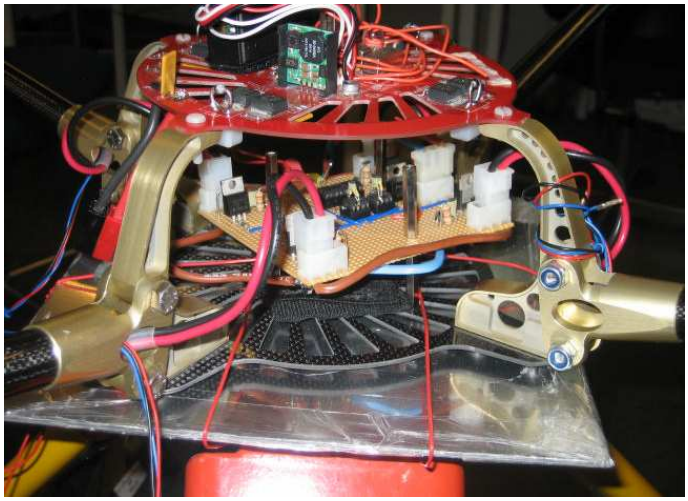

FIGURE 3.3: *Test of new power drive in laboratory.*

Firstly the X-Pro was placed on a three foot shown in the pictures above to minimize the ground effect as much as possible. Furthermore the body of the helicopter was tightened with wire to the three foot so that no matter how much throttle is applied, it would not lift off. The power drive was supplied from four $10\,\mathrm{A}$ power supplies in parralel (see picture) which should provide sufficient current during the test.

The four power drives were fed the same $300\,\mathrm{Hz}$ PWM signal from a signal generator. The duty cycle was set to 1% and power was applied to the power termnials. The duty cycle was over $10\,\mathrm{s}$ increased to 95% and a timer was started at this instant. According to the worksheet **Expected flight time performance**, chapter 1 page 71, the X-Pro has an effective flight time of approximately 6 minuttes. Therefore the timer was stopped after

8 minuttes giving two minuttes of buffering. When the 8 minuttes were reached the duty cycle was lowered to 50% in order to ensure full functionality, and then further again lowered to 1% and turned off.

The main supply was $12\,\mathrm{V}$ and all the current it would possibly want at this voltage. Even though the X-Pro battery is at $14.8\,\mathrm{V}$, this is still seen as sufficient power stress on the switches. After the test the switches were checked for any heat dissipation and there was found almost none. No component in the complete power drive circuitry was found hot or in some other way stressed. This indicates that no heat sinks have to be calculated and mounted on the switches. Through this test, which is believed to be reliable due to the test being performed directly on the X-Pro and not some other stand in, the new power drives are verified to work as planned.

Interface to radio receiver 4

The four channels each representing roll, pitch, throttle, yaw respectively will produce certain signals on the output of the Remote Control (R/C) module. The module has six channels in all, where only the first four channels are described here. Each channel will once every 22.5ms provide a positive pulse. The length of this pulse is proportional to the corresponding remote control stick. The timing of all four channels are shown in Table 4.1.

Channel	Interpret	Min pulselength	Max pulselength	Rest pos.	Min pos.
1	Roll	1.111ms	1.926ms	1.521ms	Left
2	Pitch	1.109ms	1.934ms	1.522ms	Down
3	Throttle	1.104ms	1.925ms	1.104ms	Down
4	Yaw	1.114ms	1.934ms	1.522ms	Left

TABLE 4.1: *Length of positive pulses on each channel. The period of all channels are 22.5ms.*

The frequency of each channel is 44.44Hz, but it is not possible to measure each channel other than to detect the rising edge and afterward time the pulse length. This fact can be depicted in to measurements of the timing of all five channels (the fifth channel is used for mode switching, intended for safety feature) in two different cases, as seen in Figure 4.1 on the facing page.

From the two graphs it can be seen that the channel four and five are pushed in time when the length of channel three is increased. Also notice that the length of the pulse in channel 5 has different characteristics than the four others. For more information see interface between Radio receiver module and Safety interrupt module in Section 4.0.1.

4.0.1 Radio receiver ⇒ Safety interrupt module

Channel 5 is intended to carry the interrupt signal from the R/C. The timing of this channel is shown in Table 4.2.

Channel	Interpret	Min pulse length (down pos.)	Max pulse length (up pos.)
5	Switch	0.970ms	2.074ms

TABLE 4.2: *Length of pulse when switch E on the R/C is down (minimum) and up(maximum). The period of all channels are approximately 22.5ms.*

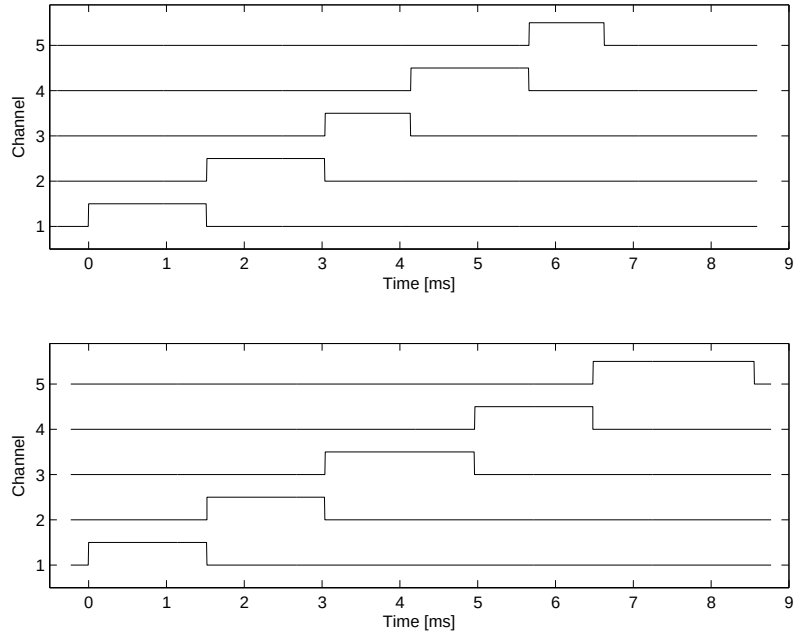

FIGURE 4.1: *Timing of all five channels, in two different cases. The top graph is with min throttle(channel 3) and the switch position down(channel 5). The bottom graph has max throttle and switch position up.*

WORKSHEET V: PRELIMINARY TEST RECORD FOR INVESTIGATING MAGNETIC FIELD IN LIKELY MAGNETOMETER PLACEMENTS ON THE DRAGANFLYER X-PRO (X-PRO) DURING OPERATION

29/9-2006

This test record documents measurements made on the X-Pro regarding the magnetic field measurable during operation. A magnetometer is intended to be used for estimating a part of the orientation of the X-Pro during operation. The orientation is in theory estimated by measuring the lines of magnetic force from the Earth's magnetic field, and filtering the results. This should give at least two out of three degrees of freedom of the X-Pro orientation. It is known from other projects [BVs04, p. 141-145] that these measurements are prone to disturbances from iron objects and electrically generated magnetic fields nearby. The effects of these are unknown, which motivates the investigation of the magnetic field in likely positions of the magnetometer under different circumstances.

CONTENTS OF WORKSHEET V

1 Analysis **43**

1.1 Disturbances . . . 43
1.2 Test equipment . . . 43
1.3 Test cases . . . 44

2 Test results **46**

2.1 Test case 1 - Rotation about ${}^{m}\hat{Z}$ . . . 47
2.2 Test case 2 - Rotation about ${}^{m}\hat{X}$ . . . 48
2.3 Test case 3 - Rotation about ${}^{m}\hat{Y}$ . . . 49
2.4 Test case 4 - Start-up sequence . . . 50
2.5 Test case 5 - Throttle . . . 51
2.6 Test case 6 - Yaw . . . 52
2.7 Test case 7 - Pitch . . . 53
2.8 Test case 8 - Roll . . . 54
2.9 Test case 9 - Iron block . . . 55
2.10 Test case 10 - Yaw 2 . . . 56

3 Conclusion **57**

ANALYSIS 1

1.1 DISTURBANCES

The disturbances, which are considered likely to occur, are:

Permanent magnets The brushed DC-motors contain permanent magnets, which could distort the magnetic field. These are also known as hard iron disturbances [Mic, p. 5].

Metal components Some parts of the X-Pro are made of metal, which could be ferromagnetic, depending on the material used. The components themselves are not magnetized, but will be as long as being exposed to a magnetic field, such as the one generated by the Earth, thereby creating a distortion in the original field. These disturbances are also known as soft iron disturbances [Mic, p. 5].

Solenoids in motors The brushed DC-motors also contain solenoids to convert the supplied current to a local magnetic field, creating the torque to make the motors and rotors turn. This field could be large enough to locally distort the Earth's magnetic field used for orientation determination [BVs04, p. 141-145].

Wires and Field Effect Transistor (FET) The wires and FETs used for supplying the DC-motors is also presumed to cause a disturbance in the magnetic field due to the high currents used [SB00, Ex. 30.4 p. 947].

Other disturbances The tests are conducted in an indoor environment, and there is a risk of both hard iron, soft iron and electrically generated disturbances in such an environment. Since this is a preliminary test, these disturbances are at first considered low enough to be disregarded.

There is also a risk of electrical disturbances being introduced in analogue measurements running in wires from the sensors to the Analog/Digital Converter (ADC).

1.2 TEST EQUIPMENT

The magnetometer used for this setup is a Honeywell HMR3000, a 3-axis magnetometer. The device contains three sensors for measuring the magnetic field, and a tilt sensor based on gravity affecting fluid in a small container. The HMR3000 has an onboard processor, which can use these sensors to estimate the heading, pitch and roll. In this test, only the Raw Compass Data (RCD), which the sensor also outputs, are considered relevant in order to isolate the data on the magnetic field.

1.3 TEST CASES

The magnetometer has been used in different environments, and with several tests conducted in each of these. In all environments, the magnetometer has been oriented in the same initial orientation. The following test cases are regarded necessary in order to investigate the properties of the measurements and the disturbances in the magnetic field.

Free In this environment, the X-Pro is tested free of the X-Pro. These tests are conducted in order to investigate the behavior of a magnetometer in an environment without the disturbances, which might be present on the X-Pro. In order to rule out magnetic field changes due to actual rotation, the X-Pro is tied firmly to a test table during these test cases. Three tests are performed in this environment:

Test case 1 - Rotation about $^m\hat{Z}$: In this test case, the magnetometer is rotated about it's z-axis, which is pointing upwards.

Test case 2 - Rotation about $^m\hat{X}$: Rotation about the magnetometer x-axis.

Test case 3 - Rotation about $^m\hat{Y}$: Rotation about the magnetometer y-axis.

On X-Pro, on iron block This environment is considered the worst case environment, which should be experienced on the X-Pro, when disregarding external disturbances. The magnetometer is placed just below the central control circuit, and just above where the battery is placed. Also, this test setup includes:

- No shielding of cables, FETs, central control circuit or motors.
- No removal of metal components present on the original X-Pro.
- Since the original battery will be exchanged with another in time, which contents are unknown, the magnetometer is placed directly on top of an iron block. This block has the same mass as the expected battery mass, and should represent the worst case distortion expected from a battery.

It is desired to measure the impact on the magnetic field from the X-Pro during normal operation. The current setup is unfortunately not able to make the X-Pro fly, so an impact slightly larger that what is measured will be expected. The reason for this reduced performance is due to either lower battery performance or, more likely, a larger period between pulses in the new Remote Control (R/C) receiver module.

The following test cases are to be conducted:

Test case 4 - Start-up sequence: This test is conducted in order to determine whether the start-up sequence of the X-Pro has any impact. This includes only turning on the R/C, turning on the X-Pro and arming it. No throttle is applied.

Test case 5 - Throttle: This seeks to examine the effects of variuos throttle levels. The test will be conducted with 0%, 25%, 50%, 75% and 100% throttle of what the battery can supply.

Since there is a risk that the magnetic field from the four motors will balance out when supplied with equal current, the following three tests are also conduected:

Test case 6 - Yaw: Using the yaw control at two different throttle levels. This will make two rotors placed opposite of eachother run with increased speed, while the speed of the two others is decreased.

Test case 7 - Pitch: On rotor will be run at a higher speed, while it's opposite is run at a lower speed. The two other rotors maintain their speed.

Test case 8 - Roll: Same as **Test case 7** but with the left and right motors instead of front and back.

Iron block This test case is conducted in order to determine the impact of the iron block.

Test case 9 - Iron block: The magnetometer is placed freely on the test table, outside the X-Pro. While measuring, the ironblock is put directly on top of the magnetometer, and removed after a specified amount of time.

On X-Pro, without iron block This environment is used in order to determine how the magnetic field changes during X-Pro operation without the iron block. This is also made in order to investigate whether the iron block has had a dampening effect on the other changes in the magnetic field.

Test case 10 - Yaw 2: This test case is the same as **Test case 6**, but without the iron block.

TEST RESULTS 2

The test results depicted are the RCD extracted from the magnetometer. Also the Conditioned Compass Data (CCD) and Heading, Pitch and Roll (HPR) are supplied by the magnetometer, but these are filtered and otherwise compensated by the onboard processor with an unknown algorithm. Due to this, these are recorded during the experiments, but disregarded in this journal.

The first three test cases are performed with the magnetometer being handheld, so some uncertainties due to this must be taken into account. Also, the X-Pro is tied firmly to the table in the last seven test cases, but mechanical vibrations might be measurable.

2.1 TEST CASE 1 - ROTATION ABOUT ${}^{m}\hat{Z}$

The results depicted in Figure 2.1 are based on the following test sequence:

$0 - 10s$: Initialization of experiment.
$10 - 20s$: System at rest.
$20 - 30s$: Magnetometer rotated $90°$ Clock-Wise (CW) about the ${}^{m}\hat{Z}$, and stopped.
$30 - 40s$: Magnetometer rotated $90°$ CW about the ${}^{m}\hat{Z}$, and stopped.
$40 - 50s$: Magnetometer rotated $90°$ CW about the ${}^{m}\hat{Z}$, and stopped.
$50 - 60s$: Magnetometer rotated $90°$ CW about the ${}^{m}\hat{Z}$, and stopped - thereby being in its initial position.
$60 - 70s$: System at rest.
$70s$: End of experiment.

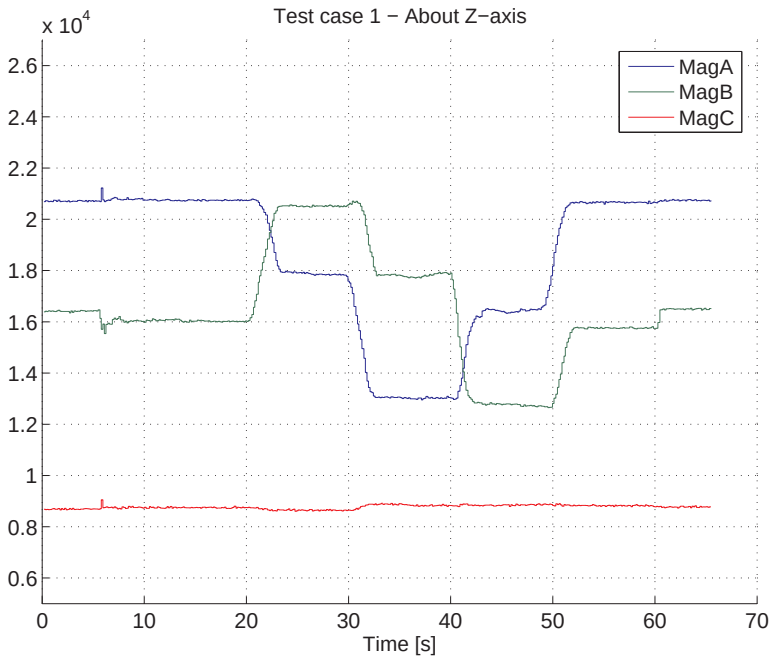

FIGURE 2.1: *Test case 1 - Rotation about ${}^{m}\hat{Z}$.*

2.2 TEST CASE 2 - ROTATION ABOUT ${}^{m}\hat{X}$

The results depicted in Figure 2.2 are based on the following test sequence:

$0 - 10s$: Initialization of experiment.
$10 - 20s$: System at rest.
$20 - 30s$: Magnetometer rotated $90°$ Counter-Clock-Wise (CCW) about the ${}^{m}\hat{X}$, and stopped.
$30 - 40s$: Magnetometer rotated $90°$ CCW about the ${}^{m}\hat{X}$, and stopped.
$40 - 50s$: Magnetometer rotated $90°$ CCW about the ${}^{m}\hat{X}$, and stopped.
$50 - 60s$: Magnetometer rotated $90°$ CCW about the ${}^{m}\hat{X}$, and stopped - thereby being in its initial position.
$60 - 70s$: System at rest.
$70s$: End of experiment.

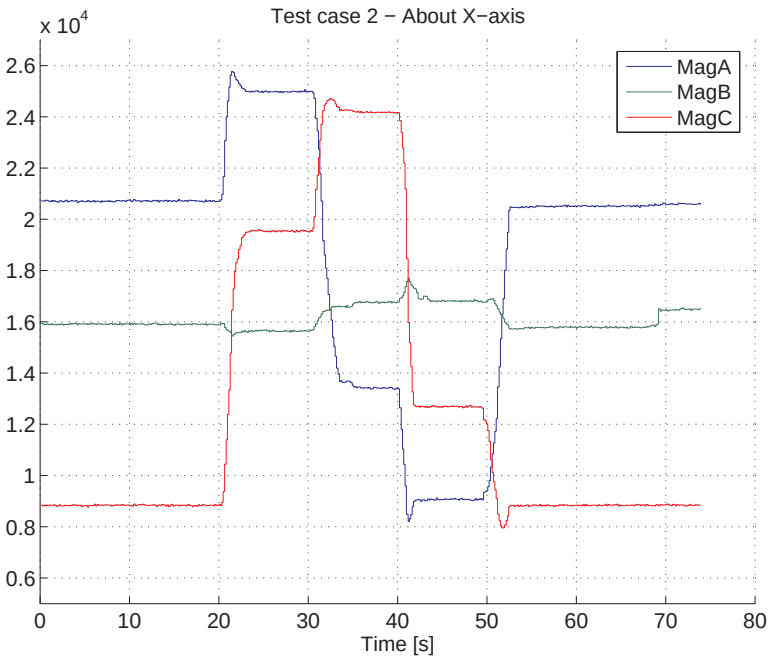

FIGURE 2.2: *Test case 2 - Rotation about* ${}^{m}\hat{X}$.

2.3 TEST CASE 3 - ROTATION ABOUT ${}^{m}\hat{Y}$

The results depicted in Figure 2.3 are based on the following test sequence:

$0 - 10s$: Initialization of experiment.
$10 - 20s$: System at rest.
$20 - 30s$: Magnetometer rotated $90°$ CCW about the ${}^{m}\hat{Y}$, and stopped.
$30 - 40s$: Magnetometer rotated $90°$ CCW about the ${}^{m}\hat{Y}$, and stopped.
$40 - 50s$: Magnetometer rotated $90°$ CCW about the ${}^{m}\hat{Y}$, and stopped.
$50 - 60s$: Magnetometer rotated $90°$ CCW about the ${}^{m}\hat{Y}$, and stopped - thereby being in its initial position.
$60 - 70s$: System at rest.
$70s$: End of experiment.

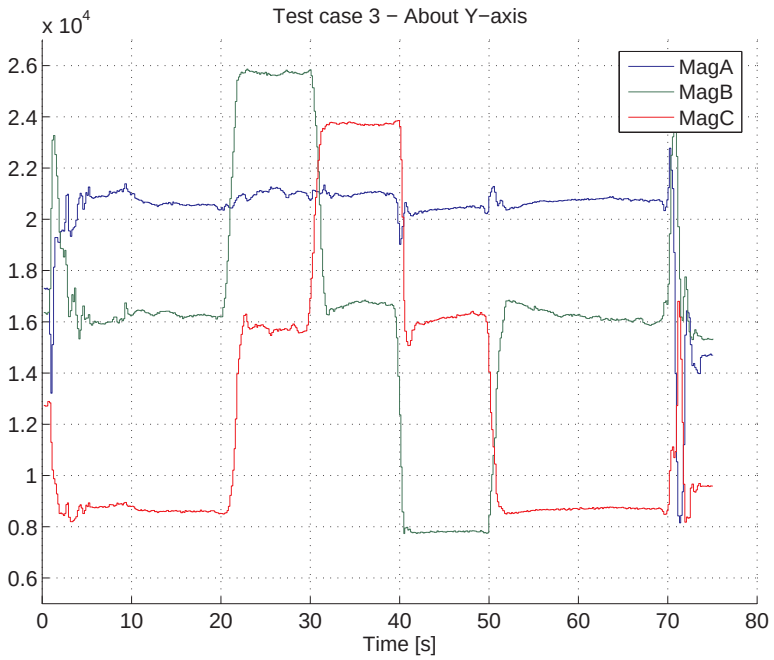

FIGURE 2.3: *Test case 3 - Rotation about* ${}^{m}\hat{Y}$.

2.4 TEST CASE 4 - START-UP SEQUENCE

The results depicted in Figure 2.4 are based on the following test sequence:

$0 - 10s$**:** Initialization of experiment.
$10 - 20s$**:** System at rest.
$20 - 30s$**:** R/C turned on.
$30 - 40s$**:** X-Pro turned on.
$40 - 50s$**:** X-Pro armed.
$50 - 60s$**:** System at rest.
$60 - 70s$**:** X-Pro off.
$70 - 80s$**:** R/C off.
$80 - 90s$**:** System at rest.
$90s$**:** End of experiment.

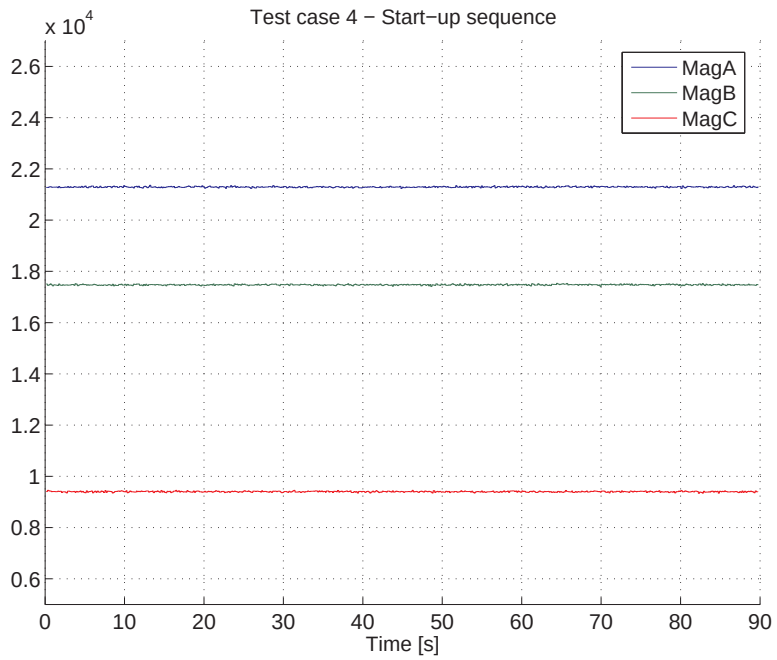

FIGURE 2.4: *Test case 4 - Start-up sequence.*

2.5 TEST CASE 5 - THROTTLE

The results depicted in Figure 2.5 are based on the following test sequence:

$0 - 10s$: Initialization of experiment.
$10 - 20s$: System at rest.
$20 - 30s$: 25% throttle.
$30 - 40s$: 50% throttle.
$40 - 50s$: 75% throttle.
$50 - 60s$: 100% throttle.
$60 - 70s$: 75% throttle.
$70 - 80s$: 50% throttle.
$80 - 90s$: 25% throttle.
$90 - 95s$: System at rest.
$95s$: End of experiment.

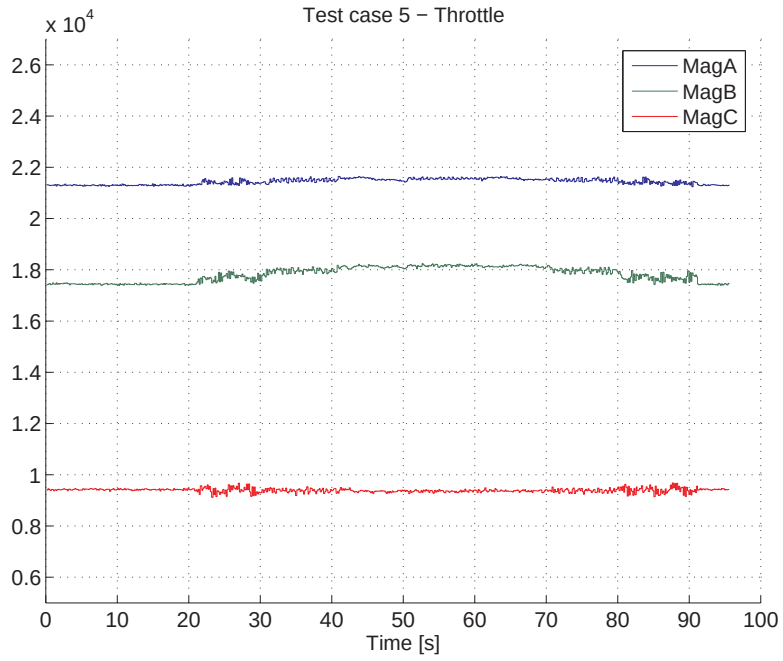

FIGURE 2.5: *Test case 5 - Throttle.*

2.6 TEST CASE 6 - YAW

The results depicted in Figure 2.6 are based on the following test sequence:

$0 - 10s$: Initialization of experiment.
$10 - 20s$: System at rest.
$20 - 30s$: 50% throttle.
$30 - 40s$: Full yaw left.
$40 - 50s$: Full yaw right.
$50 - 60s$: 50% throttle.
$60 - 70s$: 100% throttle.
$70 - 80s$: Full yaw left.
$80 - 90s$: Full yaw right.
$90 - 100s$: 100% throttle.
$100 - 110s$: System at rest.
$110s$: End of experiment.

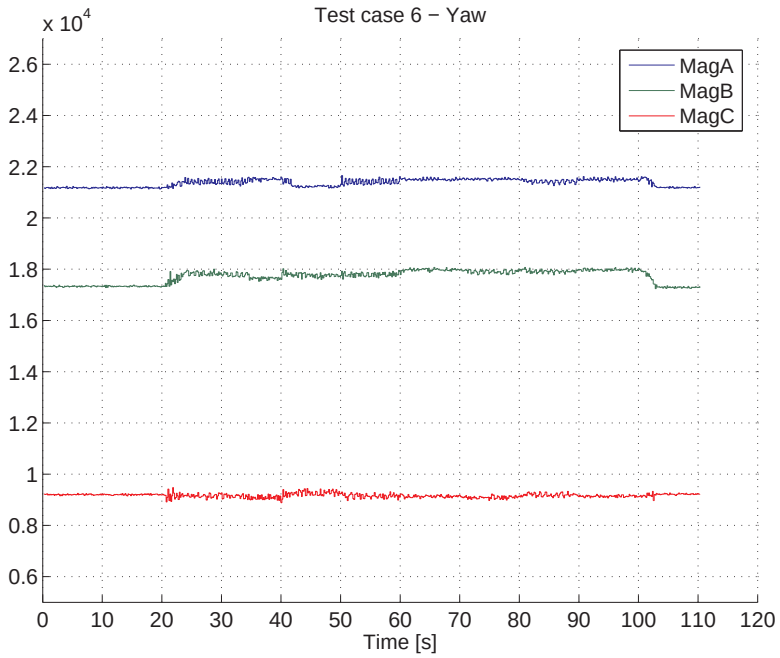

FIGURE 2.6: *Test case 6 - Yaw.*

2.7 TEST CASE 7 - PITCH

The results depicted in Figure 2.7 are based on the following test sequence:

$0 - 10s$: Initialization of experiment.
$10 - 20s$: System at rest.
$20 - 30s$: 50% throttle.
$30 - 40s$: Full pitch forward.
$40 - 50s$: Full pitch backward.
$50 - 60s$: 50% throttle.
$60 - 70s$: 100% throttle.
$70 - 80s$: Full pitch forward.
$80 - 90s$: Full pitch backward.
$90 - 100s$: 100% throttle.
$100 - 110s$: System at rest.
$110s$: End of experiment.

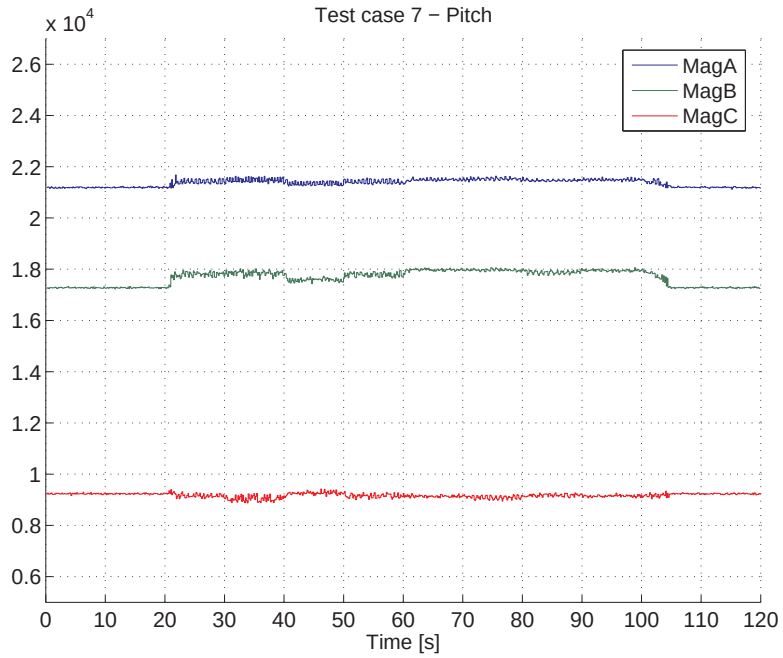

FIGURE 2.7: *Test case 7 - Yaw.*

2.8 TEST CASE 8 - ROLL

The results depicted in Figure 2.8 are based on the following test sequence:

$0 - 10s$: Initialization of experiment.
$10 - 20s$: System at rest.
$20 - 30s$: 50% throttle.
$30 - 40s$: Full roll left.
$40 - 50s$: Full roll right.
$50 - 60s$: 50% throttle.
$60 - 70s$: 100% throttle.
$70 - 80s$: Full roll left.
$80 - 90s$: Full roll right.
$90 - 100s$: 100% throttle.
$100 - 110s$: System at rest.
$110s$: End of experiment.

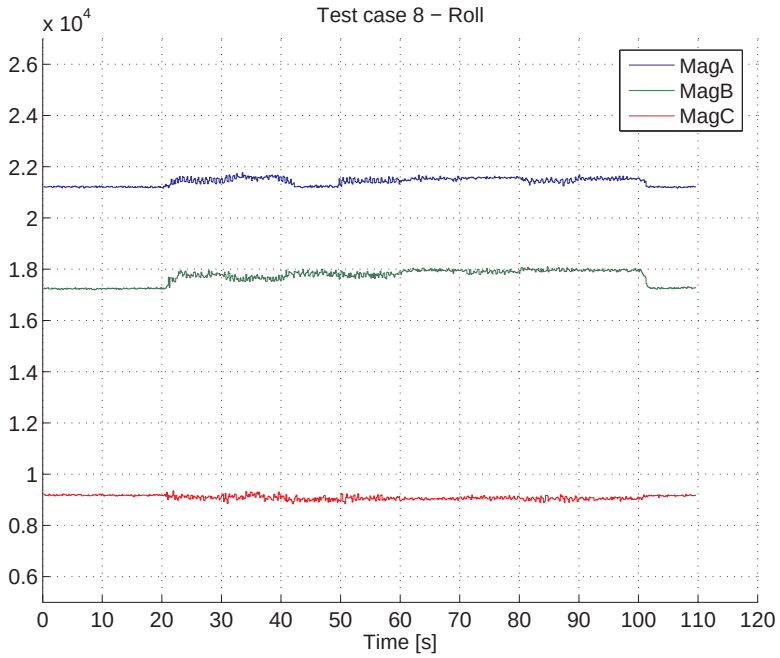

FIGURE 2.8: *Test case 8 - Roll.*

2.9 TEST CASE 9 - IRON BLOCK

The results depicted in Figure 2.9 are based on the following test sequence:

$0 - 10s$**:** Initialization of experiment.
$10 - 20s$**:** Iron block moved on top of magnetometer.
$20 - 30s$**:** System at rest.
$30 - 40s$**:** Iron block removed from magnetometer.
$40 - 50s$**:** System at rest.
$50s$**:** End of experiment.

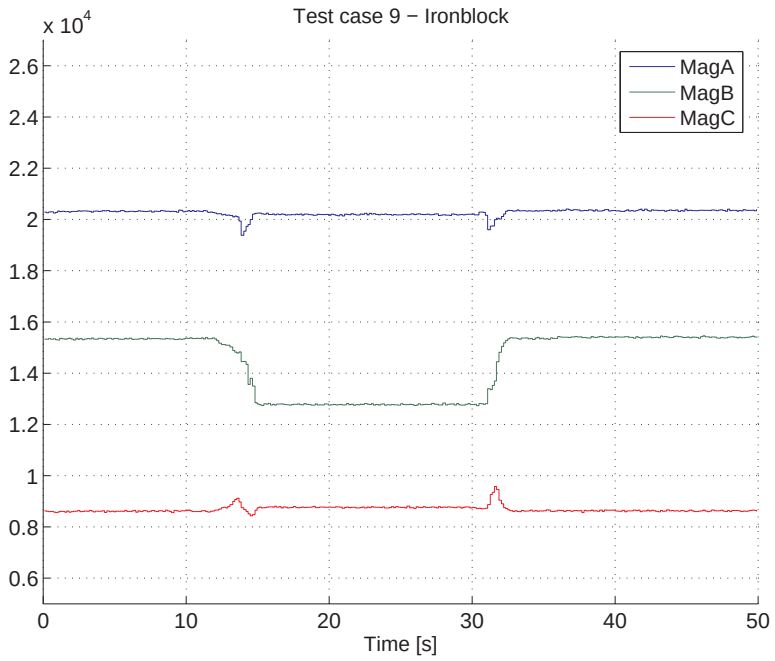

FIGURE 2.9: *Test case 9 - Iron block.*

2.10 TEST CASE 10 - YAW 2

The results depicted in Figure 2.10 are based on the following test sequence:

$0 - 10s$**:** Initialization of experiment.
$10 - 20s$**:** System at rest.
$20 - 30s$**:** 50% throttle.
$30 - 40s$**:** Full yaw left.
$40 - 50s$**:** Full yaw right.
$50 - 60s$**:** 50% throttle.
$60 - 70s$**:** 100% throttle.
$70 - 80s$**:** Full yaw left.
$80 - 90s$**:** Full yaw right.
$90 - 100s$**:** 100% throttle.
$100 - 110s$**:** System at rest.
$110s$**:** End of experiment.

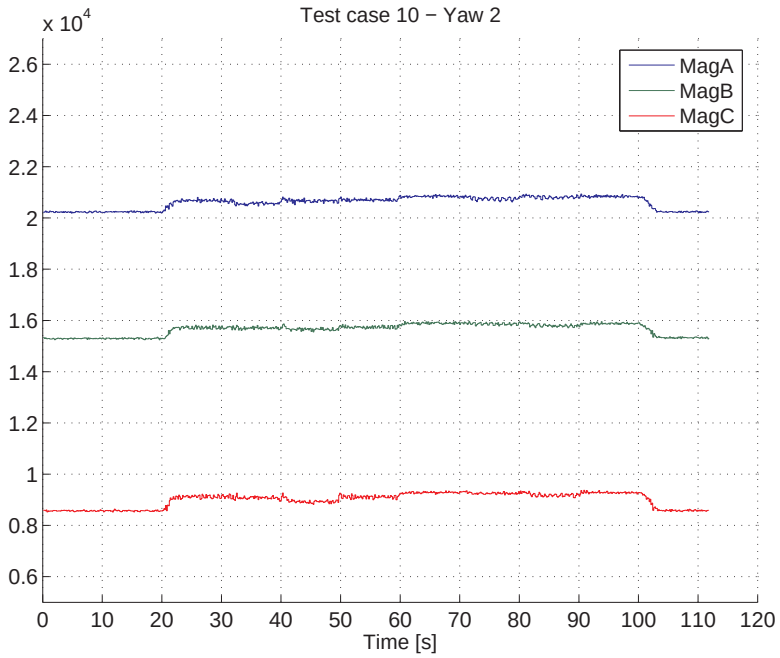

FIGURE 2.10: *Test case 10 - Yaw 2.*

CONCLUSION 3

Apparently, the magnetic field generated by the X-Pro itself does not seem to be a disturbance of such a magnitude, that the magnetometer is rendered useless. As can been seen on Figures 2.1 to 2.3, the change of signal is of a larger magnitude when the magnetometer is rotated, than the change of signal during i.e. change of throttle, see Figure 2.5. It appears in this figure, that increased throttle leads to noise with a mean value determined by some function of the throttle. The variance of the noise in the same figure also seems to follow some function of the supplied throttle, with a maximum variance at about 25% throttle. The introduction of an iron disturbance seems to give a steady rotation of the magnetic field, and it is expected, that such a disturbance could be compensated for, since the X-Pro has no moving iron objects near the magnetometer during operation.

The following measures could be applied in order to improve the data for determination of the X-Pro orientation:

- Shield the power electronics, such as motor power drives, and wires with high or fluctuating currents.
- Place the magnetometer in a location remote from iron and power electronics.
- Twist the wires supplying the motors.
- Reroute the power distribution to the motor power drives. As it is now, the power for the motor power drives is distributed through a cobber ring around the magnetometer. It is presumed that such a ring with high currents works as a solenoid, and creates a magnetic field due to this. Rerouting the distribution of power to the motor power drives could prehaps reduce the magnetic field disturbance.
- Lowpass filter the output of the motor power drives at their end of the wire. Variance in the magnetic field noise might be reduced. It is expected, that this step would reduce fluctuations in the wires from the lowpass filter to the motors.
- Reduce vibrations in the X-Pro. During the experiment, the X-Pro was tied to the test table, but it is possible, that some of the variance in the magnetic field noise seen in i.e. Figure 2.5 is due to an actual change in orientation. Especially at 25% throttle, the variance is high, and this could be due to the rotors rotating at a vibrational eigenfrequency at that specific throttle level.

According to [Hon, p. 7], the magnetic field of Earth is about $0.65\ gauss$ ($65\ \mu T$), so a magnetometer range of $\pm 1\ gauss$ seems to be sufficient.

WORKSHEET VI: FIRST CONTACT WITH THE SRF08 ULTRA SONIC RANGE FINDER.

16/10-2006

This worksheet describes the first attempt in the Draganflyer X-Pro (X-Pro) project to interface with the SRF08 Ultra sonic range finder[Roba]. An Inter-Integrated Circuit (I^2C) interface has been developed on an MSP-TS430PM64 evaluation board containing an MSP430F149, and using IAR Embedded Workbench. The MSP430F149 does not itself contain a hardware I^2C module, so one has been developed in software based on a description of the I^2C protocol [Wol] and on example code [Robb] from the supplier.

CONTENTS OF WORKSHEET VI

1 The sensor **59**

2 Interface **60**

2.1 Media, physical layer 60

2.2 Data Link Layer 60

2.2.1 Writing to slave 61

2.2.2 Receiving from slave 62

2.3 Presentation Layer 63

2.3.1 Ranging 64

2.3.2 Changing maximum range 64

2.3.3 Changing analogue gain 64

2.3.4 Using Artificial Neural Network mode 64

THE SENSOR 1

The SOund Detection And Ranging (SODAR) sensor is an SRF08 Ultrasonic range finder, equipped with two ultrasonic transducers, one light sensor, and an onboard processor. The sensor has the following characteristics:

- Supply voltage: 5v regulated.
- Max. sample period: 65ms by default (equivalent to 11m). The sample period can be configured down to a sample period equivalent to 43mm.
- Range: $\approx 3\text{cm} - 6\text{m}$. Max. range can be configured to be shorter, if a shorter sample period is required.
- Configurable analogue gain, except when in Artificial Neural Network mode.
- Interface: I^2C. [Robb] recommends an SCL-clock at or lower than 100kHz. It should be noted, that the sensor uses clock-stretching.

INTERFACE 2

The sensor can be assigned to one of sixteen adresses on the I^{2}C bus. This allows up to sixteen instances of this model to be connected to the same I^{2}C bus, along with other devices, as long as these are not assigned to the same address span. At power-up, the current address is announced on a red Light Emitting Diode (LED) mounted on the sensor - default address is `E0`.

To sufficiently describe the interface used in the test-setup, the interface is divided into layers somewhat similar to the OSI model. It should be noted, that the I^{2}C bus is a master-slave topology (with the option of multiple masters though), and the SRF08 is from the manufacturer configured to be a slave.

2.1 MEDIA, PHYSICAL LAYER

The SRF08 has the two connections used in the I^{2}C bus, used for SDA (Serial DAta line) and SCL (Serial CLock line). According to the I^{2}C standard, these are to be connected with pull-up resistors, since I^{2}C compatible devices are only allowed to pull the connections down, see Figure 2.1.

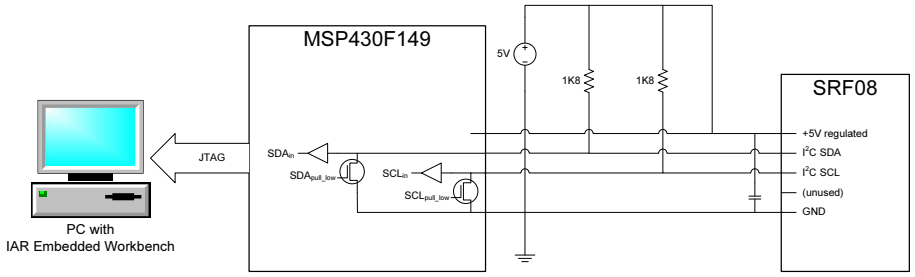

FIGURE 2.1: *Interface between SODAR and the MSP430F149.*

The physical layer consists of the hardware depicted in Figure 2.1 and software to control the hardware. From software, it is possible to pull either or both connections SDA and SCL down, and to determine the logical level of the connections.

2.2 DATA LINK LAYER

The data link layer handles the control of the bit sequences sent and received on both connections via the physical layer. Certain sequences are used when communicating with a slave - these are denoted Start, Address, Ack, Data and Stop. Due to this, it can be advantageous to divide the Data Link Layer into two sublayers: One low-level sublayer

handling these sequences individually, and one high-level sublayer controlling the order of the sequences when sending or receiving data from a slave. On some devices, an I^2C driver is already implemented in both hard- and software, and as long as these fulfill the requirements from the SRF08, the following subsection can be disregarded.

Each slave has a unique address, which can be reconfigured on the SRF08. This is only necessary once, since this information is stored in non-volatile memory. Please note, that the address is the only information stored in non-volatile memory, and all other configuration options are restored to factory defaults at power-up. In practice, the I^2C device is accessed using two adresses: One for writing (even address) and one for reading (odd address). The default address for writing to the SRF08 is `E0` and thereby reading is done through `E1`.

2.2.1 Writing to slave

An example of a master writing a sequence to a slave can be seen on Figure 2.2.

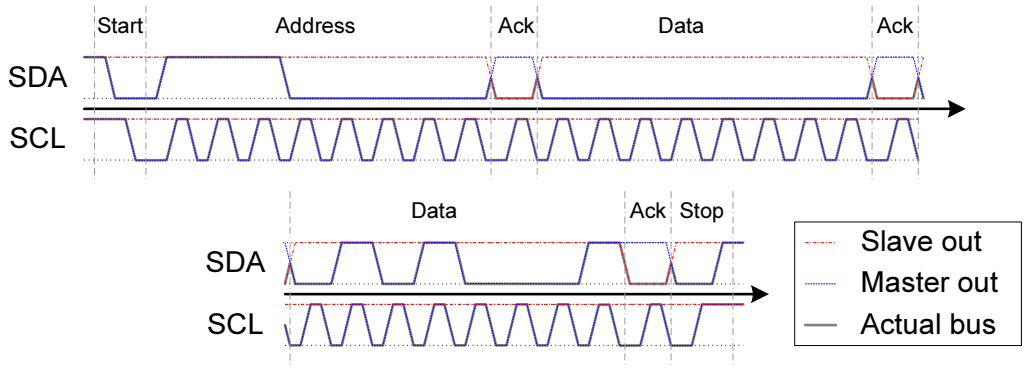

FIGURE 2.2: *The data sequence for writing `E0 00 51` to a slave.*

The writing sequence is as follows:

1. Initially, both SDA and SCL are logically high; that is, released by both master and slave.

2. Start: The master pulls SDA low first, then SCL low.

3. Address for writing (even): The address is written with MSB first. The master releases SDA if the current bit is 1, and pulls it low if it is 0. When the SDA is set to the correct value, the master releases the SCL to indicate that the SDA can be read by the slaves. After a short amount of time, the master pulls SCL low again, and the procedure in this item is repeated for the remaining bits.

4. Receiving Ack: After sending either an Address or Data byte, the master must wait for an acknowledge from the slave. As soon as the master pulls SCL low after sending the last bit and afterwards releases SDA, the SDA will be pulled down by the slave. The master can then release SCL and pull it down again to, after which the slave will release SDA, and the communication can continue.

   If Ack is not received by the master, the slave is either not there, or missed some of the bits. If either is the case, the master should issue a Stop sequence.

5. Sending Data: Sending a databyte is the same procedure as sending an Address. Both Address and Data should be followed by an Ack. This step is repeated until all databytes are sent.

6. Stop: When transmission is done, the master issues a stop command by first releasing SCL and afterwards releases SDA.

2.2.2 Receiving from slave

The procedure for reading a byte is the following:

1. Start is initiated by master.
2. Address for writing (even) is sent to slave.
3. Ack is received from slave.
4. Databyte is sent to slave, indicating what register is to be read.
5. Ack from slave.
6. Start sent again from master. This is allowed, and is also referred to as a restart command.
7. Address for reading (odd) is sent to slave.
8. Ack from slave.
9. Databyte is sent from slave. This is done as follows:
   (a) Master releases SDA, and pulls SCL low, waiting.
   (b) Slave pulls SDA low or releases, depending on first MSB bit to be sent. Optionally, the slave can also pull SCL low, and SRF08 uses this option. When the SDA has been set, it must release SCL.

(c) Master releases SCL to indicate, that it reads SDA. If implementet correctly, the master should first check whether SCL is actually released - if not, it is because it is still pulled low by the slave. This is called clock-stretching, and is a way for the slave to indicate, that SDA has not been set correctly yet, because the slave cannot keep up with the speed. The master must wait for reading the SDA until SCL has been released by both master and slave, see Figure 2.3.

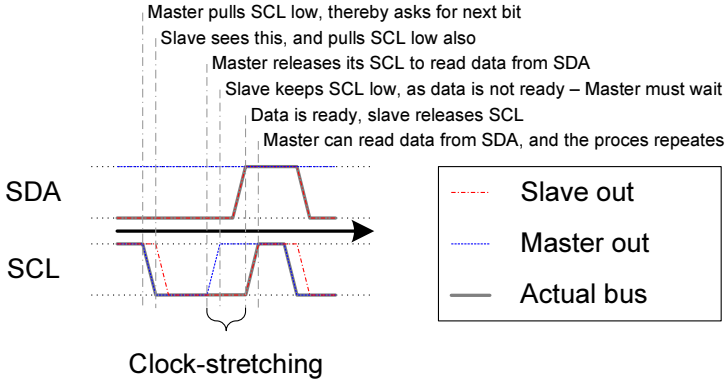

FIGURE 2.3: *The principle in clock-stretching.*

(d) When the master has read SDA, it pulls down SCL, and the sequence can be repeated for the remaining bits.

Upon receiving a whole byte, the master should issue an Ack to the slave, indicating that the next byte can be sent. The master sends the Ack by pulling SDA low and releasing SCL. Shortly after, it pulls SCL down and releases SDA. Ack is not necessary when the last byte has been received, as the next command is a Stop. The master decides how many bytes, it wants to receive.

10. Stop is sent by master.

2.3 PRESENTATION LAYER

So far, only a protocol stack for sending and receiving bytes between one master and possibly several slaves has been presented. This subsection describes how to interpret these byte sequences. The SRF08 supports many commands, so this will only outline those considered most relevant.

2.3.1 Ranging

When a ranging is required, the following sequence must take place:

1. Master initiates a ranging by sending (all in hex):

   ```
   E0 00 51
   ```

   In this case, device `E0` is ordered to start ranging, and return the results in centimeters. It is also possible to broadcast this message to all SRF08 devices on the bus, but this has not been tested.

2. The slave `E0` starts ranging, which takes a while - 65ms as default, depending on configuration.

3. The master then asks for the result by:

   ```
   E0 01 E1 ?? ?? ??
   ```

   That is, the master writes, that it wants to start reading from register `01`, and then establishes a read-connection to the device. The following three read bytes 3 x `??` then represent lightsensor-reading, ranging high byte and ranging low byte respectively. The lightsensor-reading is between 0 and 255, representing complete darkness and bright light respectively.

   If the master asks too soon, the SRF08 returns `FF`, since it does not respond to the I^2C bus while ranging.

2.3.2 Changing maximum range

(This has not yet been explored, documentation goes here if conducted.)

2.3.3 Changing analogue gain

(This has not yet been explored, documentation goes here if conducted.)

2.3.4 Using Artificial Neural Network mode

(It is uncertain whether this is useful to the project, so it has not been explored yet)

WORKSHEET VII: ECONOMICAL BUDGET FOR THE DRAGANFLYER X-PRO (X-PRO)

29/9-2006

This document describes the desired components for the X-Pro, which are to be purchased.

CONTENTS OF WORKSHEET VII

1 Economical budget **66**

ECONOMICAL BUDGET

1

	Product	Price	P.
Onboard computer package	gumstix™ Connex 400xm	129 *USD*	√
	robostix™	49 *USD*	√
	WiFistix	79 *USD*	√
	Tweener	20 *USD*	√
	Serial null-modem cable	12 *USD*	√
	$5.0v$ Euro Power Adapter	10 *USD*	√
	$2 \times$ Screws & Spacers kit	2×4 *USD*	√
	Shipment	$\approx 2 \times 57.53$ *USD*	√
	Subtotal (USD)	$\approx 422,06$ *USD*	–
	Subtotal (DKR)	$\approx$ **2490, 15 DKR**	–
Power-supply (shipment individual)	Main-battery LiPo	2150 *DKR*	√
	Charger (Ultramat 14)	650 *DKR*	√
	Lithium Balancing Adapter (LBA 10 Net 6s+)	650 *DKR*	√
	Parts for electronics PSU	$\approx$ 80 *USD*	(√)
	Subtotal (DKR)	$\approx$ **3922, 00 DKR**	–
R/C reconfiguration	New crystal, antenna, receiver	1778 *DKR*	√
	Subtotal	$\approx$ **1778, 00 DKR**	–
Sensors (shipment individual)	GPS	1200 *DKR*	√
	IMU	760 *USD*	√
	MicroMag 3-Axis Magnetometer	54.95 *USD*	√
	Ultrasonic Ranger	61.99 *CAD*	(√)
	Subtotal	$\approx$ **6570, 28 DKR**	–
Total		$\approx$ **14760, 43 DKR**	–

TABLE 1.1: *The budget estimated for equipping the X-Pro for autonomous flight. The exchange rates used are* 590 *for USD and* 526 *for CAD.* ***P.*** *is short for* ***Purchased****. Prices are without tax.*

Worksheet VIII: Analysis of the power consumption by the additional electronics added to the Draganflyer X-Pro

16/10-06

Contents of Worksheet VIII

1 Power consumption **68**

1.1 Introduction 68

1.2 Specifications 68

POWER CONSUMPTION 1

This worksheet gives an overview of the power consumption of the additional sensors that have been mounted on the Draganflyer X-Pro (X-Pro).

1.1 INTRODUCTION

This worksheet describes the power consumption of the chosen sensors and additional electronics that are made on the X-Pro to give an estimate if an additional battery should be purchased.

1.2 SPECIFICATIONS

Product	**Supply voltage**	**Max. supply voltage**	**Typical current**	**Max. current during operation**	**Startup current**
gumstix™[1]	-	-	$< 250\ mA$	-	-
robostix™	-	-	-	-	-
WiFistix	-	-	-	-	-
GPS[2]	$[2.7\ V;3.3\ V]$	$3.6\ V$	$56\ mA$	-	$125\ mA$
IMU	$[4.74\ V;5.25\ V]$	-	$89\ mA$	-	-
Magnetometer[3]	$3\ V$	$5.25\ V$	$0.4\ mA$	$0.5\ mA$	-
Ultrasonic ranger[4]	$5\ V$	-	$11\ mA$	$40\ mA$	$275\ mA$
R/C receiver	-	-	-	-	-
Powerdrive	-	-	-	-	-

TABLE 1.1: *Overview of the power consumption but the sensors. All additional circuits that are needed to interface to each sensor or circuit should be included in the calculations.*

WORKSHEET IX: * WEIGHT AND FLIGHT TIME

29/9-2006

CONTENTS OF WORKSHEET IX

1 Expected flight time performance **71**

1.1 Total weight estimate 71

1.2 Thrust needed in hover 72

1.3 Hover time 73

2 Considerations on optimal construction **74**

2.1 Weight 74

The associated m-file is called: **x_pro_weight.m**

EXPECTED FLIGHT TIME PERFORMANCE 1

Changes in battery capacity and weight and the mounting of additional sensors and On Board Computer (OBC) result in a drastic change in general performance of the Dragan-flyer X-Pro (X-Pro). Issues concerning weight and flight time will be dealt with in this worksheet. Lastly a small discussion on the optimality of the construction seeks to set the general mood concerning added weight on the final construction.

1.1 TOTAL WEIGHT ESTIMATE

New components on the X-Pro has been added but not all new components are chosen yet thus some weights needs to be estimated. In the following table all component weights are listed.

Component	est/meas	Weight [g]
Recycle: includes 4 $\times$ rotor arms, 4 $\times$ rotor speed sensor module, bottom main board and anchoring mountings for the rotor arms, new screws.	M	1606
Battery	M	488
Radio receiver	M	29
OBC (Gumstix and robostix) and wifi module	M	56
SODAR $\times$ 4	M	40
GPS	M	28
IMU	E	40
Magnetometer	E	15
Top mount board including power supply, power drives for DC motors, safety switch circuit and wiring.	E	120
Total Weight	E	2408
+ 5%	E	2528

TABLE 1.1: *Weight of all components on the X-Pro, some are estimated since they are not yet at hand.*

Since the weight is a estimate an additional 5% is added to the final weight, in order to give some slag to the calculations. In this case the weight is estimated to $2.528\,\text{kg}$ which will be the weight that is used in the following.

1.2 THRUST NEEDED IN HOVER

Since the weight before the start of this project was lower than it is now the thrust needed to hover is increased. No measurements where taken on the X-Pro that can provide the knowledge about the real connection between power to the motors and the produced thrust. This measurement was though made during a project on Massachusetts Institute of Technology (MIT) denoted as the The Phaeton Project [The03]. Plotting the thrust as a function of the power drawn by the motors is reproduced in Figure 1.1

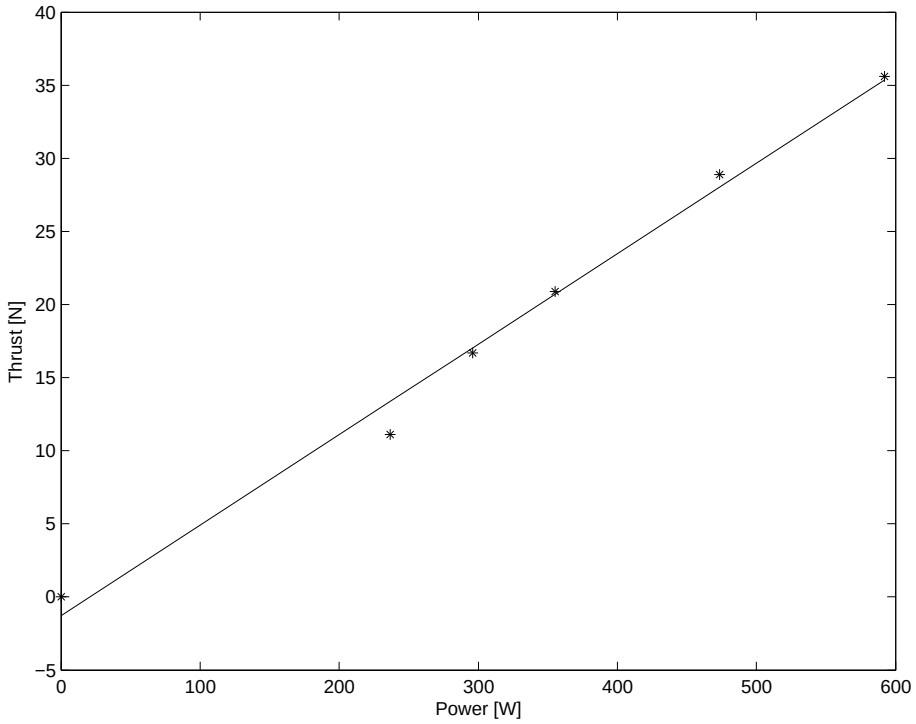

FIGURE 1.1: *Measurements taken on a simalar project. Data combines the power and thrust, and it can be shown that it is reasonable to assume a linearity between these two parameters.*

From the figure it is clearly seen that there is a almost linear relationship between the thrust and the power. This result is used on the one measurement that is taken on the X-Pro in order to give a estimate on the power needed in a hover mode.

The old weight was $2379\,\mathrm{g}$ and the power consumption was $10.8\,\mathrm{V} \cdot 9.2\,\mathrm{A} \cdot 4 \approx 397\,\mathrm{W}$. Using the linearity the new power needed given the new weight is $\frac{397\,\mathrm{W}}{2379\,\mathrm{g}} \cdot 2528\,\mathrm{g} \approx 422\,\mathrm{W}$. It is estimated that the power drive module has an efficiency of $95\,\%$ (based on estimate by the group) which will add to total powerconsumption. Furthermore is it estimated that sensors and OBC will not use more than $1\,\mathrm{A} \cdot 5\,\mathrm{V} = 5\,\mathrm{W}$. Knowing the new power consumption ables the the calculation of the possible hover time.

1.3 HOVER TIME

The new battery that have been bought is capable of producing $6\,\text{Ah}$ and will provide up to $108\,\text{A}$ in burst. The total energy in the battery is known and the power needed to hover is estimated, thus a hover time can be calculated.

$$T_{maxf} = \frac{6\,\text{Ah} \cdot 3600\,\text{s} \cdot 14.8\,\text{V}}{448\,\text{W}} \approx 713\,\text{s} \tag{1.1}$$

The nature of the Lithium-Polymer (Li-Po) is that if it is drained of all its power, the battery will be destroyed. An example is that, if 95% of the power is used before recharging the lifetime of the battery will last about 50 cycles, if only 50% is used the number of cycles rises to about 200 [Han06]. So in order to reduse the wear of the battery, the calculations on flight time is done in the 50% manner. This means that the overall flight time can not be expected to be more than.

$$T_f = \frac{713\,\text{s} \cdot 100}{60\,\text{s} \cdot 50} \approx 5\,\text{min}56\,\text{s} \tag{1.2}$$

It is not possible to drasticly change the weight of the X-Pro or to change the battery capabilities, the flight time most be obeyed at all times.

CONSIDERATIONS ON OPTIMAL CONSTRUCTION 2

2.1 WEIGHT

From the data sheet of the motor a set of optimal working conditions are given. These figures are rewritten here but they can all be found in the data sheet [mot]. The data of interest is put in Table 2.1.

Motor speed $[rpm]$	Current $[A]$	Torque $[mN \cdot m]$	Output $[W]$	η $[\%]$
20510	7.65	31.1	66.7	73

TABLE 2.1: *Withdrawal from data sheet concerning the maximum efficiency point.*

Measurements were taken on the previous (lighter) configuration in hover. The data for this experiment is given in short in Table 2.2.

Motor speed $[rpm]$	Current $[A]$	Voltage $[V]$	Power [W]	Prior weight $[g]$
≈14000	9.2	10.8	99.36	2379

TABLE 2.2: *Test made on hover in prior configuration. Current, voltage and power is for one motor.*

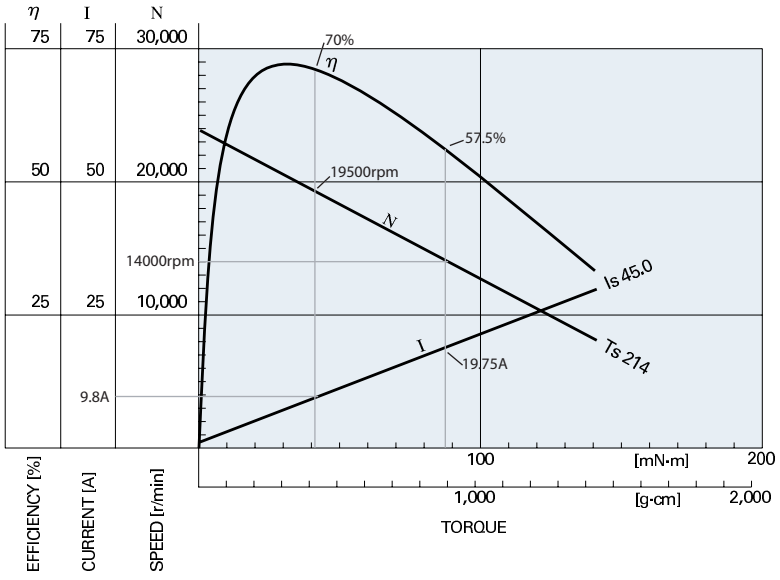

FIGURE 2.1: *Graph showing efficiency current and speed as a function of torque.*

One important point can be drawn from these data. Since the motor has its optimal

performance in one point, and in this point the output is $66.7\,\mathrm{W}$ and the efficiency is $73\,\%$ this means that the motor will be outside the optimal point if one moter uses more than $\frac{66.7 \cdot 100}{73} \approx 91.4\,\mathrm{W}$. The former configuration used $99\,\mathrm{W}$ which already was surpassing the optimal point. With the new configuration is the power consumption estimated to be $^{422}/_{4} \approx 106\,\mathrm{W}$, which is eaven further away from the optimal point.

It is difficult to say what the efficiency of the motor is in hover with the data that is gathered now. One of the blurry topics is concerning the efficiency vs. speed. and current consumption. A short outline of the problem is given in the following but the problem is not investigated further since the work is not considered to contribute to the main task.

The parameters for a hover, was taken in a prior project, see Table 2.2 on the preceding page. In short is the problem that these measurements does not coincide with the graph from the datasheet, see Figure 2.1 on the facing page. In hover the angular velocity of the motor was $14000\,\mathrm{rpm}$ but the current consumption was only $9.8\,\mathrm{A}$. These two points should match to the same torque, but depending on the entry the efficiency can be read to be either $57.5\,\%$ or $70\,\%$. Although this posses a problem the effect is assumed little since the new weigth of the X-Pro differs relatively little from the original weight.

Some more topics could be investigated if time allows.
Problems:

- Weight up => rotor speed up => motor speed up => less optimal motor speed
- Weight up => motor less efficient => more heat in motor => More wear of the motor => shorter fly time
- Power for hover is over the efficiency point of the 4 motors => Others motors are needed

WORKSHEET X: * GPS MODULE FOR THE DRAGANFLYER X-PRO

8/11-2006

A description of how to interface the GPS module to the gumstix and how to relay Radio Technical Commission for Maritime services (RTCM) correction packages for Differential Global Positioning System (DGPS) augmentation.

CONTENTS OF WORKSHEET X

1 Introduction **77**

2 Hardware setup **78**

3 Serial communication **79**

3.1 Protocol specification 79

3.2 Module configuration 80

4 Navigation **83**

4.0.1 DGPS 83

4.1 Extraction of the position estimate 86

4.1.1 Data flow 87

4.1.2 Variance of position estimate 88

4.2 Implementation documentation 88

4.2.1 Design flowchart 89

4.2.2 synchronization on receiving GPS packets 89

4.2.3 Function call listing 89

INTRODUCTION 1

In order for the Draganflyer X-Pro (X-Pro) to obtain an absolute position, a Global Positioning System (GPS) module must be incorporated. A SAM-LS from Ublox, [Ublb], has been chosen due to a number of reasons. First of all the module is very small and weigh only 23 g. This is important as the GPS module along with numerous sensors and other devices must be placed within a relatively small area. Furthermore it had the best performance, down to three meters Spherical Error Probability (SEP), in its weight class when investigating the GPS market. Other relevant features:

- Integrated antenna
- Low power consumption
- Fully Electro Magnetic Interference (EMI) shielded
- 16 channel receiver
- 4 Hz position update rate

The product specific data sheet of the module can be found at via the producers homepage [Ublc]. Chapter 3 in the extended data sheet, [Ubld] goes more into detail with some aspects of the module. Along with the SAM-LS module an interfacing cable aid, Smart Adaptor Board (SAB), was bought [Ubla]. Finally the protocol specifications for the package in and out flow can be found here [Uble].

The remaining worksheet will first describe the hardware interfacing of the GPS module to the gumstix via the serial STUART connection. Secondly the protocol, package components etc. will be dealt with. After that the navigation aspects will be covered, which packages to use for positioning and also how to relay RTCM correction packages to the GPS module.

HARDWARE SETUP 2

The GPS module has two serial ports, Serial Port 1 is configured to 9600 Baud and Serial Port 2 is configured to 57600 Baud. Both ports are configured to 8 bits, no parity bits, 1 stop bit and no flow control. The serial input ports accept the Ublox Binary Protocol (UBX), National Marine Electronics Association (NMEA) and Radio Technical Commission for Maritime services protocol without explicit configuration and are activated upon startup. The output of Serial Port 1 is configured to transmit both UBX and NMEA, but only NMEA is activated at startup. The Serial Port 2 is configured in the same way, but opposed to Port 1 the UBX is activated upon startup. Both Serial Port Input/Output (I/O)s are however configurable.

Serial Port 2 with the highest speed is chosen along with the Ublox specific UBX protocol which supports a greater amount of adjustable features. Furthermore it has less overhead due to the fact that the NMEA protocol is ASCII based and not binary like the UBX.

The following electrical diagram in Figure 2.1 shows the GPS module hardware connection.

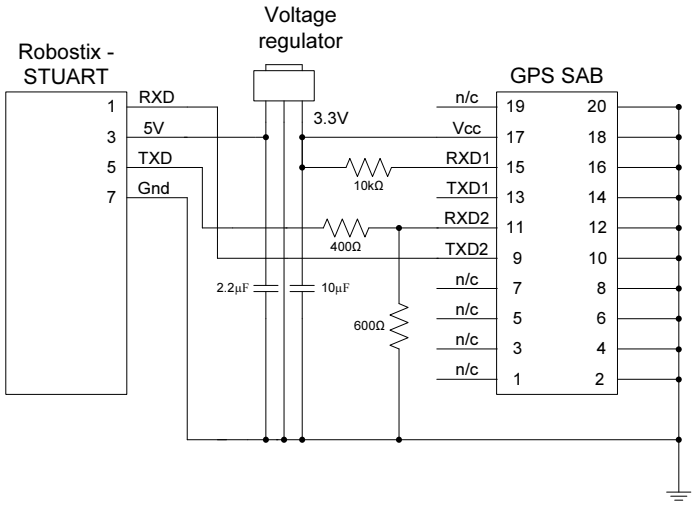

FIGURE 2.1: *Interface between GPS module and gumstix. Notice that there is a fault in the SAB data sheet, seriel port 2 TX and RX are flipped.*

As can be seen from the Figure the `RXD` and `TXD` from the STUART is connected to the `TXD2` and `RXD2` on the robostix respectively. The GPS module is supplied from the robostix through a $5\,\mathrm{V} \rightarrow 3.3\,\mathrm{V}$ voltage regulator. Two capacitors are regarded as sufficient for decoupling and one pull-up resistor is used on `RXD1` to avoid floating inputs. The `TXD` signal from the robostix is voltage divided to suit the $2.7\,\mathrm{V} - 3.3\,\mathrm{V}$ GPS range.

SERIAL COMMUNICATION 3

As mentioned earlier a Ublox proprietary protocol is used to transmit data to and from the gumstix module. This chapter handles the UBX protocol specification and how to configure the GPS module.

3.1 PROTOCOL SPECIFICATION

The key features of the protocol are the following, [Uble]:

- It is compact, 8 bit binary data is used.
- Checksum protected, using a low overhead checksum algorithm.
- Modular, using a 2-stage message identifier (Class- and message ID).

The UBX package is configured as illustrated in the following Figure 3.1.

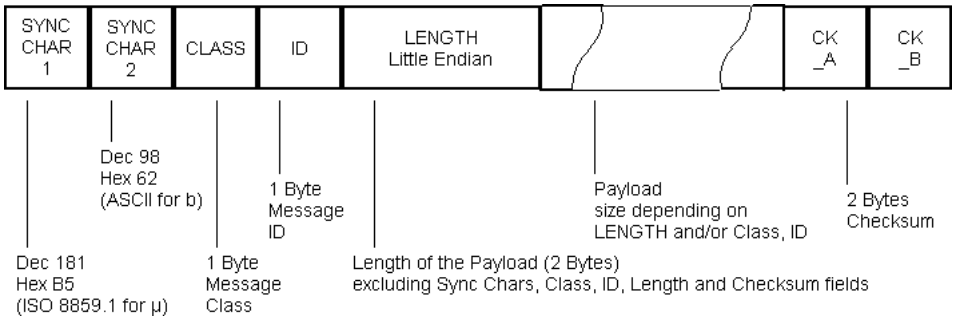

FIGURE 3.1: *UBX protocol, [Uble].*

Every message starts with two synchronizing bytes, namely `0xB5` and `0x62`. One byte follows containing the class which defines the basic subset of the message. Another byte ID defines more specifically what is actually contained in the package. The length part consists of two bytes which represents the length of the payload only. The payload is a variable length field. The message ends with a 2 byte checksum `CK_A` and `CK_B` whose calculation is defined below.

```
CK_A = 0, CK_B = 0
For(I=0;I<N;I++){
  CK_A= CK_A + Buffer[I]
  CK_B= CK_B + CK_A
}
```

The checksum is an 8 bit Fletcher algorithm calculated over the package, beginning at the class byte and ending just after the payload. The Buffer contains the data over which the checksum is calculated. After the loop, the two bytes, CK_A and CK_B are transmitted at the end of the package.

3.2 MODULE CONFIGURATION

The GPS module must be configured in accordance with the specific goals for the project. As mentioned earlier it is possible to configure everything from the serial port setup to what packages to receive from the module, either by polling or periodic output and guide the way the position is calculated. These various options can be found in the protocol specification [Uble].

Before any configuration packages will be received correctly by the module, a version package of the MON-VER type must be transmitted. This verfifyes the sender so no configuration is done unintended. This specific package is shown below in hexadecimal form.

```
MON-VER - 0A 04 46 00 35 2E 30 30 20 20 20 20 4A 61 6E 20 30 39 20
32 30 30 36 20 31 32 3A 30 30 3A 30 30 00 01 30 30 30 30 30 30 34
30 00 00 54 4C 4C 31 2E 32 20 20 4A 61 6E 20 30 39 20 32 30 30 36
20 31 35 3A 34 33 3A 34 32 00 00
```

where the MON-VER relates to first two bytes of the package, that is 0A and 04 respectively. The two sync chars are always assumed to be placed in front of the shown code and likewise the checksum is omitted. After this transmission, various CFG- messages can be transmitted and are either acknowledged or not acknowledged by the module. Only CFG- messages are acknowledged or not acknowledged.

In order to receive the correct packages from the module and mask other unwanted packages out, the CFG-MSG can be invoked. An example of a message configuration is shown below

```
CFG-MSG - 06 01 06 00 01 02 00 00 01 00
```

The first two bytes indicate that it is a CFG-MSG message (06 and 01). The next two bytes indicate the length of the payload which in this case is 6 bytes. The 01 and 02 is the message class and message identifier respectively that the configuration is concerning. The four last bytes indicate which serial port to send the message to. The 01 contained in the second last byte indicates that the package is sent to I/O target 2 which is Serial Port 2.

The following package is able to get or set the navigation engine.

```
CFG-NAV2 - 06 1A 28 00 01 00 00 00 03 03 10 02 50 C3 00 00 18 14 05
3C 00 01 00 00 FA 00 FA 00 64 00 2C 01 00 00 00 00 00 00 00 00 00
00 00 00
```

The immediate interesting byte in this package is byte five (01) which determines what dynamic platform model to run with. The available models are:

- 0x01 Stationary
- 0x02 Pedestrian
- 0x03 Automotive
- 0x04 Sea
- 0x05 Airborne with < 1g acceleration
- 0x06 Airborne with < 2g acceleration
- 0x07 Airborne with < 4g acceleration
- 0x00 and 0x08-0xFF is reserved

The model will be set to airborne with < 1 g when put in the X-Pro.

The rate of which the packages are transmitted from the module can be configured by the following package.

```
CFG-RATE - 06 08 06 00 FA 00 01 00 00 00
```

The immediate byte of interest is again here the fifth byte (FA) which indicates 250 ms and is the maximum output of the GPS module. Only the messages that are maped by

CFG-MSG will be send periodic. It is worth mentioning that any messages can be polled from the gps module at any time disregarding of the rate setting.

The configuration file can be found on the CDROM at this location `gps/ad_hoc_conf.txt` and is fully compatible with the configuration file that can be obtained and sent through Ucenter. Ucenter is a free GPS evaluation software made available from [Ublb].

NAVIGATION 4

The gps module is more or less a stand alone sensor, understood in the way that all the calculations that needs to be done in order to calculate an absolute position is carried out inside the GPS module. This is also the case concerning the DGPS signal.

4.0.1 DGPS

The most important blocks in the GPS system are the GPS satellites. They transmit a signal to the earth, but before reaching the earth the signal passes through several layers in the atmosphere doing this induces a deflection of the signal. This nature will show as a bias on the position estimate. To accommodate this problem the DGPS technology is conceived. In short the technique is using one stationary GPS receiver with a known position. By subtraction it is possible to get a correcting signal which will be valid for all gps receivers nearby.

This of course posses the problem of relay of data from the stationary GPS receiver to the gps module mounted on the X-Pro.

The Ublox GPS module purchased is compatible with the RTCM SC-104 data link protocol. Since the communication to the X-Pro is wireless 802.11g standard, it would be possible to use a protocol that can relay RTCM SC-104 packets over the Internet. Such a protocol exist and is called Networked Transport of RTCM via Internet Protocol (Ntrip). The structure is divided into four different element types:

- NtripSource that produces the DGPS signal.
- NtripServer that sends the RTCM packets over the internet
- NtripCaster that obtains the DGPS signal from the NtripServer and makes it possible to make http requests on the RTCM SC-104 packets
- NtripClient is the element that obtains the RTCM signal from the NtripCaster, the communication between NtripCaster and NtripClient is fully compatible with HTTP 1.1.

The Danish firm GPSnet.dk is providing the DGPS signal using the Ntrip protocol, thus only a NtripClient is to be set up by the group to correct the GPS signal. GPSnet.dk has a evenly distributed network of NtripSources through out Denmark as can be seen on Figure 4.1.

The nearest Sources are in Hals and Suldrup which both are approximately 25 km away from campus. At the moment only the transmitter in Hals are active thus this is

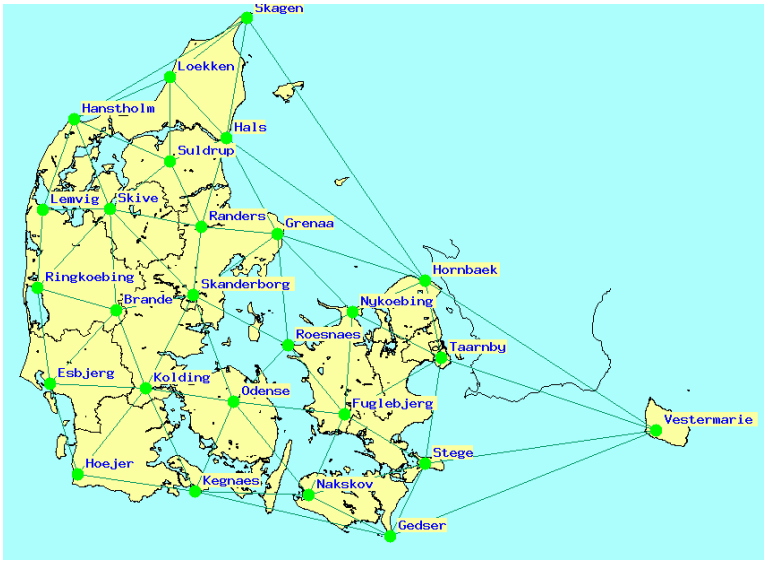

FIGURE 4.1: *Locations of all NtripSources that are owned by GPSnet.dk.*

the preferred node. It is important that stationary GPS module sees the same satellites as our moving GPS module does, this can only be insured by having the stationary gps module close to the moving one. If the solution should be optimal we should set up a DGPS signal our selfs, but this solution is assumed to be too time consuming.

Interfacing to the RTCM signal is a somewhat simple since the GPS module supports the the signal transmitted by GPSnet.dk. There is although some buts, the internal network at Aalborg university is protected by a http proxy which has posed some problems for the obtained sample code from the founders of the Ntrip scheme (`http://igs.ifag.de/index_ntrip.htm`). The solution for this problem has been to make a forward of the external signal according to Table 4.1.

Adr.	`Makalu.GPSnet.dk`	⇔	`www.control.aau.dk`
Port	9000	⇔	9090

TABLE 4.1: *The forwarding of RTCM signal*

The license for the DGPS signal from GPSnet.dk is provided by Peter Cederholm at Department of Development and Planning. The agreement is that the signal can be used for non commercial use until the end of the project in June 2007. In practice the password and user name is used to encode the Ntrip packets and thus is requiring some computation. Furthermore there is a security issue if the wireless router with a lower security level than the rest of the network, and be a backdoor to get inside the the AAU-network. The solution to both problems is to let the Development Host Machine (DHM) handle

the encoding and let it relay the the pure RTCM signal to the gumstix^{TM}via a socket. This means that the main computation is handled on a fast machine and will therefor introduce a smaller delay compared to a solution where the Ntrip signal is feed to the gumstix^{TM}. Also there is no direct connection from the internal wireless network to the outside world.

Most of the setup is captured in Figure 4.2

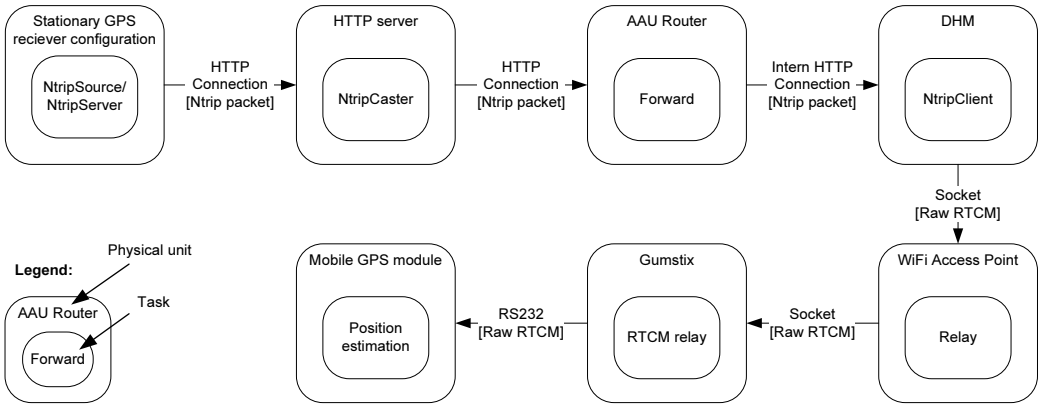

FIGURE 4.2: *The flow of data from the DGPS source to the user of the differential signal.*

Delay of DGPS signal

As can be seen from Figure 4.2 is the communication channel relatively long which poses the problem of a unnecessary delay of the DGPS signal. Studies on this problem has been conducted by others and show that the delay can be as much as 4 s, but this surely depend on the number of hubs between source and the destination and the congestion at the given time [Pet04], [PSV04]. Test of the delay on the signal used has not been conducted but it is expected that the delay is no longer than 1.5 s. Papers on the subject does not turn down this time delay thus it is expected that the signal will improve the position. The gps module it self does a integrity check on the signal based on a time stamp origination from inside the inside the RTCM packet. The DGPS signal is discarded if the signal is too old. If the software have knowledge on that the DGPS signal is lost, it is recommended to configure the GPS module to estimate the position on pure GPS signals.

4.1 EXTRACTION OF THE POSITION ESTIMATE

After the module is configured and the DGPS signal is enabled (if available), the position acquisition mode can be started. In this mode the favorable setup is to get a position estimate as often as possible which is every $1/4$ s, the setup is enabling the periodic transmission of the position estimate at this rate.

Although the protocol linked to the GPS module lists many packets that can be sent,the number of necessary packets are relatively few. The Packet Named NAV-SOL (Navigation Solution Information) contains the result of the calculations carried out inside the GPS module. Following here is a listing of the content of this packet:

Byte offset	Number format	Name	Unit	Comment
0	U4	ITOW	ms	GPS Millisecond Time of Week
4	I4	Frac	ns	Nanoseconds remainder of rounded ms above, range -500000 .. 500000
8	I2	week	-	GPS week (GPS time)
10	U1	GPSfix	-	GPSfix Type, range 0..4 0x00 = No Fix 0x01 = Dead Reckoning only 0x02 = 2D-Fix 0x03 = 3D-Fix 0x04 = GPS + dead reckoning combined 0x05 = Time only fix 0x06..0xff: reserved
11	U1	Flags	-	0x01=GPSfixOK (i.e. within DOP & ACC Masks) 0x02=DiffSoln (is DGPS used) 0x04=WKNSET (is Week Number valid) 0x08=TOWSET (is Time of Week valid) 0x09..0xFF: reserved
12	I4	ECEF_X	cm	ECEF X coordinate
16	I4	ECEF_Y	cm	ECEF Y coordinate
20	I4	ECEF_Z	cm	ECEF Z coordinate
24	U4	Pacc	cm	3D Position Accuracy Estimate
28	I4	ECEFVX	cm/s	ECEF X velocity
32	I4	ECEFVY	cm/s	ECEF Y velocity
36	I4	ECEFVZ	cm/s	ECEF Z velocity
40	U4	Sacc	cm/s	Speed Accuracy Estimate
44	U2	PDOP	-	Position DOP
46	U1	res1	-	reserved
47	U1	numSV	-	Number of SVs used in Nav Solution
48	U4	res2	-	reserved

TABLE 4.2: *NAV-SOL packet. The left most column is the offset inside the payload. 2nd column states format that is either integer (I) or unsigned integer (U) and the number following indicates the size in bytes.*

The Table 4.2 shows the content of the most important packet with respect to the position estimate. Starting from the top of the table is the first 10 bytes representing a time stamp, this time is set in Greenwich Mean Time (GMT) stating Monday as the first day of the week. GPSfix will state the state the module are able to give a position in. Under normal condition this signal will be "3D-fix"(0x03), if only few satellites (3-4) are available, or if configured to it, the gps module can give the position in 2D coordinates, this can though be disabled so only "3D-fix" or "No fix" is the possible states. If the gps module only sees 0-2 satellites the result will be "No fix" which invalidates the following position and velocity. The "Flags" is to be read bitwise and contain the validity of DGPS, time , and the position estimate. Bytes 12-23 and bytes 28-39 are the position and velocity given in the Earth Centered Earth Frame (ECEF), which also means that a position is fixed to the location on the Earth. "Pacc" and "Sacc" are the gps modules estimate of the accuracy, the calculation of these figures are hidden for us but the figures are correlated with the variance of the position and velocity respectively. Further investigation on these figures can be made in order to estimate the variance on the GPS position estimate. This variance can even be made dynamic, because of the online calculation of accuracy. The Position Dilution Of Precision (PDOP) is a geometric measure of the quality of the positioning due to triangularization. The number is calculated as the reciprocal value of the largest volume of a pyramid that can be drawn from the gps module to any four satellites that is seen by the gps module. The optimal value will be 1, and under normal conditions tests show numbers from 1,6-3 which both are assumed to be a good satellite constellation [fGR98]. Finally the last element of relevance is the number the number of satellites used in navigation. This number should at least be three to have a position fix, but test have shown that under normal conditions this figure is about 6-8 satellites.

If it should be wanted to weight the individual axes of the position the packet NAV-DOP could be obtained, in this packet more elaborate figures are given on the precision of the position, eg. in the north/south direction or in the horizontal level.

Amongst other packets that is it would be favorable to use information from. The NAV-DGPS packet has information on the the size of the correction on each satellite at the current time. This data might be useful of off line evaluation of the GPS signal, and through that test the effect of the DGPS feature.

4.1.1 Data flow

In the minimum configuration is the only packet of interest the NAV-SOL packet, thus the contribution to the data flow from the gps module is a simple calculation. It is possible

to get the position 4 times a second and each packet is 52 bytes of payload and 8 bytes of overhead for checksum, length indication, class and id identification and sync characters. In the minimum case the data amount is:

$$(52 + 8) \cdot 4 = 240^{B}/_{s} = 1920^{b}/_{s} \tag{4.1}$$

Should other packets be enabled for periodic transmission will the data amount of cause go up. Following is a list of packets of interest and there sizes.

Packet name	Usage	Size
NAV-SOL	Position and velocity estimate and global time	60 bytes
NAV-DGPS	Information on the correction which is put on the individual satellites. Is dependent on the number of satellites tracked by the gps moudule.	$24 + NCH \cdot 12$ bytes
NAV-DOP	Measures on the quality of the position on the estimate.	26 bytes
NAV-TIMEGPS	Small packet containing the global time obtained from the satellites, could be used in a time sync routine.	24 bytes

TABLE 4.3: *Sizes of different packets of interest.*

Further more the bit rate of the DGPS signal is $1500\,\mathrm{b/s}$ including the overhead. Combined with the other sensor inputs it should be possible to estimate the bandwidth needed to log all information of the running system.

4.1.2 Variance of position estimate

This work has not been conducted, but as previously described will the "Pacc" entry in the NAV-SOL packet be correlated with the variance if a online estimate of the variance on the position estimate. It will also be possible to estimate a static variance, with the help of windows software provided by the manufacturer. This should be done in the case of pure GPS and in DGPS aided measurements.

4.2 IMPLEMENTATION DOCUMENTATION

Much of the implementation is already described in previous sections, and this section is meant to gather the threads to give a more homogeneous overview. Besides this some of the essentials building blocks will be described.

4.2.1 Design flowchart

The steps taken to obtain the GPS packets on the gumstix^{TM}is done according to Figure 4.3.

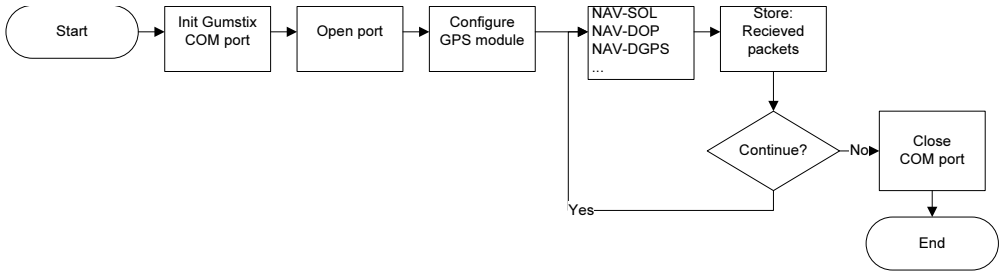

FIGURE 4.3: *The routine for starting the GPS driver all of the blocks are implemented except for the storing of packets. Since the dicision on where to put them has not been made.*

The initialisation/setup of connection is a line of sequential task and after the setup the packets of interest is arriving on the COM port in a periodic fashion. On a time scale the arrival of packets can be visualized as in Figure 4.4 on the following page.

The position estimate is arriving four times a second but the DGPS signal only is send back ones a second. The RTCM and the GPS packets are not synchronized in the manner that they wait on each other. If the DGPS signal should drop out, the GPS module will be discard the newest DGPS signal in the case that it is rated too old.

4.2.2 synchronization on receiving GPS packets

The ublox protocol [Uble] dictates that every packet has 2 sync bytes, as described in Section 3.1. These sync character are $0xB5$ and $0x62$. The synchronization has been implemented in accordance with the Finite State Automata (FSA) shown in Figure 4.5 on the next page.

The only two times the synchronization can be lost is in a startup phase and in the case of lossy connection. In the sync state the following id, class length, payload and checksum can be received.

4.2.3 Function call listing

Clear interface specification is missing on the topic. When the interface to the Gumstix is given, it should be written here.

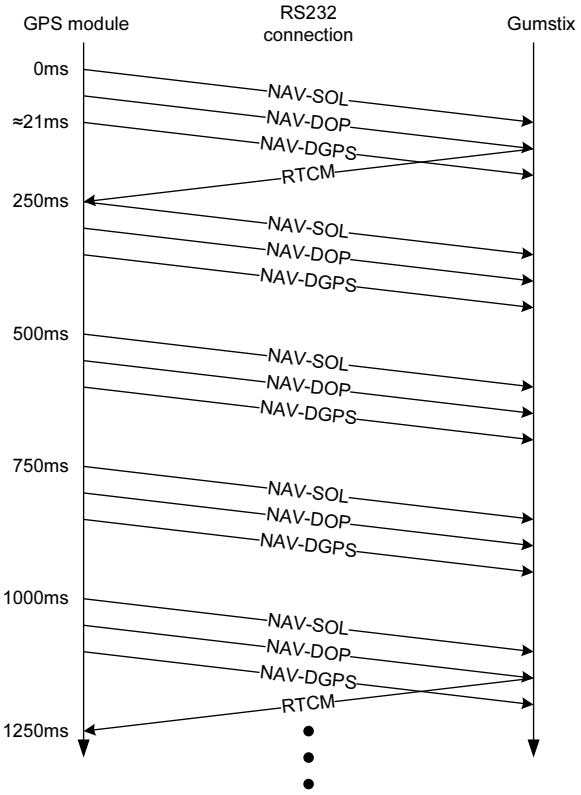

FIGURE 4.4: *The timescale on the figure is not shown linearly since the bursting of GPS packets are only active* $\approx 10\%$ *of the time.*

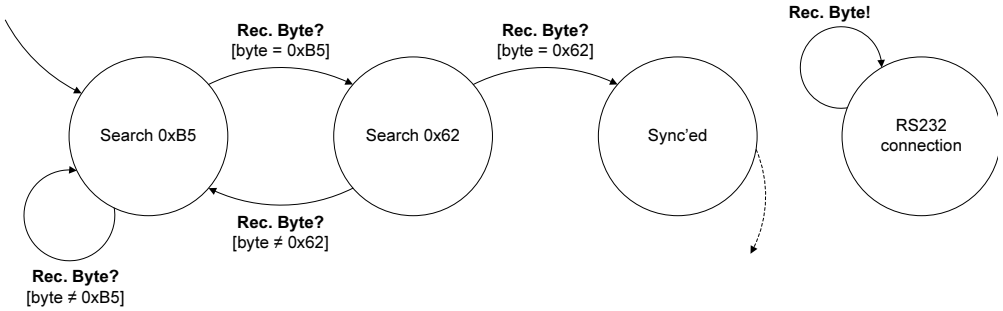

FIGURE 4.5: *The synchronization in receiving packets from the GPS module.*

WORKSHEET XI: * MAGNETOMETER

23/10-2006

This worksheet introduces the chosen MicroMag3 magnetometer and reviews its interfacing procedure used to gain the first contact across its interface.

CONTENTS OF WORKSHEET XI

1 Sensor specification **92**

2 Interface **93**

2.1 Physical layer / Media . . . 93

2.1.1 SPI Port Lines . . . 93

2.1.2 Hardware Handshaking Lines . . . 94

2.1.3 Pull-up resistor . . . 94

2.1.4 Decoupling capacitor . . . 94

2.2 Datalink layer . . . 94

2.3 Presentation layer . . . 95

2.3.1 Command byte . . . 96

2.3.2 Results . . . 97

3 First contact **98**

SENSOR SPECIFICATION 1

The MicroMag3 is a 3-axis magnetic field sensing module. It uses Magneto-Inductive sensors that change inductance by 100% over its field measurement range. The module is constructed with a Serial Peripheral Interface (SPI).

- Low power: draws < 500 mA at 3 VDC
- Small size: 25.4 x 25.4 x 19 mm
- Large field measurement range: ± 1100 mT(± 11 Gauss)
- High resolution field measurement: 0.015 mT(0.00015 Gauss)
- Fast sample rate: up to 2000 samples/second
- Operation: 3.0 VDC
- Fully digital interface: SPI protocol at 3 V

INTERFACE 2

SPI bus is a 4-wire serial communications, master/slave interface. Whenever two devices communicate, one is referred to as the "master" and the other as the "slave" device. The master provides and drives the serial clock signals and activates the "slave" it wants to communicate with. The synchronous clock shifts serial data into and out of the microcontrollers in blocks of 8 bits and the data is simultaneously transmitted and received, making it a full-duplexed protocol.

2.1 PHYSICAL LAYER / MEDIA

The test setup was conducted where a "MCU" was a "master" device, while the magnetometer was the "slave" device. The magnetometer is a SPI device and has a standard connectors and configuration, as illustrated in Figure 2.1.

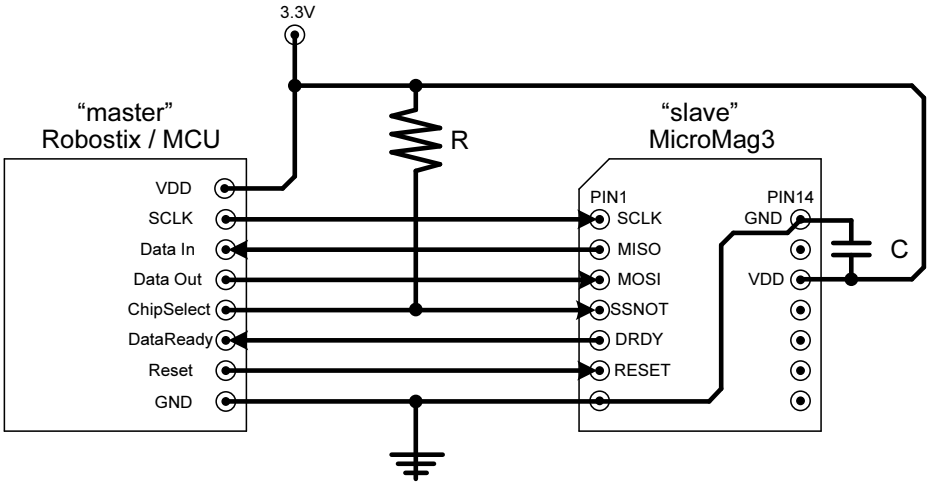

FIGURE 2.1: *Connection setup between "master" and "slave".*

The connections on the device can be divided in two groups, (1) SPI Port Lines and (2) Hardware Handshake Lines.

2.1.1 SPI Port Lines

The SPI configuration requires two control lines (SSNOT and SCLK) and two data lines MOSI (Master-Out-Slave-In) and MISO (Master-In-Slave-Out).
The MASTER device provides the clock signal (SCLK) and determines the state of the slave-select line (SSNOT), i.e. it activates the SLAVE it wants to communicate with when signal on line is low. SSNOT and SCKL are therefore outputs. The SLAVE device receives

the clock (SCLK) and slave-select (SSNOT) from the MASTER, SSNOT and SCKL are therefore inputs.

2.1.2 Hardware Handshaking Lines

There are two handshaking signal lines RESET and DRDY (Data Ready). The RESET signal is usually low, but is toggled low-high-low at the start of data transmission. The DRDY signal is set low after a RESET and it is set high after a command has been received and the data from the MicroMag3 is ready. The purpose of the DRDY signal is to ensure that data is clocked out of the MicroMag3, but only when available.

2.1.3 Pull-up resistor

Wire (SSNOT) is supposed to be logically high when the "slave" is not selected and is therefore pulled up with the Vcc-connector. This is useful when the system powers up, and the master's SSNOT port is input by default. Without the pull-up resistor, the slave could think it was selected until the master reconfigures its SSNOT port to be output and logically high.

2.1.4 Decoupling capacitor

Placing a capacitor very close to the IC power and ground pin connections takes RF energy generated by rapid changes of current demand on the power supply during switching, and then channels it to the ground return path. This prevents power line channeled noise from subverting normal circuit operation. Uncontrolled power supply noise has many effects on a digital system. Those problems include intermodulation and crosstalk. [DB]

2.2 DATALINK LAYER

This layer describes the bit-sequence for transmission between the master and the slave device. The bit-sequence is depicted in Figure 2.2.

The sequence is as follows:

1. Initially, `SSNOT` is logically high, `SCLK` is low, and `RESET` is low.
2. The master selects the slave setting `SSNOT` low.
3. A reset pulse on `RESET` must be sent before sending a command.

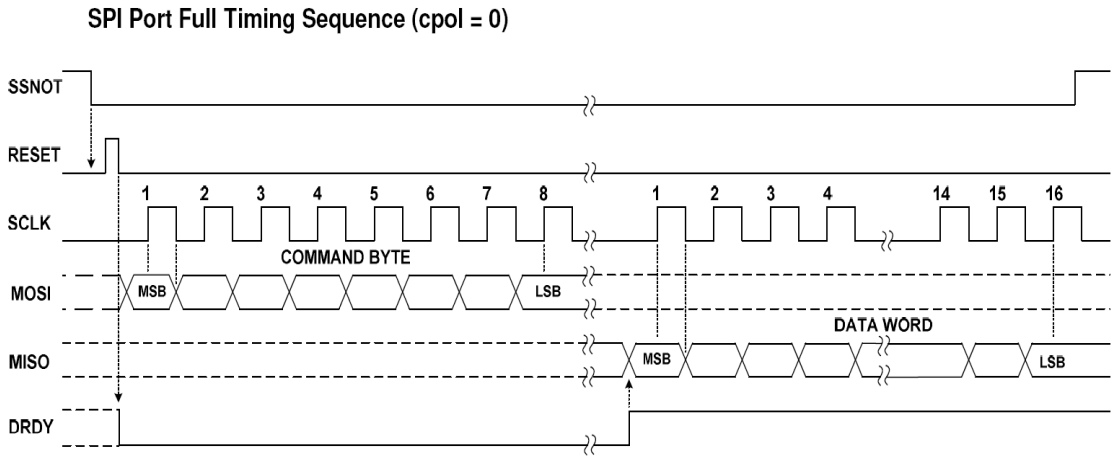

FIGURE 2.2: *SPI Port time diagram.*

4. The master sets `MOSI` to the MSB of the command byte, and waits a minimum of 100 ns before setting `SCLK` high to indicate a bit is ready.

5. The master keeps the `SCLK` high for a minimum of 100 ns before setting it low and proceeding with the next bit.

6. When all eight bits have been sent, `SCLK` is set low, and the slave will execute the command.

7. After conducting the measurement requested, the slave will set `DRDY` high to indicate data is ready.

8. The master will wait for a minimum of 100 ns. Meanwhile, the slave will set `MISO` to the MSB of the result.

9. The master then sets `SCLK` high, and immediately reads `MISO`. The master keeps `SCLK` high for a minimum of 100 ns before setting it low. This repeats until the master has read its desired number of bits.

10. When done, `SCLK` is set low, and `SSNOT` high.

2.3 PRESENTATION LAYER

The sensor only allows the magnetic field to be measured one axis at the time. To collect a sample, a command byte is sent to the sensor, and the requesting unit must wait until the sample is ready.

2.3.1 Command byte

The command byte is composed as follows:

Position	7	6	5	4	3	2	1	0
Bit	DHST	PS2	PS1	PS0	ODIR	MOT	AS1	AS0

DHST - High Speed Oscillator Test: Feature not used during operation, should be 0.

PS2, PS1, PS0 - Period Select: The period select determines the sample period to be used. A longer sample period gives a higher resolution, but also a lower sample rate. The connection between these is described in Table 2.1.

Period Select	Maximum Delay	PS2	PS1	PS0
$1/32$	$500\ \mu$s	0	0	0
$1/64$	1.0 ms	0	0	1
$1/128$	2.0 ms	0	1	0
$1/256$	4.0 ms	0	1	1
$1/512$	7.5 ms	1	0	0
$1/1024$	15 ms	1	0	1
$1/2048$	35.5 ms	1	1	0
$1/4096$	60 ms	1	1	1

TABLE 2.1: *Possibilities for selection of period. To comparision,* ≈ 3000 *counts were aquired along the Z axis with a period select of* 60 *ms, and* ≈ 1500 *counts with a period select of* 35.5 *ms.*

ODIR - Oscillator Direction: Feature not used during operation, should be 0.

MOT - Magnetic Oscillator Test: Feature not used during operation, should be 0.

AS1, AS0 - Axis Select: Used to select the axis to be sampled:

	AS1	AS0
X	0	1
Y	1	0
Z	1	1

For measuring the magnetic intensity along the Z-axis with a period select of 35.5 ms, the command byte would be:

```
0110 0011 bin = 63 hex = 99 dec
```

2.3.2 Results

The results are returned as sixteen bits, with MSB first. The format used to describe the result is 16 bit 2's complement format, ranging from -32768 to 32767. This was the same format as used on the MSP430F149 used in the test setup. Otherwise, the result can be calculated as:

```
(0xEFFF & input) - (0x8000 & input ? 65536 : 0)
```

or

```
(input > 32767 ? input - 65536 : input)
```

That is, if the `input` is larger than 32767, 65536 is substracted.

The signed result is then converted to μT by multiplying it with the Period Select fraction, and multiplying this by $62.48\ \mu\mathrm{T}/\mathrm{counts}$ at $3V$ supply. This factor depends on the supply voltage.

$$\texttt{result} \cdot \eta_{\mathrm{ps}} \cdot 62.48\ \mu\mathrm{T}/\mathrm{counts} \tag{2.1}$$

Example:

$$3000 \cdot \frac{1}{4096} \cdot 62.48 \approx 45.76\ \mu\mathrm{T} \tag{2.2}$$

FIRST CONTACT 3

The first contact was achieved by sending the command byte (in hex) `71`, which demands a measurement of the magnetic field along the X-axis during a period of 60 ms. This is the maximum period possible for sampling, and gives the highest resolution. The result returned was `01 2A` or 298 counts, equivalent to $4.47\ \mu$T.

In order to verify the results, three measurements perpendicular to each other was conducted in sequence, see Table 3. The sample period was changed to 35.5 ms to make it identical to one used in the MicroMag datasheet [PNI06].

Axis	**Command** (hex)	**Returned** (hex)	**Equivalent to**	**Geomag estimates**
X-axis	`61`	`01 8E` $= 398$ counts	$12.14\ \mu$T	$\approx 17\ \mu$T
Y-axis	`62`	`FE C7` $= -313$ counts	$-9.55\ \mu$T	
Z-axis	`63`	`05 9C` $= 1436$ counts	$43.81\ \mu$T	$\approx 45\ \mu$T

TABLE 3.1: *Three sequential measurements from the MicroMag, each perpendicular to the others. The table also contains estimates of the magnetic field measured by USGS National Geomagnetism Program [Sur05] in 1990.*

The intensity of the magnetic field in the horizontal plane can be calculated by:

$$\sqrt{12.14^2 + (-9.55)^2} \approx 15.45 \mu\,\mathrm{T} \tag{3.1}$$

This is fairly close to the estimate made by USGS National Geomagnetism Program [Sur05] in 1990 which is $\approx 17\ \mu$T. Similarly, the vertical intensity is measured to $43.08\ \mu$T and is estimated to approximately $45\ \mu$T. The total intensity is calculated by:

$$\sqrt{12.14^2 + (-9.55)^2 + 43.81^2} \approx 46.45 \mu\,\mathrm{T} \tag{3.2}$$

This is also considered close to the estimate by the Geomac Program, which is just below 50μT.

To test the sensor furtherly, a test over time is made with the sensor:

1. The sensor is placed on the table.
2. The sensor is rotated 90 deg. about the X-axis
3. The sensor is rotated 90 deg. about the X-axis back to horizontal.
4. The sensor is rotated 90 deg. about the Y-axis
5. The sensor is rotated 90 deg. about the Y-axis back to horizontal.

It should be noted, that the test is conducted by rotating the sensor by hand, so some uncertainties in connection to this should be taken into account. Also, during the test, the sensor is not mounted on the X-Pro, which is expected to be a more noiseful environment, see Chapter 2 on page 46. The test results can be seen in Figure 3.1.

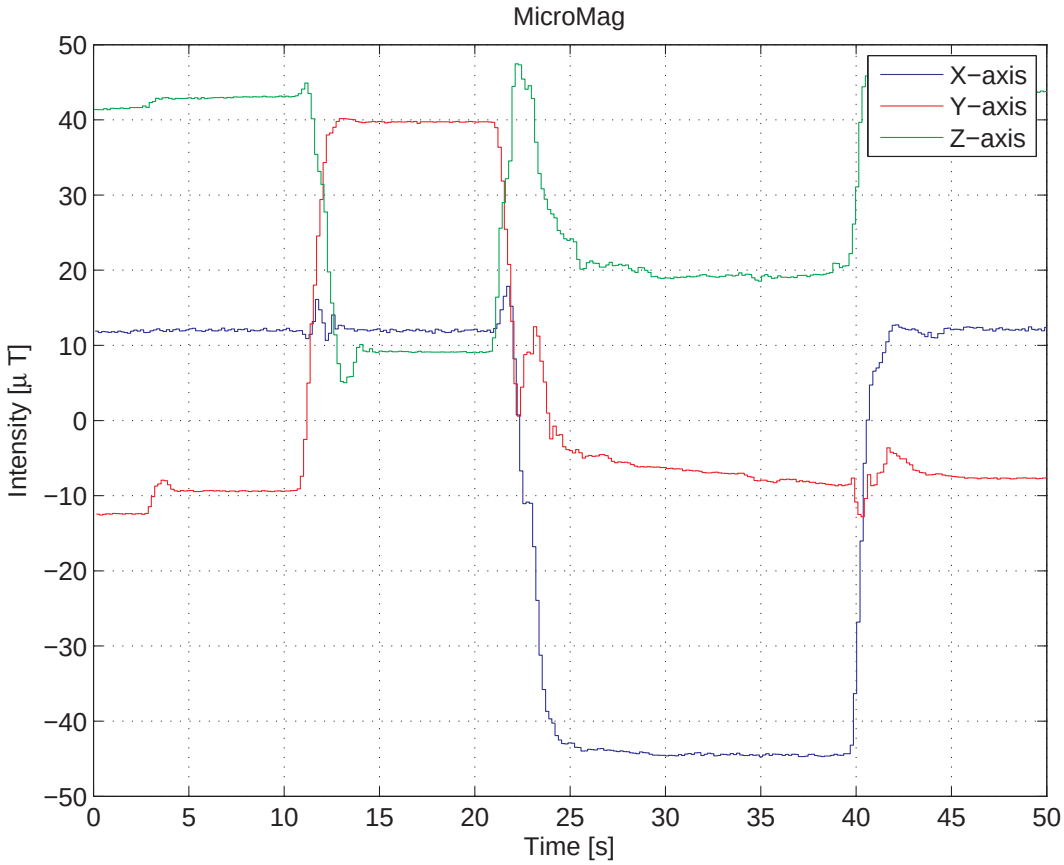

FIGURE 3.1: *Test over time of the MicroMag raw data.*

Apparently, the axes seem to switch places, when the magnetometer is rotated 90 deg., which is also expected. The magnitude of the intensity has also been investigated, and is showed in Figure 3.2. On the figure, the magnitude does not appear to be constant. This indicates one, or a combination, of the following:

- The axes are unevenly scaled when measured, perhaps due to inaccuracies in the individual sensors.
- One or more soft- or hard-iron disturbances are present. These can appear both in the indoor environment, where the test has been conducted, or on the print circuit board on which the MicroMag is mounted. The print circuit board has four metal objects mounted on it, which might not be far enough from the sensor.

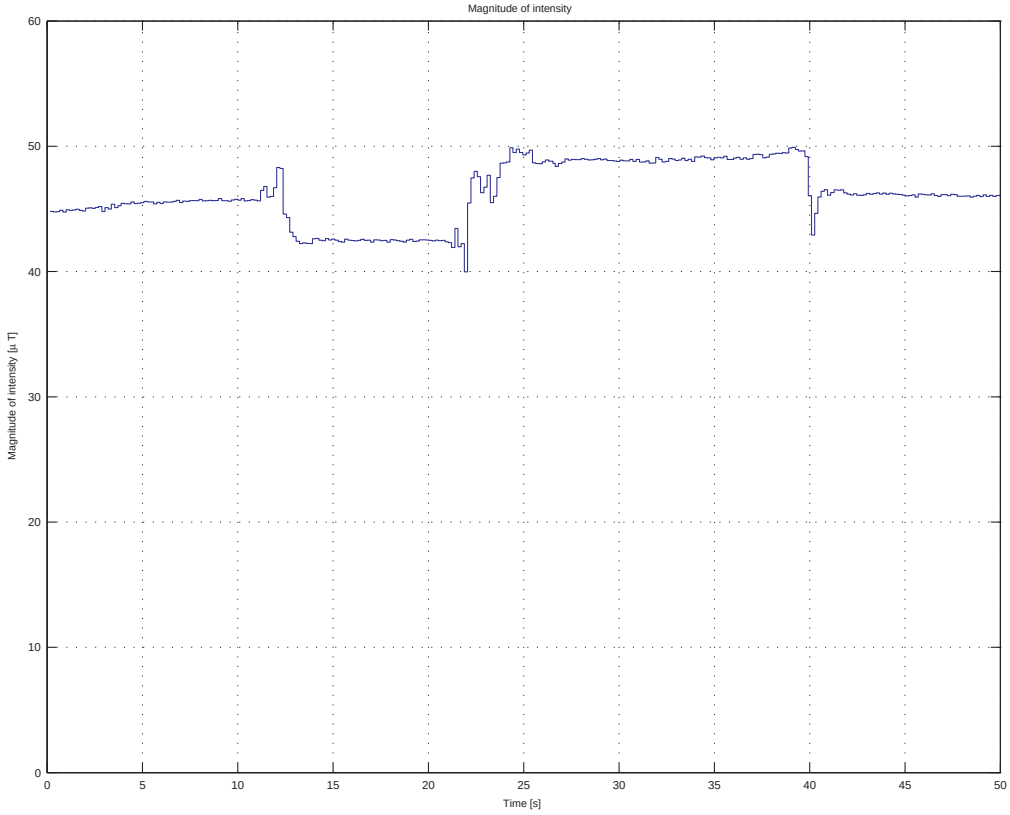

FIGURE 3.2: *Magnitude of the test data.*

WORKSHEET XII: * ROBOSTIX^{TM} INTERFACE AND SW DESIGN.

17/10-06

This worksheet deals with the assignment of the I/O ports and the software design for the robostix^{TM}. Figure 1.1 on page 16 describes the overall structure and is the foundation of the pin assignment. Most drivers for the sensors are designed as estimators of Discrete Event Systems, and fault detection is built in to an extent considered appropriate on this low-level handling of the sensor.

CONTENTS OF WORKSHEET XII

1 robostix^{TM} main process **103**

1.1 robostix^{TM} supervisor design . . . 103

1.1.1 Priorities . . . 104

1.1.2 Overall design . . . 105

1.1.3 Implementation . . . 105

2 robostix^{TM} hardware interface **107**

2.1 General description of the robostix^{TM}'s pin set up . . . 107

2.2 Input/Output (I/O) pin assignment . . . 107

2.2.1 Sensors using UARTs or I^2C bus . . . 107

2.2.2 PWM-output . . . 108

2.2.3 PORT A + Interrupts . . . 109

3 robostix^{TM} driver design **112**

3.1 Assignment of timers . . . 112

3.1.1 Tasks . . . 112

3.1.2 Priorities . . . 113

3.1.3 Available timers . . . 114

3.1.4 Allocation of timers . . . 114

3.2 Motor driver . . . 116

3.2.1 Interface . . . 116

3.3 Magnetometer interface . . . 117

3.3.1 Overall design . . . 117

3.3.2 Interface . . . 118

3.4 Tachometer interface . . . 120

3.4.1 Overall design . . . 120

3.4.2 Interface . . . 122

3.5 Remote Control (R/C) interface . . . 122

3.5.1 Overall design . . . 124

3.5.2 Interface . . . 126

3.6 Inertial Measurement Unit (IMU) interface . . . 126

3.6.1 Overall design 127
3.6.2 Interface 127

ROBOSTIX^{TM} MAIN PROCESS

1

The purpose of this chapter is to introduce the robostix^{TM} main process, its interaction with connected sensors and design of the main process.

Most of the system sensors are connected to the robostix^{TM}, although most of the data is transmitted to the gumstix^{TM} for further estimation and controlling. The fact that the robostix^{TM} is the only connection point to these sensors, demands that there is liable and stable communication between these. Drivers must be written for each of the sensors, ensuring the demanded communication between the robostix^{TM} and the sensors. These drivers have to be included as well in a single process which makes sure that every single sensor is overviewed as frequently as needed. This process is defined as main function from now on.

The main function should be to the system sensors, as the brain is to the human body. It initializes the robostix^{TM} and the connected sensors at system startup and enters a state where it listens and responds to the sensors whenever needed. The main routine is supposed to be active as long as the system is running. The processes that the main routine interacts with are the following:

- IMU
- Magnetometer
- Tachometer
- R/C receiver
- Motor drivers
- gumstix^{TM} communication

This internal communication between those and the main function is illustrated on Figure 1.1.

1.1 ROBOSTIX^{TM} SUPERVISOR DESIGN

The following subsections explain how the robostix^{TM} main process will be designed in such a way that its behavior will be predictable and as fail-safe as possible. If error is detected in one state, the system should respond to it by indicating the failure and/or return to a safe state, proceed if appropriate or abort otherwise.

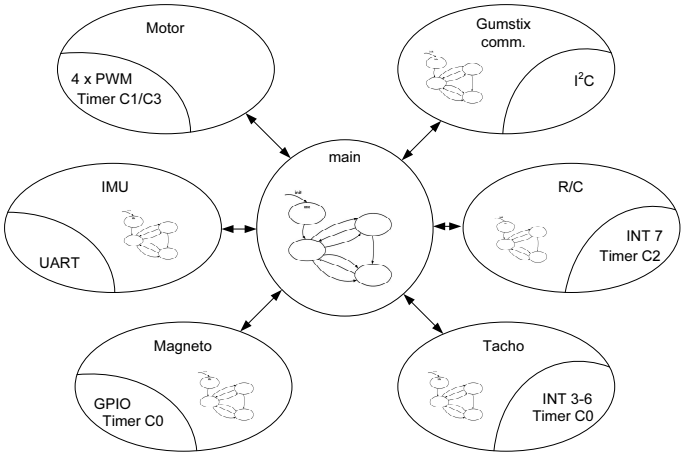

FIGURE 1.1: *The main process interacting with drivers.*

1.1.1 Priorities

The goal in designing the main function is to gain control of the Draganflyer X-Pro (X-Pro) both manually and automatically. There are couple of demands that need to be stated for the operation of the robostix^{TM}. These demands will have the purpose of ensuring a stable system with known behavior. At startup, the normal behavior of the system is the `idle` state. The reason for this is to ensure a safe startup, where the rotors are inactive and the R/C transmitter is turned on, before any controlling takes place. This way, the X-Pro will not automatically take off at power-up.

There are three active states the X-Pro can be controlled from:

1. The `human pilot` state, where the X-Pro is controlled from the R/C transmitter, is the state that every flight should be started from. It must always be possible for the human pilot to take control from any state, by use of a safety switch on the R/C transmitter, in case of an emergency.

2. The gumstix^{TM} `pilot` state is where the control signals are calculated by the gumstix^{TM}, based upon sensor measurement. It can be switched between the `human pilot` and the gumstix^{TM} `pilot` as desired during flight. If a fault is detected on the R/C connection, the gumstix^{TM} should take over control, as it is presumed to have the best controller and state estimates.

3. The robostix^{TM} `pilot` state has the main purpose of being the last resort for controlling. In case of gumstix^{TM} faults, it serves mainly as indicator for the `human pilot` to take over. Meanwhile it uses some primitive controlling, which should

serve as the system ”parachute” to prevent major damages.

These demands require the need for a supervision system that ensures that, initially defined, normal behavior is followed and alarms and/or responds if the normal behavior is not followed. A typical behavior that is not normal would have to be responded to accordingly by the supervisor, which can be based on Finite State Automata (FSA). The robostix^{TM} main function is where the supervision system is implemented and it is illustrated in Figure 1.2.

1.1.2 Overall design

The main function is designed as an FSA, the design is illustrated on Figure 1.2.

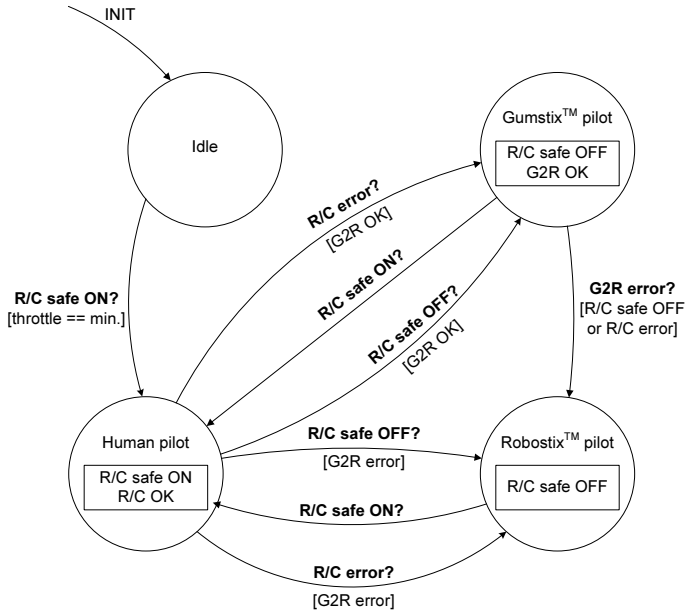

FIGURE 1.2: *The FSA for the main procedure.*

The FSA bases its transitions on status messages from the communication lines (Gumstix-to-Robostix connection (G2R) and R/C), hence it must therefore be possible to detect errors in these.

1.1.3 Implementation

`rb_idle`: At startup the main routine ensures that the system initializes and enters the `rb_idle` state. The system stands still in idle mode until another state is requested.

rb_human: When R/C safe is switched ON and it has been assured that throttle is adjusted to minimum this state can be reached, otherwise the system makes sure the system does not change from idle state.
The `rb_human` state can also be reached from the `rb_gums_pilot` state or the `rb_robo_pilot` state as well, as described in the following.

rb_gums_pilot: When the G2R is OK, the X-Pro can be set to be controlled by the gumstix^{TM} (`rb_gums_pilot` state). This is done manually by switching OFF the R/C safe or automatically if the R/C receives error signals.
The FSA returns from the `rb_gums_pilot` state to `rb_human` state, if R/C safe is switched ON.
The FSA switches from the (`rb_gums_pilot` state to the `rb_robo_pilot` state, if there occurs an G2R error meanwhile the R/C safe is switched OFF.

rb_robo_pilot: When the G2R is not OK, the X-Pro can be set to be controlled by the robostix^{TM} (`rb_robo_pilot` state).
This is done by manually switching the R/C safe OFF or automatically if the R/C transmits error signals to the robostix^{TM}. The buzzer and indicator Light Emitting Diode (LED) must be turned on in this state it indicate towards the pilot that something is wrong. The FSA returns from the `rb_robo_pilot` state to `rb_human` state, if the R/C safe is switched ON.

ROBOSTIX^{TM} HARDWARE INTERFACE 2

2.1 GENERAL DESCRIPTION OF THE ROBOSTIX^{TM}'S PIN SET UP

In order to get measurements from all the sensors needed to get the X-Pro up and flying, it is important to distribute the I/O-pins of the robostix^{TM} board correctly. This section is written in aspect to [Gum06c].

All the pin assignments will be carried out using the following Figure 2.1.

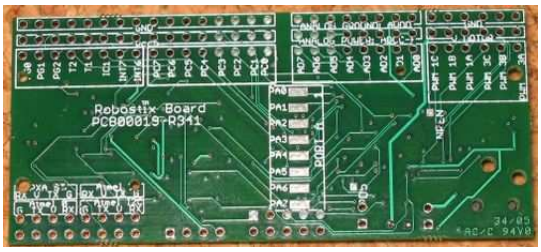

FIGURE 2.1: *Bottom view of the robostix^{TM} board[Gum06c].*

With this picture in mind the pin assignments to the robostix^{TM} board can take place.

2.2 I/O PIN ASSIGNMENT

In this section, the sensors will be assigned to the robostix^{TM} and the pin connections will be described. The sensors will be connected as described in Figure 1.1 on page 16. The sensors will be divided into subsections dependent on which type of I/O the sensors use.

2.2.1 Sensors using UARTs or I^{2}C bus

Figure 2.2 shows the location of the Universal Asynchronous Receiver Transmitter (UART)s and Inter-Integrated Circuit (I^{2}C) bus on the robostix^{TM}.

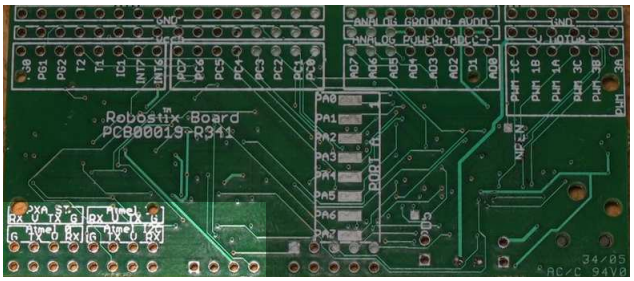

FIGURE 2.2: *The location of the UARTs and I²C on the robostix™ is marked with the light area in the lower left corner.*

When programming the ATMega128, the Standard Universal Synchronous-Asynchronous Receiver/Transmitter (STUART) and UART-0 are connected as described on [Gum06b]. The I²C connector is located together with the UARTs.

In Table 2.1, the pin connections for the sensors which are to be connected to either the robostix™ or gumstix™ using a UART or I²C are listed.

Sensor	Port name	Destination
Global Positioning System (GPS)	`PXA_ST` (STUART)	gumstix™
IMU	`UART-0 (Atmel 0)`	robostix™
SOund Detection And Ranging (SODAR)	`Atmel I²C`	robostix™or gumstix™
PC Console	`GUMSTIX`	gumstix™

TABLE 2.1: *The sensors which are connected to the robostix™ or gumstix™, either using a UART or I²C.*

Due to the fact that the GPS-module uses the STUART, and the IMU uses the UART-0 these sensors are to be removed every time the robostix™ is to be programmed. The I²C interface will be used as communication between the robostix™ and gumstix™.

2.2.2 PWM-output

Figure 2.3 shows the location of the Pulse Width Modulation (PWM)-output ports of the robostix™ which are used to generate the duty cycles to the motor drivers.
As can be seen from the picture, there are six output pins, but only four motors, so the functionality of the two last PWM pins can be altered if needed.

Table 2.2 shows which output pins from the robostix™ that is to be connected to the motor drivers. It also shows which pin to connect each motor drive to the robostix™.

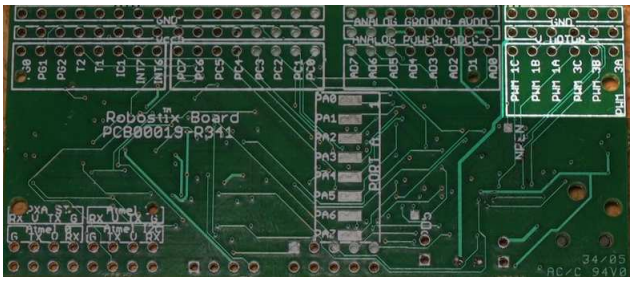

FIGURE 2.3: *The location of the PWM on the robostix^{TM} is marked by the light area in the upper right corner.*

Port name	Destination
`PWM 1A`	Front motor driver
`PWM 1B`	Left motor driver
`PWM 1C`	Back motor driver
`PWM 3A`	Right motor driver

TABLE 2.2: *This table describes which motor drive is connected to the PWM-output of the robostix^{TM}.*

2.2.3 PORT A + Interrupts

Figure 2.4 shows the location of `PORT A` and `INT3 - INT7`.

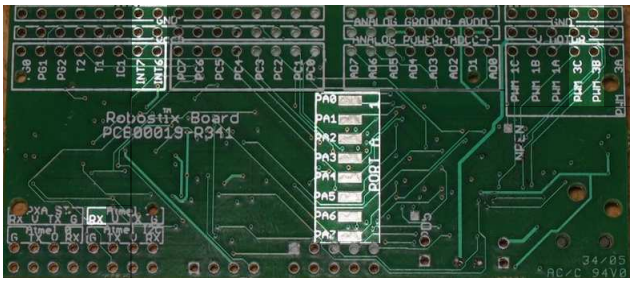

FIGURE 2.4: *The location of `PORT A` and `INT3 - INT7` on the robostix^{TM} which is marked the light areas.*

Table 2.3 shows where the 4 rotor speed sensors, the R/C module and the magnetometer are to be connected to the robostix^{TM}

Sensor	Port name	Destination
Front rotor speed	INT3 (Atmel RxD1)	robostix^{TM}
Left rotor speed	INT4 (PWM 3B)	robostix^{TM}
Back rotor speed	INT5 (PWM 3C)	robostix^{TM}
Right rotor speed	INT6	robostix^{TM}
R/C	INT7	robostix^{TM}
Magnetometer (SCLK)	PA0	gumstix^{TM}
Magnetometer (MOSI)	PA1	gumstix^{TM}
Magnetometer (MISO)	PA2	gumstix^{TM}
Magnetometer (SSNOT)	PA3	gumstix^{TM}
Magnetometer (RESET)	PA4	gumstix^{TM}
Magnetometer (DRDY)	PA5	gumstix^{TM}
Buzzer	PC7	robostix^{TM}
Indicator LED	PC6	robostix^{TM}

TABLE 2.3: *Description of where the four rotor speed sensors, R/C module and magnetometer are to be connected to the robostix^{TM}. The last column describes which architecture is to get the data from the sensor.*

As it can be seen from the table above, the four motor speed sensors are using the external interrupts (3 - 6). The reason for this is that the external interrupt pins of the robostix^{TM}-board are located different places such as INT3 is located on the UART-1 (Atmel 1)s receive pin. This limits the numbers of robostix UARTs to one, which was allocated earlier in this section (Subsection 2.2.1 on page 107). INT4 and INT5 are located on the unused PWM outputs.

The R/C-module uses only one input to the robostix^{TM}. The R/C channels 1, 3 and 5 are added in an OR-gate and the output of the OR-gate is connected to INT7. By investigating the first contact analysis of the R/C, it can be seen that the channels seperately are presented sequentially, and with a minimum length of $1ms$. Hence it is possible to combine the signals as described into one, and determine channel lengths based on the timing of changes in the logical level.Eventuelt et scope billede for at underbygge dette??

The magnetometer uses 6 input-pins where 4 of these are used to imitate a Serial Peripheral Interface (SPI) interface, namely the SCLK, MOSI, MISO and SSNOT. The robostix^{TM} already has a SPI interface, but this is used when programming the robostix^{TM} from the gumstix^{TM}, therefore one has been created.

The idea by the buzzer and LED is to indicate when the on board computer (gumstix^{TM}) fails or the communication between the robostix^{TM} is down. The further usage will be described then designing the main application to the robostix^{TM}.

Now the sensors can be mounted to the robostix^{TM} and the drivers needed to access

the sensors can be developed.

ROBOSTIX^{TM} DRIVER DESIGN

3

This chapter contains documentation of the design and certain implementation details of the drivers, that are responsible for collecting samples from their respective sensors or generating signals for the motors and status indicators. First, the hardware ressources of the robostix^{TM}'s Atmega128 are distributed among the drivers, and shared ressources are configured. Secondly, the drivers are designed, most of them based on FSA design. This is considered a suitable method, since each sensor produce signals of a certain complexity, and since simultaneous monitoring and communication with several sensors is required.

3.1 ASSIGNMENT OF TIMERS

First, this section describes the tasks, where timers are considered necessary, and what priorities lays the grounds for distributing the timers. Secondly, it describes what tasks the four timers on the Robostix processor, Atmega128, are assigned to, and how they are configured.

3.1.1 Tasks

The Atmega128 contains four timers, which are to be used for different tasks on the processor. These tasks where time is considered a necessity are:

PWM for motors: The four motors are each interfaced through a PWM signal with a period of $1/300$ s. It should be noted, that the signal must be inverted, since a signal of constantly 5 V (100% duty-cycle) will result in no voltage to the motors, and vice versa. The resolution must be high, since the nominal operating area is considered to be very narrow compared to the possible operating area of the motors.

R/C-receiver: The R/C-input is read through a single interrupt-pin, Table 2.3 on page 110, and it must be possible to measure the time between interrupts in a reasonable resolution. The time interval should be at least 1 ms, except if the Robostix is enabled in the middle of a signal, by which the time interval could be down to ≈ 0 ms. Nominally, there should be between 1 ms and 2 ms between pulses, but between sequences/packages of R/C inputs, there can be up to $22.5\,\text{ms} - 5 \times 1\,\text{ms}$. For precision control, it is desired to also have a high resolution on the time interval measurements.

Tachometer: At hover, the rotors rotate at $\approx 146\ ^{\text{rad}}/_{\text{s}} \approx 23.25\ ^{\text{rounds}}/_{\text{s}}$. The maximum

expected rotor velocity is $30\ ^{\text{rounds}}/_{\text{s}}$, which would introduce a pulse every $^{1}/_{30}$ s. If the rotor is put to a halt, no pulses would be introduced, by which the rotor velocity would not be updated. It is a possibility to define a timeout, after which the rotor is defined as zero or low velocity. Based on experiences from the previous project on the X-Pro with the same tachometers, it is found that a threshold of $^{1}/_{8}$ s is suitable [ABG+06, Fig. 13.2 p. 105].

Magnetometer: To take measurements from the magnetometer, a commandbyte is transmittet to it, and after a delay, the magnetometer replies with `Data Ready` (`DRDY`). If the magnetometer does not respond within a certain maximum period, it can be assumed that either the commandbyte has not been transmitted correctly, the magnetometer is without power, or the communication is lost. The same can be assumed if the magnetometer replies before a certain minimum period. Due to this, it must be possible for the magnetometer driver to determine the time between sending the commandbyte and receiving `DRDY`.

Also, the magnetometer transmits through SPI and two extra I/O-ports. The Atmega128 has an SPI port, but this is used for programming it, so a software SPI-driver is developed. According to the magnetometer datasheet [PNI06], a pulse must be at least 100 ns long. One clock cycle on the Atmega128 is $^{1}/_{16}\ \mu$s $= 62.5$ ns, and a maximum of 16 bits are transmitted, which gives a minimum of $2 \cdot 16 \cdot 100\,\text{ns} = 3.2\ \mu$s transmission time. Additionally, there is a risk that it will take more clock cycles to calculate the correct transmission sequence. A prototype implementation of the software SPI has been developed and implemented, and showed $\approx 60\ \mu$s for receiving 16 bits, which is considered acceptable. Thus, a timer for timing the SPI transmission is not necessary.

3.1.2 Priorities

The allocation of the timers is based on the following priorities:

- Some timers are by hardware dedicated to specific purposes. This priority is made with respect to the fact, that two of the timers are dedicated for producing PWM-signals, which is useful for the motor HW interface.

- Many software interrupts should be avoided. This is due to the fact, that each interrupt creates an overhead. If the interrupt can be avoided by changing the prescaling of the clock to the timer or choosing a timer with larger resolution, this should be done first.

- Tasks are grouped by similar time characteristics. Since there is a limited number of timers, some tasks must share the same timer. This grouping is based on tasks, where the timer period or clock frequency can be the same, without violating the demands described in 3.1.1 on page 112.

3.1.3 Available timers

The Atmega128 has four timers available, which are described here, based on the datasheet [Atm, p. 92-161]:

Timer/Counter0: 8-bit timer, one double-buffered output compare register for glitch-free PWM generation, designed for external 32 kHz watch crystal or prescaled system clock ($^1/_1$, $^1/_8$, $^1/_{32}$, $^1/_{64}$, $^1/_{128}$, $^1/_{256}$ or $^1/_{1024}$).

Timer/Counter1: 16-bit timer, three double-buffered output compare registers for glitch-free PWM generation, possibility of external clock / event-generator, shared prescaler with Timer/Counter2 and Timer/Counter3 (individual prescale select of $^1/_1$, $^1/_8$, $^1/_{64}$, $^1/_{256}$ or $^1/_{1024}$).

Timer/Counter2: 8-bit timer, one double-buffered output compare register for glitch-free PWM generation, possibility of external clock / event-generator, same prescaler as Timer/Counter1.

Timer/Counter3: Same propterties as Timer/Counter1. Please note, that the output compare register `OC3B` and `OC3C` are not available on the Atmega128 output pins, since there are occupied by the tachometers.

3.1.4 Allocation of timers

Based on the tasks, the available timers and the priorities, the following assignment of timers has been made:

Timer/Counter1 + Timer/Counter3

Timer/Counter1 and Timer/Counter3 are primarely assigned to generation of four PWM-pulses for the motor HW interface. Both timers have been configured as follows:

- Waveform Generation Mode: Fast PWM, `ICRn` as `TOP`.
- Compare Output Mode for `1A`, `1B`, `1C` and `3A`: Inverting mode (Set `OCnx` on compare match, clear at `BOTTOM`)

- Prescaler: $^1/_1 \Rightarrow 16\,\text{MHz}$
- `TOP`-value defines via `ICR1` and `ICR3`: $\frac{16\,\text{MHz}}{300\,\text{Hz}} - 1 \approx 53\,332\,\text{counts}$

This gives a theoretical maximum resolution of

$$\frac{\log\left(\text{TOP}+1\right)}{\log\left(2\right)} = \frac{\log\left(53\,332+1\right)}{\log\left(2\right)} \approx 15.7\,\text{bits} \tag{3.1}$$

and a frequency of:

$$\frac{16 \cdot 10^6}{53\,333} \approx 300.002\,\text{Hz} \tag{3.2}$$

Timer/Counter0

Timer/Counter0 is assigned to the magnetometer, and is used to determine whether samples are returned from the magnetometer within a specific interval. The tachometer also uses this timer for determining intervals between pulses. The timer is configured as follows:

- Waveform Generation Mode: Normal, `MAX` as `TOP`.
- Compare Output Mode: Normal port operation, `OC0` disconnected.
- Prescaler: $^1/_{1024} \Rightarrow 15\,625\,\text{Hz}$
- Interrupt on overflow. Increments an 8 bit variable, giving a timer with a total of 16 bits.

The resolution of this counter is:

$$\frac{1\,\text{count}}{15\,625\ {}^{\text{counts}}/_{\text{s}}} = 64\ {}^{\mu\text{s}}/_{\text{count}} \tag{3.3}$$

Using only the 8 bit timer, the maximum counter period is:

$$\frac{256\,\text{counts}}{15\,625\ {}^{\text{counts}}/_{\text{s}}} = 16.384\,\text{ms} \tag{3.4}$$

This is not enough for determining the maximum delay of the magnetometer. Using the 16 bits available by combining the counter and the overflow counter, the maximum counter period is:

$$\frac{65\,536\,\text{counts}}{15\,625\ {}^{\text{counts}}/_{\text{s}}} = 4.19\,\text{s} \tag{3.5}$$

Timer/Counter2

This timer/counter is assigned to the R/C-module to calculate the length of each channel. To avoid timer interrupts all the time timer/counter2 is configured by the following:

- Waveform Generation Mode: Normal, `MAX` as `TOP`.
- Compare Output Mode: Normal port operation, `OC2` disconnected.
- Prescaler: $1/64 \Rightarrow 250\,\text{kHz}$
- Interrupt on overflow. Increments an 8 bit variable, giving a timer with a total of 16 bits.

The resolution of this counter is:

$$\frac{1\,\text{count}}{250000\ {}^{\text{counts}}/_{\text{s}}} = 4\ {}^{\mu\text{s}}/_{\text{count}} \tag{3.6}$$

Using only the 8 bit timer, the maximum counter period is:

$$\frac{256\,\text{counts}}{250000\ {}^{\text{counts}}/_{\text{s}}} = 1.024\,\text{ms} \tag{3.7}$$

This is not enough for determining the total length of all R/C channels. Using the 16 bits available by combining the counter and the overflow counter, the maximum counter period is:

$$\frac{65\ 536\,\text{counts}}{250000\ {}^{\text{counts}}/_{\text{s}}} = 262.144\,\text{ms} \tag{3.8}$$

3.2 MOTOR DRIVER

As described in Section 3.1.4 on page 114, Timer/Counter1 and Timer/Counter3 are used for generating the PWM signals. The dutycycle is set by changing the output compare registers, which are each assigned to a motor. Both timers have double-buffered output compare registers, which essentially means, that new output compare values will be buffered until the timer hits the `TOP` value, after which they will take effect. This ensures a glitch-free PWM-signal when changing the dutycycles.

3.2.1 Interface

Two functions are considered necessary for the Motor SW driver: One for initializing the PWM signals, and one for updating the dutycycles.

`void motor_hw_init(void):` This function initializes the timers with the parameters described in Section 3.1.4 on page 114. By default, the dutycycle is set to 0%.

`void motor_set_duty(motor_package_t *):` This function takes four dutycycles in a package, saturates them to be between `BOTTOM` and `TOP`, and puts them in the output compare buffer, see Figure 3.1.

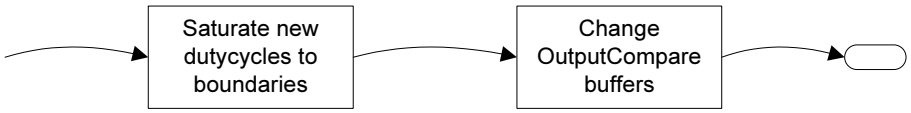

FIGURE 3.1: *The procedure for updating dutycycles.*

3.3 MAGNETOMETER INTERFACE

The magnetometer has an onboard processor, which takes samples on demand. In its datasheet, a sequence of signals for taking a sample is described, and an FSA is designed based on this describtion.

3.3.1 Overall design

The FSA has been designed in such a way, that it is possible to detect certain errors in the communication. These errors are:

- Unexpected signals such as `DRDY` when sending a command, or `!DRDY` when receiving a sample. This could indicate a loose wire, or a fault in the transmission.
- Timeout, when the magnetometer does not respond within a maximum delay. This could indicate either a fault when transmitting the command, a loose wire, or the magnetometer is without power.
- `DRDY` before a minimum, indicating a loose wire, or a fault in the transmission of the command.

When an error has been detected, the driver will retry to collect a sample. After three retries, the communication to the magnetometer is declared lost to the main system, but the driver will continously try to re-establish contact to the magnetometer. Figures 3.2, 3.3 and 3.4 illustrate how the driver is designed to control the communication with the magnetometer, keeping track of samples and shifts between sampling one axis at the

time, respectively. The magnetometer driver will independently request samples for one axis at the time, and make them available to the main system.

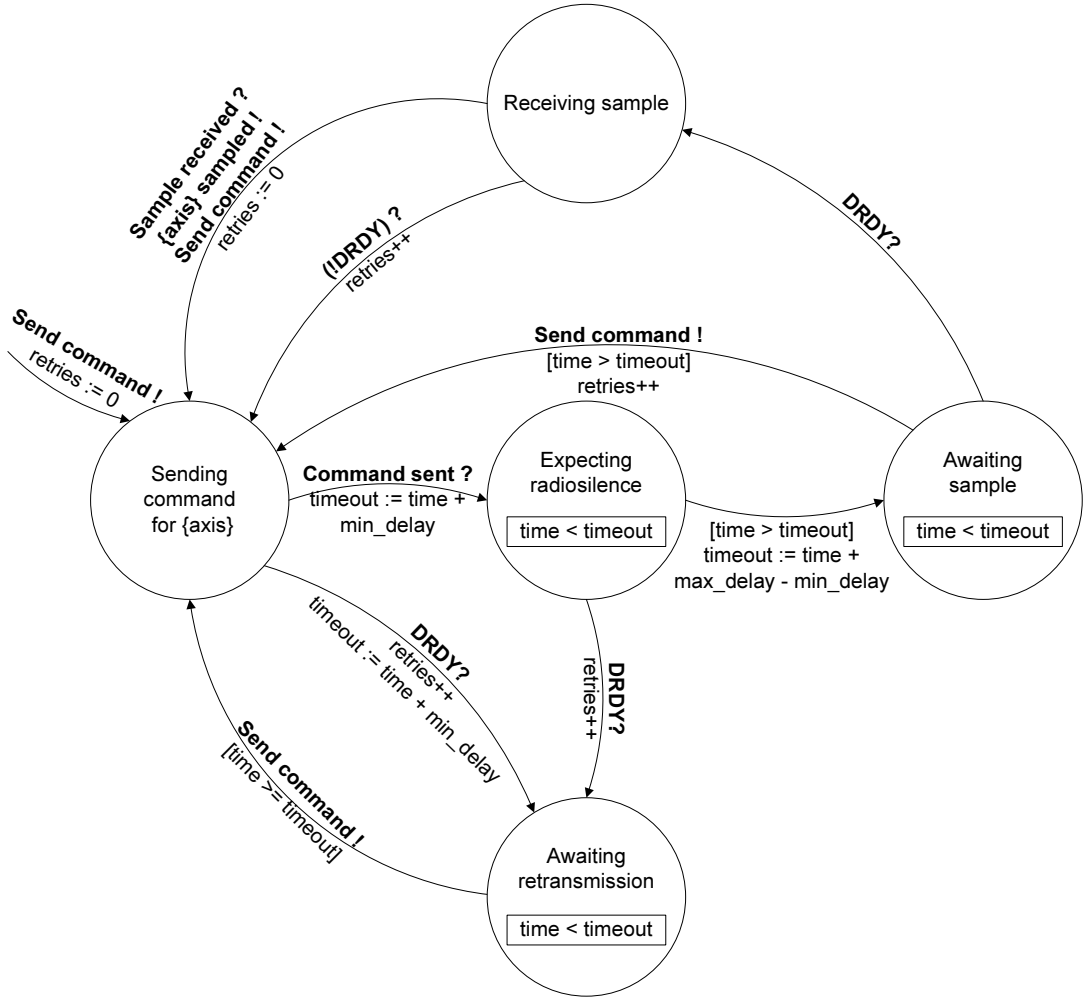

FIGURE 3.2: *The FSA for the magnetometer driver.*

3.3.2 Interface

The magnetometer driver interface defines three types specific for the magnetometer driver:

`mag_axis_t`: This is an enumeration type of the values `mag_x_axis`, `mag_y_axis` and `mag_z_axis` to be used in function calls when reading a sensor.

`mag_sensorvalue_t`: This is the type used for storing received samples.

`mag_return_t`: This is an enumeration type of the values `mag_new_sample`, `mag_no_sample`

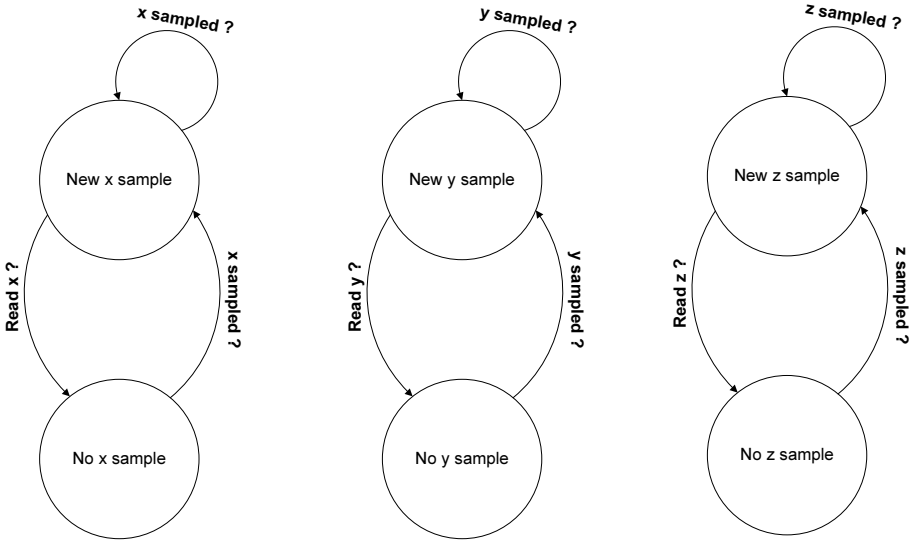

FIGURE 3.3: *The FSA for the availability of new samples.*

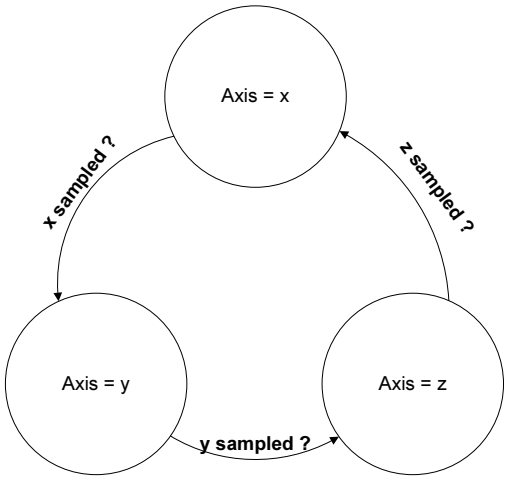

FIGURE 3.4: *The FSA for the current axis being sampled.*

and `mag_comm_error`, which is the possible values to be returned, when the main system requests the magnetometer driver for samples.

The interface is also defined by the functions, which can be called from the main system:

`void magnetometer_init(void):` Initiates the Atmega128 hardware associated with the magnetometer driver, and initiates sampling from the magnetometer.

`mag_return_t magnetometer_readsensor(mag_axis_t, mag_sensorvalue_t *):` The function used for continously asking the magnetometer driver for new samples of a certain axis. This function should be called with a maximum interval of the minimum delay between sending a command byte and receiving a sample.

`void magnetometer_update_state(void):` This should be called as frequent as possible, as it updates the state of the internal FSA.

`void magnetometer_receive_timerrollover(void):`] This is used by the Timer/Counter0 overflow Interrupt Service Routine (ISR) to notify the magnetometer driver, that the timer has rolled over.

3.4 TACHOMETER INTERFACE

The tachometers have the purpose of measuring for the velocity of each rotor. The tachometers are located on the X-Pro by the rotors, whereas the measurements are conducted by generating a pulse each time a rotor has taken one revolution. These pulses are lead directly into the robostix^{TM} through four interrupt pins, i.e. `INT3` to `INT6`. The FSA for the tachometer driver is described in the following.

3.4.1 Overall design

The FSA for the tachometers is designed in such a way that if an error occurs, in the sensor or the connection between the sensor and robostix^{TM}, the sensor should be discarded, i.e.:

- A minimum time between two pulses from the same tachometer has been set. If pulses occur more frequent than this minimum time, the reading is discarded. This could be an indication of a loose wire or an error in the sensor.
- A maximum time between two pulses has been introduced to define, that if the time is larger than this value, the velocity of the rotors are one third of the velocity in hover and are therefore said to stand still. This is set because the time to detect that the rotors are standing still would take too long time if it would detect it at all.

Figure 3.5 illustrates the FSA for the tachometer driver. A revolution is defined as the time between two measured pulses signals from the same tachometer to the robostix^{TM}. When the driver is initialized it will wait for the first interrupt to come before doing anything else. The driver then measures the time it takes for the rotor to take one revolution.

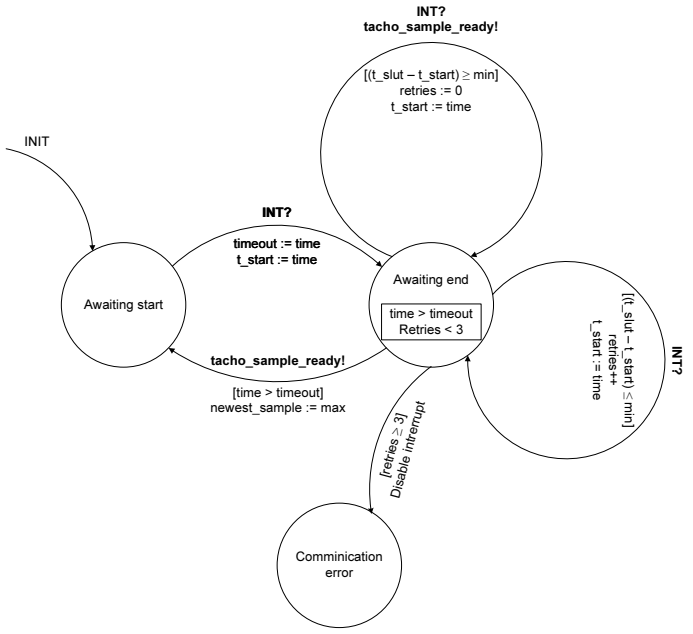

FIGURE 3.5: *The FSA for the tachometer driver.*

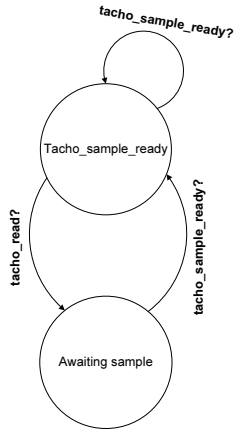

FIGURE 3.6: *The FSA for the availability of new samples in the tachometer.*

This second pulse has to occur between a minimum and maximum time to be accepted. If two pulses from the same tachometer occurs within the minimum time the sample will be discarded and the driver will wait for another interrupt from that tachometer. If this happens three times in a row before a sample has been accepted, the tachome-

ter will be discarded and the robostix^{TM} must be reset to make the discarded tachometer functional again. If the time is higher, than the maximum allowed time, the driver considers the rotor to be standing still. Figure 3.6 illustrates how the tachometer driver updates the state that shows if a new sample is ready.

3.4.2 Interface

There are three types defined for the tachometer driver:

`tacho_return_t`: Is an enumeration of the type of values, `tacho_new_sample`, `tacho_no_sample` and `tacho_comm_error`, which are the values that can be returned when the main system investigates if there are new tachometer samples.

`tacho_id_t`: Is an enumeration of the type of values, `tacho_front`, `tacho_left`, `tacho_back` and `tacho_right`, which are the values that can be returned form the tachometer ISR.

`tacho_state_t`: Is an enumeration of the type of values, `tacho_awaiting_t_start`, `tacho_awaiting_t_end` and `tacho_comm_err`, which are the possible states, the internal FSA of the tachometer can enter.

`long_time_t`: Is defines the type to be an unsigned 16bit integer. This type definition is made in the driver for Counter0.

Furthermore two functions are defined which can be called from the main system:

`void tacho_hw_init(void)`: This function initializes the robostix^{TM} hardware used for the tachometer driver.

`tacho_return_t tacho_sample_ready(tacho_id_t, long_time_t *)`: This function is used from the main system, to check if there is a new tachometer measurement from any of the four tachometers.

3.5 R/C INTERFACE

In order to control the X-Pro using the R/C, the robostix^{TM} needs to get the information from the R/C-module and together with the IMU generate the appropriate PWM-signals. This section deals with the driver design for the R/C-module.

The R/C-module is connected to `INT7` on the robostix^{TM} through an OR-gate (see Section 2.2.3 on page 109). Each rising and falling edge of the signal generates an interrupt to

the robostix^{TM}, which gives the possibility to determine each channel length. A example of the signal from the OR-gate is illustrated in Figure 3.7.

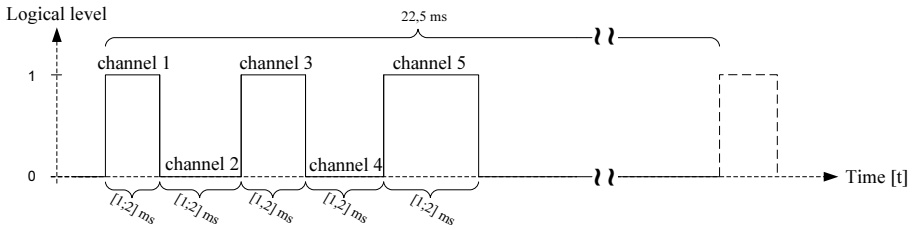

FIGURE 3.7: *The signal lead into* `INT7`*. The five channels can vary individually between* $[1; 2]$ *ms and a total R/C-package is* $22, 5$ *ms.*

Further details on the R/C-module signal is located in the "Hardware architecture and interface" worksheet in Section 4 on page 40.

The convenience and cleverness of this solution is that a minimum of interrupts is used on the robostix^{TM} without making the software calculation heavy.

Figure 3.8 show the Discrete Event System (DES) of the R/C-module.

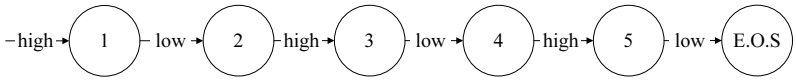

FIGURE 3.8: *The DES system when receiving the R/C-signal.*

The DES changes state every time the transition is fulfilled. Based on this a estimator can be set up in order to give a clear view when designing the driver. In Figure 3.9 a DES description of an R/C signal estimator is illustrated, based on the input.

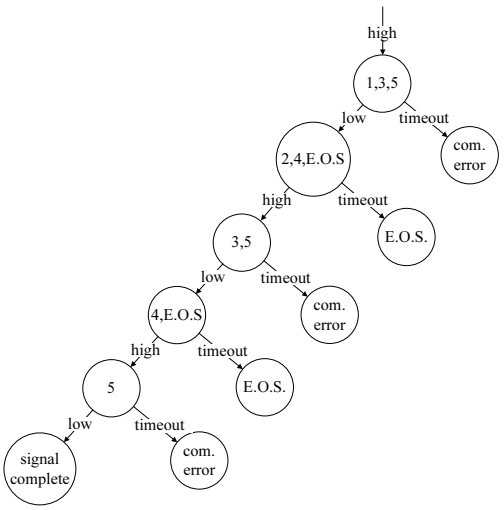

FIGURE 3.9: *The DES-estimators possible transitions, based on Figure 3.8.*

There are three possible places where a low-to-high interrupt can be started, namely channel 1, 3 or 5. This is illustated by the first circle, which contains the states 1, 3 and 5. If no timeout occur then the next circle contains state 2, 4 and E.O.S, etcetera. The state names **com.error**, **E.O.S.** and **signal complete** will be further described in the following.

- **com.error**: If a timeout occur when being in either state 1, 3 or 5 the only transition is to the **com.error** state. Since this means that the signal has been high longer than the defined timeout, the only solution for this is a broken wire or a broken R/C-module

- **E.O.S.**: This is end of signal and there by also a timeout, because the next channel does not appear within the defined time.

- **signal complete**: Then the signal is complete the whole R/C-package is available and the transitions starts over.

Now the signal has been analyzed and the driver design can be carried out.

3.5.1 Overall design

The FSA is designed in such a way, that the input has to be correct, as described just in Figure 3.8 on the previous page. Also if the time between two edges is outside the interval $[1; 2]$ ms, the sample will be discarded and the error variable will be increased.

Figure 3.10 illustrates the FSA-driver for the R/C-module. When the robostix^{TM} gets an `INT7` it checks whether the input is correct or not, by checking the input port `INT7` to validate whether it is high or low when needed. The driver also consider whether the signal is within the a given time of two edges. If this is the case, the FSA switches state and await interrupts to gather the rest of the R/C package.
If the whole package has be accepted the error variable is reset and a flag is set indicating that a new package is ready, `RC_package_received`. When the error variable, `RC_suberror` reaches its max limit, `INT7` is disabled for a given time period, after this period a retry is initialized. The FSA is designed this way to accommodate error samples from the R/C-module which is not desired.

Figure 3.11 illustrates the sub FSA for when a new sample is available to the main program.

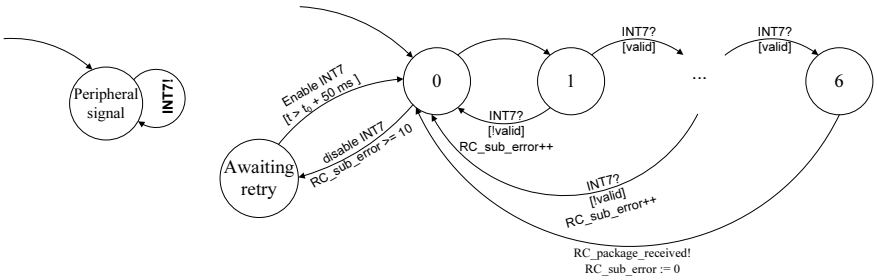

FIGURE 3.10: *The FSA for the R/C driver. Upon every transition, $t_0 = t$. `valid` means validating the signal according to time between to edges and check the input in order to assure the correctness of the signal. The FSA, to the left, is how the `INT7` interacts with the FSA to the right.*

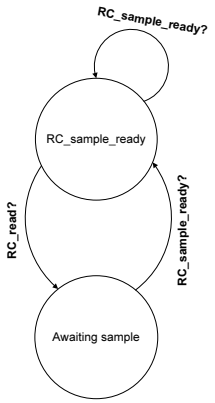

FIGURE 3.11: *The FSA for the availability of new samples in the R/C driver.*

3.5.2 Interface

In order to know which state the R/C FSA is in, the return argument and the R/C-channels to the main program, two enumerations and a structure have been created. These is listed below:

`RC_return_t`: This enumeration consist of `RC_com_error`, `RC_new_sample` and `RC_no_sample`. This is the possible return arguments to the main program to locate the state of the R/C-module.

`RC_state_t`: This enumeation consist of `RC_awaiting_sample`, `RC_incoming_sample` and `RC_awaiting_retry`. The internal possible states of the FSA.

`RC_chans_t`: This is a structure which contains the five R/C channels: `RC_roll`, `RC_pitch`, `RC_throttle`, `RC_yaw` and `RC_safety_switch`. This is the type that the they are returned in.

There are two functions which is used in order to get the R/C-driver working propperly, they are:

`void RC_init(void)`: This function is used to initialize the R/C-driver. It initializes `INT7` hardware and the structure to return the R/C-channels to the robostix^{TM}.

`RC_return_t RC_read_sensor(RC_chans_t *)`: It is the main function of the R/C-driver. The function is to be called from the main program in order to get samples from the R/C-module. The function takes one input argument, namely a pointer to the `RC_chans_t` structure.

`void RC_update_state(void)`: This function is used called when ever a sample is needed in order to assure that the state is up to date.

3.6 IMU INTERFACE

The IMU have the purpose of measuring how the X-Pro is moves during flight by measuring the angular velocity and linear acceleration. The measurements from the IMU are lead to the UART0 on the robostix^{TM}where an interrupt is generated every time a byte is received. The FSA design for the IMU is described in the following.

3.6.1 Overall design

The FSA for the IMU is designed such that if an error happens in a complete IMU packet the packet will be discarded and the driver will search for the beginning of the next packet. The errors that are accounted for are:

- A checksum for each packet the robostix^{TM} receives is calculated to verify that the packet is correct. If there is a mismatch in the checksum the packet is discarded.
- Timeout occurs if a successful packet has not been received within a maximum delay. This could indicate either several faults in the received packet, a loose wire, or the IMU is without power.

Since the IMU is of great importance for controlling of the X-Pro, both when using the R/C and in autonomous flight, the IMU is not discarded permanently although several errors are detected, in the hope that one packet can be transfered to the robostix^{TM} successfully or the connection can be reestablished. The bytes send out as parts of a packet can be considered the result of a Discrete Event System running on the IMU - Figure 3.12 illustrates an estimator for the system, based on the data sheet. Figure 3.13 illustrates how the robostix^{TM} indicates that there is a new sample ready from the IMU.

3.6.2 Interface

The IMU driver interface defines one type specific for the IMU driver:

`imu_return_t`: Is an enumeration of the type of values, `imu_new_sample`, `imu_no_sample` and `imu_comm_error`, which are the values that can be returned when the main system investigates if there are new imu samples.

`imu_sample_t`: This is a structure that contains the 14 bytes in the IMU packet, which is the type they are returned in.

Furthermore four functions are defined which can be called from the main system:

`imu_return_t imu_sample_ready(volatile imu_sample_t**)`: The function used for continuously asking the IMU driver for new samples.

`void imu_receive_timerrollover(void)`: This is used by the Timer/Counter0 overflow ISR to notify the IMU driver, that the timer has rolled over.

`void imu_hw_init(void)`: Initiates the Atmega128 hardware associated with the IMU driver.

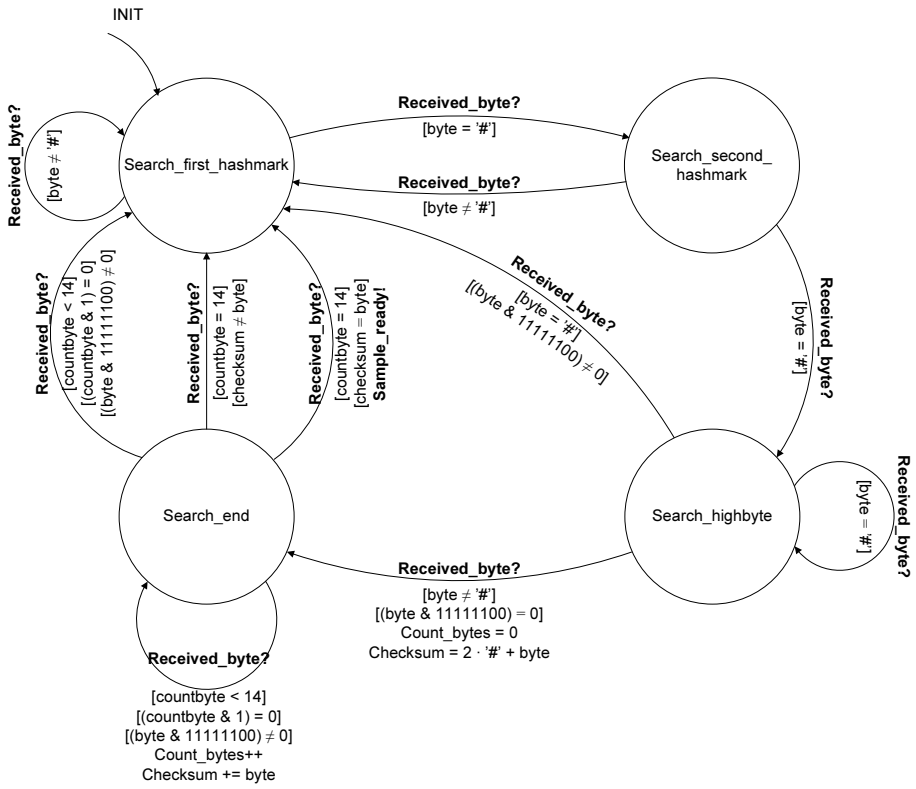

FIGURE 3.12: *The FSA for the IMU driver.*

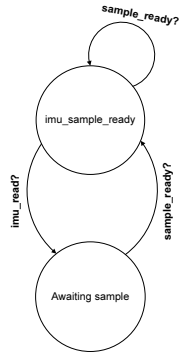

FIGURE 3.13: *The FSA for the availability of new samples.*

`void imu_update_state(void):` This should be called as frequent as possible, as it updates the state of the internal FSA.

WORKSHEET XIII: * GUMSTIX^{TM} AND ROBOSTIX^{TM} COMMUNICATION

STARTED: 20/11-06

ENDED: XX/12-06

This worksheet contains an analysis of the necessary communication between the gumstix^{TM} and the robostix^{TM}, which leads to requirements for the communication, followed by a design of a communication protocol.

CONTENTS OF WORKSHEET XIII

1 Analysis **131**

1.1 Data for transmission . . . 131

1.1.1 Sensor samples . . . 131

1.1.2 Motor control input . . . 132

1.2 Inter-Integrated Circuit (I^2C) bus connection . . . 132

1.2.1 Bus setup . . . 132

1.2.2 Bus load . . . 134

2 Design **137**

2.1 Package contents . . . 137

2.1.1 Package length . . . 137

2.1.2 Status indicators . . . 138

2.1.3 Checksum . . . 139

2.2 Complete package structure . . . 140

2.2.1 gumstix^{TM} to robostix^{TM} . . . 141

2.2.2 robostix^{TM} to gumstix^{TM} . . . 141

ANALYSIS 1

To control the Draganflyer X-Pro (X-Pro) during flight, it has previously been decided to divide this task between two processors mounted on individual boards, see Section 1.1 on page 16: A robostix^{TM} board handling most of the peripherical hardware, and a gumstix^{TM} board handling the heavier calculation, such as estimation and control, and the communication to a ground station. Hence it is necessary to transfer sensor samples from the robostix^{TM} to the gumstix^{TM}, and transfer calculated control input back to the robostix^{TM}. The interface available for communication is an I^2C bus connection, and its properties will also be shortly presented. Further information can be found in Chapter 2 on page 60.

1.1 DATA FOR TRANSMISSION

1.1.1 Sensor samples

The sensors sampled by the robostix^{TM} are listed in Table 1.1, along with information on their respective size of a sample and sampling frequency.

Sensor	Data-type	Size	Sample frequency
4× tachometer	4× `uint16_t`	4 × 2 bytes	[8 Hz, 30 Hz] for each
Inertial Measurement Unit (IMU)	14× `char`	14 bytes	100 Hz pr. package
3× Magnetometer	3× `uint16_t`	3 × 2 bytes	≈ 25-50 Hz [1] (sequentially)
Remote Control (R/C) status	—	(1 bit)	44.4 Hz
4× Motor input control (During other than gumstix^{TM}-pilot)	4× `uint16_t`	4 × 2 bytes	Unknown, max. 100 Hz

TABLE 1.1: *The information of the sensor connected to the robostix^{TM} board.*

Since it is possible to detect certain sensor errors on the robostix^{TM}, it should also be possible to transmit these informations to the gumstix^{TM}. This information has not been included in the data for the sensors listed in the table, except for the R/C sensor. This is because it depends on the properties of the communication protocol, which will be designed later in the process.

Sensor	Data-type	Size	Sample frequency
SRF08 Command	1× `char`	1 byte	15 Hz or higher
SRF08 Sample	3× `char`	3 bytes	Same as above

TABLE 1.2: *The information of the SRF08 sensor also connected to the I^2C bus.*

1.1.2 Motor control input

The information transmitted from the gumstix™ to the robostix™ is listed in Table 1.3.

Information	Data-type	Size	Sample frequency
4× Motor input control (During gumstix™-pilot)	4× `uint16_t`	4 × 2 bytes	Unknown, max. 100 Hz

TABLE 1.3: *The information of the sensor connected to the robostix™ board.*

There is no further communication from the gumstix™ to the robostix™.

1.2 I²C BUS CONNECTION

The available communication media between the gumstix™ and the robostix™ is an I²C bus, according to Subsection 2.2.1 on page 108. This is also connected to the sonic range finder SRF08, and a few devices used internally on the gumstix™ and robostix™ boards. Both robostix™ [Atm, p. 199] and gumstix™ [Gum06a] support an I²C clock frequency of 400 kbit/s [2], while only 100 kbit/s [3] is guaranteed by the SRF08. Consequently, the protocol design should be carried out with this bandwidth restraint. It should be noted, that these frequencies can be temporarely reduced if a node on the bus performs clock stretching [i2c00, p. 8]. 400 kbit/s gives a maximum of 50 kbytes/s. Furthermore, initiating a write to a node takes one byte for adressing the node, and initiating a read from a node takes one to three bytes on the same account, dependent on the protocol.

1.2.1 Bus setup

The I²C bus is a master/slave setup. Optionally, multiple masters can be present on the same bus, but with the risk of package collisions. The SRF08 is a slave, and can only be configured to be a master by its manufacturer, while both the robostix™ and the gumstix™ can be configured to be either master or slave. The master can write to a slave, and read from a slave, while the slave can not initiate a transmission. Since it is necessary to transmit data both ways between gumstix™ and robostix™, there are three options for the bus setup:

gumstix™ master / robostix™ slave setup: The gumstix™ writes motor dutycycles to the robostix™, and reads both the robostix™ and the SRF08 on a regular basis for new sensor data. Dependent on the protocol, this can introduce some overhead in

[2] This is defined as Fast-mode [i2c00, p. 8]

[3] This is defined as Standard mode [i2c00, p. 8]

the communication, since the robostix™ receives new sensor measurements from the IMU at up to 100 Hz, while other sensors are updated less frequently.

robostix™ master / gumstix™ slave setup: In this case, the robostix™ writes new sensor values as soon as they are sampled to the gumstix™, and reads new motor dutycycles on a regular basis. Since a slave cannot initiate a transmission, the robostix™ must also sample the SRF08 and relay the data to the gumstix™. The latter demands extra processing time on the robostix™, which has a much slower processor than the gumstix™ board. Secondly, the control frequency can not be altered without either recompiling the robostix software, or adding the feature to the communication protocol. Dependent on the chosen protocol, there can also be an overhead in the communication, since each package of sensor data must have a header specifying which sensor data is being transmitted. Furthermore, initiating many transmissions produces an overhead due to the I^2C protocol itself. A solution to this could be to collect data from several sensors before sending.

gumstix™ and robostix™ dual masters setup: This could be a compromise of the abovementioned, but introduces problems with collisions. Even though the I^2C bus standard defines a way to detect this, and defines which master should resign, there are still issues to be solved regarding retries, delays due to collisions, and how these delays affect estimation and control.

In either of the abovementioned cases, an issue is introduced with respect to the time of the sampling for each sensor: Delays will be present from sensors being sampled by the robostix™ until the samples can be transmitted to the gumstix™. In the robostix™ master / gumstix™ slave setup and the in the dual masters setup, congestion can occur on the I^2C bus when sensors are sampled at the same time. In the gumstix™ master / robostix™ slave setup, the gumstix™ collects the samples from the robostix™ on a regular basis, thereby also introducing delays. The magnitude of the delays depends on the chosen I^2C setup, as well as the possible solutions to it, such as timestamping samples.

Considering the benefits and drawbacks of the listed options, the gumstix™ master / robostix™ slave setup has been chosen as the appropriate setup. The most frequently updated sensor is the IMU at 100 Hz, so it has been decided to let the gumstix™ collect samples from the robostix™ at the same frequency. This might result in a drift over time, if either of the clocks on the gumstix™, robostix™ or IMU are not precise, or if the load on either of the processors change during runtime. This gives the potential risc of missing an IMU sample, but the risc is not considered significant.

The I²C bus clock frequency is set to 400 $^{\text{kbits}}/_{\text{s}}$, which results in a maximum of 500 bytes per $^{1}/_{100}$ s, including bytes for adressing. If 400 $^{\text{kbits}}/_{\text{s}}$ is too fast for the SRF08 to perform clock stretching, a configuration of 100 $^{\text{kbits}}/_{\text{s}}$ might be necessary, resulting in 125 bytes per $^{1}/_{100}$ s.

1.2.2 Bus load

If sampling of all sensors coincide with transmission of new motor duty cycles, the minimum load on the bus will be 32 bytes[4] during normal operation, excluding bytes for adressing, sensor status', and optional information such as checksums and timestamps with attendant time synchronization packages. If samples have been put in queue during a communication line fault, up to 36 bytes can be waiting for transmission, assuming that only the newest samples are in the queue. Additionally, traffic to and from the SRF08 should be taken into account, which is 3 bytes for writing the range command, and 6 bytes for reading the measurement, all included.

In Subsection 1.2.1 on the previous page, the motivation for including a timestamp is discussed. Some samples can be gathered into one package with it's own timestamp, since the data arrive simultaneously. Samples that would require an individual timestamp would be each tachometer sample, IMU sample package as one sample, each magnetometer axis, and motor input signals package as one sample. This gives a total of nine timestamps. The size of these timestamps depend on the resolution desired for an estimator running on the gumstix™. If each timestamp is one or two bytes long, the amount of data to be transmitted would be increased significantly, compared to the available bandwidth in I²C standard mode.

The maximum delay time, from the robostix™ takes a sample and the sample is collected by the gumstix™, is $\approx$ $^{1}/_{100}$ s, assuming no delays originate in processing time, see Figure 1.1 on the facing page. Samples from the magnetometer are a result of continuous measurement over a period of 20-40 ms. Similarly, a tachometer sample will only represent a rotor velocity averaged over the time from the first to the last pulse given by the hall sensors. Due to this averaging effect over more than $^{1}/_{100}$ s, a timestamp with a higher resolution than $^{1}/_{100}$ s, is not regarded a more accurate mark of the time at which the given measurement was conducted. A more accurate result might be obtained by modelling the averaging effect as a part of the sensormodels, or consider all samples to be delayed by $^{1}/_{100}$ s and take this into account when updating an estimate.

[4]Please note, that the three axes of the magnetometer are sampled sequentially, so a maximum of one sample will be available at any given time during normal operation.

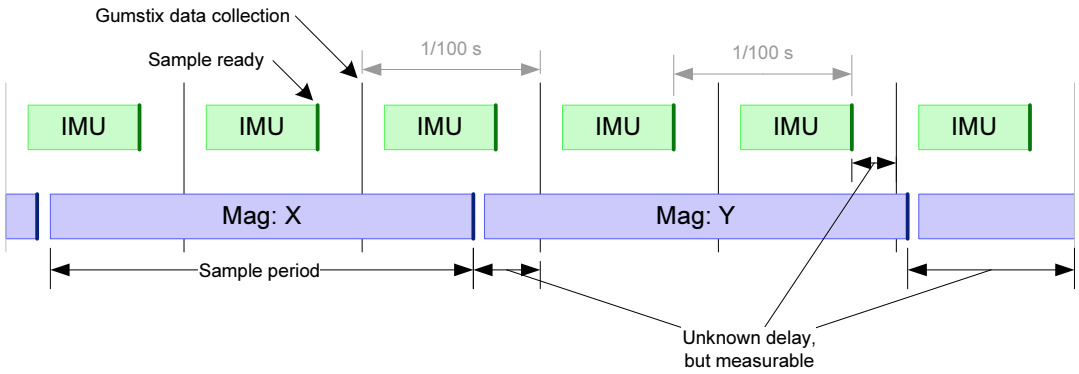

FIGURE 1.1: *Example of delays from the robostix^{TM} receives samples until they are collected by the gumstix^{TM}. Notice that the sample frequency of the IMU is the same as the frequency for collection of data, while the period for sampling the magnetic field exceeds the $^{1}/_{100}$ s.*

Table 1.4 on the next page provides an overview of the minimum expected traffic, based on the information known to be transmitted at this point of the process. The size of checksums and sensor status' has not yet been discussed, as it is part of the design of the protocol. Hence the checksum has been estimated to be at least 1 bit, and since each sensor can return one of three states, see Chapter 3 on page 112, it has been estimated that 2 bits per sensor could contain this information. The total load to be transmitted is estimated to at least 51 bytes per $^{1}/_{100}$ s, which is below the maximum 125 bytes per $^{1}/_{100}$ s in I^{2}C standard mode. In conclusion, it should be possible to transfer the minimum required amount of data over the I^{2}C bus, unless clock stretching performed by the robostix^{TM} or the SRF08 reduces the average bandwidth to under 50% of standard mode bandwidth.

Message	Contents	gumstix™ pilot	Not gumstix™ pilot
Read sensorinformation from robostix™	I²C read header	≥ 1 bytes	≥ 1 bytes
	robostix™ status	2 bits	2 bits
	Sensor status	$\approx 2\,^{\text{bits}}/_{\text{sensor}} = 18$ bits	≈ 20 bits
	Sensor samples	28 bytes	28 bytes
	Motor input signals	0 bytes	8 bytes
	Checksum	≥ 1 bit	≥ 1 bit
	Subtotal	$\geq$ **32** bytes	$\geq$ **40** bytes
Write motor duty cycles to robostix™	I²C write header	1 byte	0 byte
	Motor input signals	8 bytes	0 bytes
	Checksum	≥ 1 bit	0 bit
	Subtotal	$\geq$ **10** bytes	**0** bytes
Read range from SRF08	I²C read header	3 bytes	3 bytes
	Sensor sample	3 bytes	3 bytes
	Subtotal	**6** bytes	**6** bytes
Write range command to SRF08	I²C write header	2 byte	2 byte
	Command	1 byte	1 byte
	Subtotal	**3** bytes	**3** bytes
Total		$\geq$ **51** bytes	$\geq$ **49** bytes

TABLE 1.4: *Preliminary overview of the minimum load, which can occur on the I²C bus within $^1/_{100}$ s.*

DESIGN 2

The analysis has already taken some decisions on the design, in order to thoroughly analyse the oppoprtunities and restrictions when using the I²C bus for communication between the gumstix™ and robostix™. This chapter will investigate potential package structures for containing the information, and will consider different methods for calculating checksums and storing sensor status information. At the end, the final packages for transmitting the relevant data are presented.

2.1 PACKAGE CONTENTS

2.1.1 Package length

Since each sensor and actuator has an individual sampling frequency according to Tables 1.1, 1.2 and 1.3, there is reason to consider different possibilities for structuring the packages. The following options have been considered:

One fixed length package of samples: Even though it is only the IMU which supplies a new sample at the same frequency as the robostix™ is being polled, it could prove advantageous to use a fixed length package with a standard structure. An advantage could be, that no indicators are needed for describing the structure of a package, but an indication of which samples are new / not blank is needed. A disadvantage is, that most packages will contain one or several blank spaces, when no new sample has been taken for a given sensor - this gives unnecessary traffic, thereby prolonging the times for transmission. The length of such a package would be similar to the calculated in Row Read sensorinformation in Table 1.4 on the preceding page

One variable length package of samples: Another possibility is minimizing the package length by only including new samples. This reduces unnecessary traffic, which is beneficial. Unfortunately, the I²C Application Program Interface (API) supplied by the manufacturer of the gumstix™ does not support requests for variable length packages. Apart from this, the packages would also have to include information on which samples are contained in the current package, which furtherly consumes bandwidth.

Due to the fact, that an API for the I²C bus has already been developed and tested by the manufacturer of the gumstix™, it is desirable to use this. Since it does not support requests for variable length packages, it has been chosen to use packages of fixed lengths.

2.1.2 Status indicators

Apart from the dissimilar sample frequencies, the status of all sensors must be transmitted as well as the status of the robostix^{TM}. With a few exceptions, the possible states are common for most sensors:

Tachometers: Each tachometer can return either new sample, error or no sample.

IMU: An IMU package is regarded as one sample, and can return either new sample, error or no sample.

Magnetometer: Each axis is sampled using the same sensor. Due to this, each axis of the magnetometer can return new sample or no sample, while the magnetometer itself can be either ok or error.

R/C: Samples from the R/C module are not forwarded to the gumstix^{TM}. Consequently, the R/C module will only report ok or error.

Motor input control: The motor input is only relevant when either the robostix^{TM} or a human is the pilot of the X-Pro. Since the values are calculated internally, an error state is not possible. Only new sample or no sample are regarded relevant.

robostix^{TM} status: According to Section 1.1.2 on page 105, the robostix^{TM} will be in one of four states: Idle, Human pilot, robostix^{TM} pilot and gumstix^{TM} pilot. This, along with the status of all other sensors, is regarded sufficient information on the status of the robostix^{TM}.

For those sensors with two or four states, the information can be stored in one or two bits respectively. This embodies the magnetometer status, magnetometer axes' status, R/C, Motor input control and robostix^{TM} status, which sums up to one byte. Since the remaining sensor status' is based on one of three states, there are several different ways for storing this information. Two methods are considered:

- One bit for sensor status, one for sample status. This would give a total of 10 bits for the remaining five sensors, which would take up 2 bytes in the package. This gives some blank space in the second byte, which can be used for other purposes. The bytes are encoded and decoded using binary `OR` and `AND` operations respectively, which are usually present in processors' instruction set. A disadvantage is, that the data is not very compact, since a sensor will not return both error and new sample at the same time. Oppositely, this fact can be used for error detection in the communication.

- Store information in radix 3. Since there are three states for each sensor, there is a possibility of storing the information in a number with base 3. Within one byte, 5 three-state bits can now be stored, since:

$$\left\lfloor \frac{\log 256}{\log 3} \right\rfloor = 5 \tag{2.1}$$

Encoding the data on the robostix™ would be done by:

$$[0,2] \cdot 3^4 + [0,2] \cdot 3^3 + \ldots + [0,2] \cdot 3^0 \tag{2.2}$$

Decoding the data on the gumstix™ could be done by:

$$S(n) = \begin{cases} \left\lfloor \frac{\text{byte} - \sum_{i=n+1}^{5} \left(S(i) \cdot 3^i\right)}{3^n} \right\rfloor & 0 \leq n < 5 \\ 0 & n \geq 5 \end{cases} \tag{2.3}$$

... which could advantageously be implemented as an algorithm starting with the highest three-state bit.

It has been chosen to use the two bit representation of the data, since it contains the possibility for error detection, and since the remaining space on the second byte can be used for part of a checksum of the entire package.

2.1.3 Checksum

A checksum is included in packages between robostix™ and gumstix™, enabling the gumstix™ to detect data corruption occuring in the transmission, if any. Two algorithms are considered:

Sum of bytes: All bytes are summed, and the result is truncated to a given bit length. The bit length should be dimensioned based on the length of the package. A bit length of 1 bit, as assumed in Table 1.4 on page 136, will only detect an odd number of error due to the truncation. If using e.g. 1 byte, it will take 256 errors in the same transmission before the checksum truncates to the same value. Unfortunately, some errors can not be detected: If one byte is received with a higher value than the transmitted byte, and another byte is received with an equally lower value than the original, the packages bytes will sum up to the enclosed checksum.

Fletcher's checksum: For the string D, the checksum is based on the following formulas [UB90, App. I]:

$$A = \sum_{i=1}^{n} D[i] \tag{2.4a}$$

$$B = \sum_{i=1}^{n} ((n - i) \cdot D[i]) \tag{2.4b}$$

This can be implemented as:

```
for (i = 1; i <= n; i++)
{
  A += D[i];
  B += A;
}
```

Results are then truncated to a given bit length. The advantage of such a checksum is increased integrity. A drawback is the also increased calculation complexity.

Other algorithms exist with higher data integrity, but these have been disregarded in order to limit the scope of the project, and since their calculation complexity is higher, which is undesirable. The Fletcher's checksum is chosen due to its increased data integrity compared to a regular sum of the bytes, while retaining a relatively undemanding algorithm compared to other methods.

For packages from the robostix™ to the gumstix™, the storage of sensor status' has left 6 bits to be used for other purposes, such as A truncated to 6 bits. Since the sensor status' will be placed in the start of the package, it would be necessary to calculate the checksum in advance when the robostix™ is contacted. This can be a demanding task, compared to summing up the checksum as the bytes are sent, and include it as the final two bytes. Due to this, it has been chosen to use one byte for A and one for B, placed at the end of the package.

For packages from the gumstix™ to the robostix™, one byte is used for A and one for B.

2.2 COMPLETE PACKAGE STRUCTURE

The following two subsections describes the final structure of packages sent between robostix™ and gumstix™. The initiation of a transmission is not included, since this is handled by the I^2C hardware. Even though the robostix™ [AVR] is little endian, 16 bit values are transmitted in big endian format, which is also common for all sensors used in the project.

2.2.1 gumstix™ to robostix™

Table 2.1 describe the package structure for transmission from the gumstix™ to the robostix™. The abbreviations used are:

$^{MOT}DC_F$, $^{MOT}DC_L$, $^{MOT}DC_B$, $^{MOT}DC_R$: The dutycycles for the front, left, back and right motor respectively.

^{CS}A, ^{CS}B: The checksum variables A and B, which are both unsigned 8 bit integers.

Byte	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Contents	$^{MOT}DC_F$		$^{MOT}DC_L$		$^{MOT}DC_B$		$^{MOT}DC_R$		^{CS}A	^{CS}B

TABLE 2.1: *Package structure for transmission of motor input control signals from the gumstix™ to the robostix™*

The size of this package sums up to 11 bytes, include initiation of transmission.

2.2.2 robostix™ to gumstix™

Table 2.2 on the next page gives a description of the package, which is received by the gumstix™, when it requests the robostix™ for sensor samples. The package is structured with status bytes first, then samples and the checksum last.

The abbreviations used are:

ST_1: This is the first status byte, which uses two bits for the robostix™ state, and one bit each for R/C status, calculated motor input new sample, magnetometer status, and magnetometer axes x, y and z new sample.

Bit	7	6	5	4	3	2	1	0
Contents	$^{RB}ST_1$	$^{RB}ST_2$	^{RC}ST	^{MOT}NS	^{MAG}ST	$^{MAG}NS_X$	$^{MAG}NS_Y$	$^{MAG}NS_Z$

The robostix™ states $^{RB}ST_1$ and $^{RB}ST_2$ are defined in the following table:

$^{RB}ST_1$	$^{RB}ST_2$	State
0	0	Idle
0	1	Human pilot
1	0	gumstix™ pilot
1	1	robostix™ pilot

ST_2: The second status byte contains information on the tachometers:

Bit	7	6	5	4	3	2	1	0
Contents	$^{TM}ST_F$	$^{TM}NS_F$	$^{TM}ST_L$	$^{TM}NS_L$	$^{TM}ST_B$	$^{TM}NS_B$	$^{TM}ST_R$	$^{TM}NS_R$

ST_3: The third byte gives information of the known status of the IMU and indicates if an IMU package has been received:

Bit	7	6	5	4	3	2	1	0
Contents	${}^{IMU}ST$	${}^{IMU}NS$	0	0	0	0	0	0

${}^{IMU}SA$: Package received from IMU, without header and own checksum.

${}^{MAG}SA_X$, ${}^{MAG}SA_Y$, ${}^{MAG}SA_Z$: Magnetometer samples for the x, y and z axis respectively.

${}^{TM}SA_F$, ${}^{TM}SA_L$, ${}^{TM}SA_B$, ${}^{TM}SA_R$: Tachometer samples for the front, left, back and right rotor respectively.

${}^{MOT}CA_F$, ${}^{MOT}CA_L$, ${}^{MOT}CA_B$, ${}^{MOT}CA_R$: Calculated duty cycles for the front, left, back and right motor respectively.

${}^{CS}A$, ${}^{CS}B$: Checksum variables A and B of the Fletcher checksum.

Byte	**0**	**1**	**2**
Contents	ST_1	ST_2	ST_3

Byte	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Contents	${}^{IMU}SA$													

Byte	**17**	**18**	**19**	**20**	**21**	**22**
Contents	${}^{MAG}SA_X$		${}^{MAG}SA_Y$		${}^{MAG}SA_Z$	

Byte	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
Contents	${}^{TM}SA_F$		${}^{TM}SA_L$		${}^{TM}SA_B$		${}^{TM}SA_R$	

Byte	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**
Contents	${}^{MOT}CA_F$		${}^{MOT}CA_L$		${}^{MOT}CA_B$		${}^{MOT}CA_R$	

Byte	**39**	**40**
Contents	${}^{CS}A$	${}^{CS}B$

TABLE 2.2: *Package structure for transmission of motor input control signals and sensor samples from the robostix™ to the gumstix™*

WORKSHEET XIV: * PRINTED CIRCUIT BOARD (PCB) DESIGN ON THE DRAGANFLYER X-PRO (X-PRO)

14/12-2006

A description of how the PCB layout has been designed.

CONTENTS OF WORKSHEET XIV

1 PCB design on the X-Pro **144**

1.1 Hardware driver design 144

1.2 PCB design and layout 144

1.3 Additions made to the PCB 148

PCB DESIGN ON THE X-PRO

1

The chapter will first give an overview of the hardware design for each sensor and then lead to the final PCB design and some improvements that has been made to it.

1.1 HARDWARE DRIVER DESIGN

The hardware driver for the power drives has been designed and tested in Chapter 3 on page 34 and its drivers are illustrated in the top left side of Figure 1.1. The rest of the hardware designs are standard designs which the produces of the components and sensors have given in the data sheets. These designs are also illustrated in Figure 1.1. The placement of each of the components on the final PCB can be seen by using the names of on the component and then transfer that to Figure 1.2 and Figure 1.3. The PCB is described in the next section.

1.2 PCB DESIGN AND LAYOUT

The shape of the PCB have been chosen to be similar to the top platform of the base of the X-Pro as it came form the producer. This design is chosen so that the old protection cap from the old design can be used again. The PCB are designed such that the positive supply is on the outer ring of the upper ring as depicted on Figure 1.2 and the negative supply is placed the same way on the other side of the PCB shown in figure 1.3. The supply to the X-Pro is constructed in this way to minimize the Electro Magnetic Interference produced by changing current consumption. In addition to this the robostix^{TM}, gumstix^{TM}, Global Positioning System (GPS) and Inertial Measurement Unit (IMU) modules are placed as far possible from the supply, which in this case are in the center of the PCB. This design also makes it possible to place the IMU close to the Center of Mass (CM) of the X-Pro, which is the optimal placement of it.

Again to minimize the potentially Electro Magnetic Interference (EMI) that the motor drives induces they are placed close to the motors but still on the same PCB. It is vital that the is broad PCB wires going to the power drives (mostly to the Field Effect Transistor (FET) and to the motor) since the largest currents are going through these points.

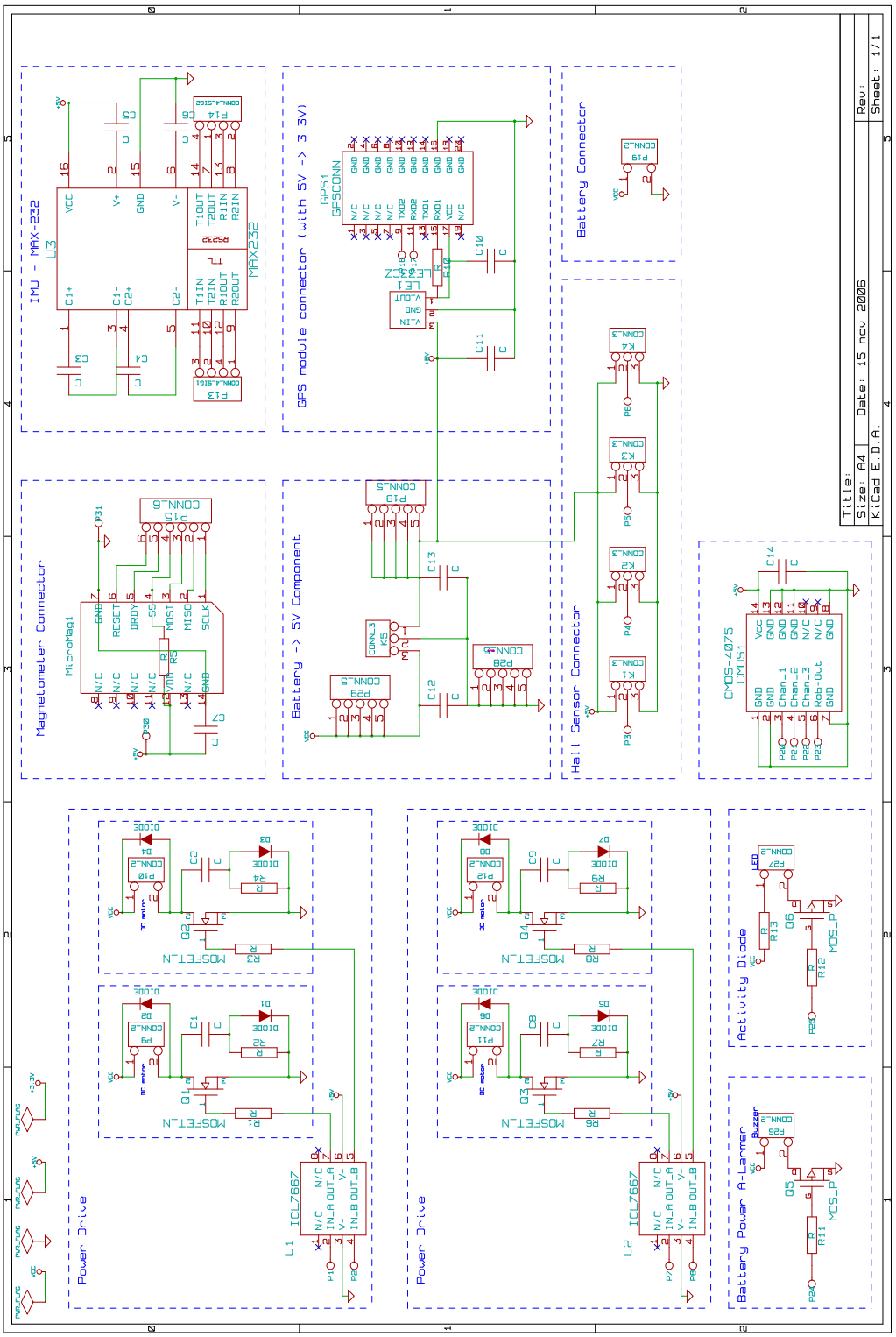

FIGURE 1.1: *Overview of the design of all hardware drivers for the X-Pro.*

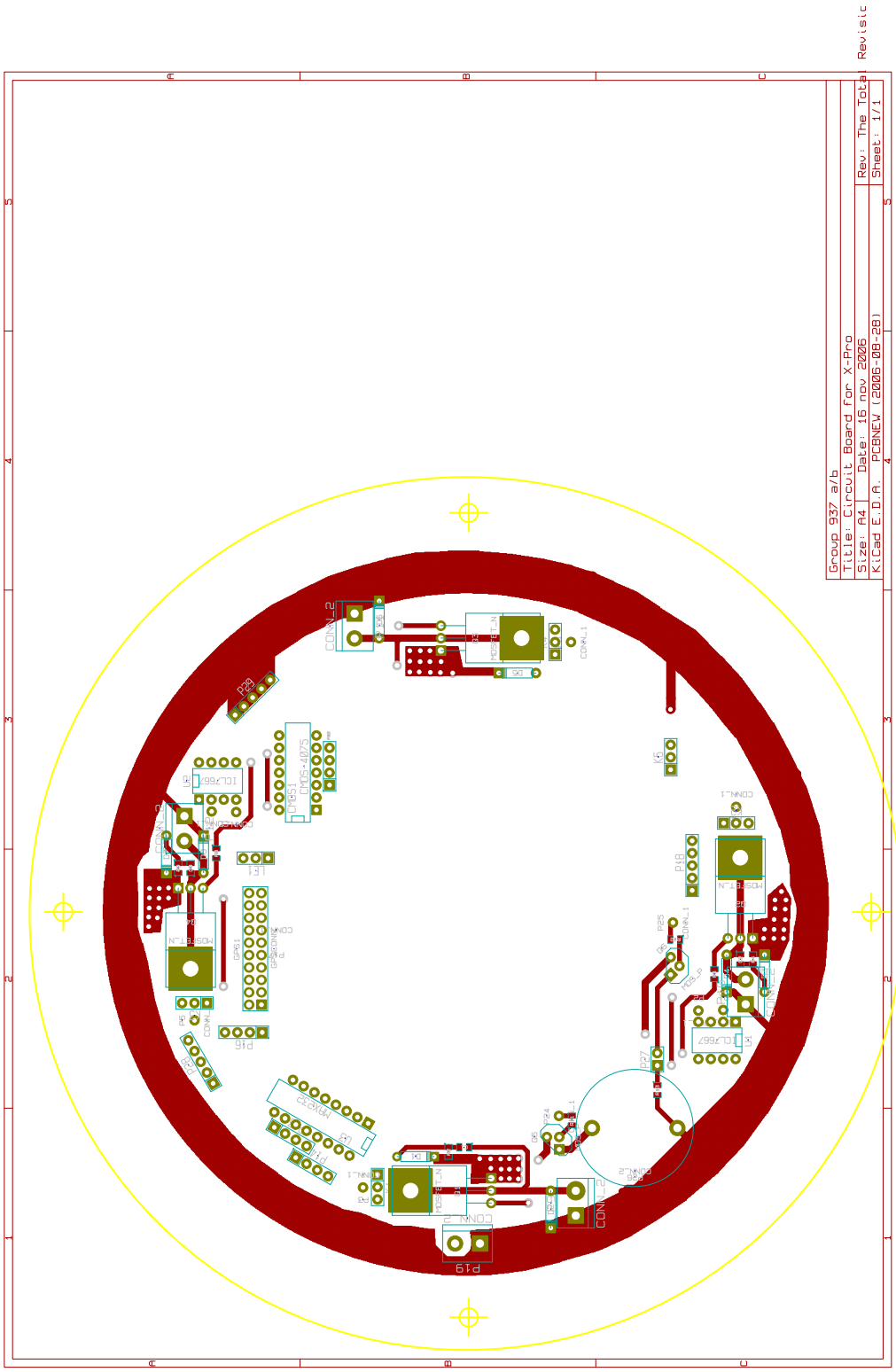

FIGURE 1.2: *The PCB top-layer seen from above where most components are mounted.*

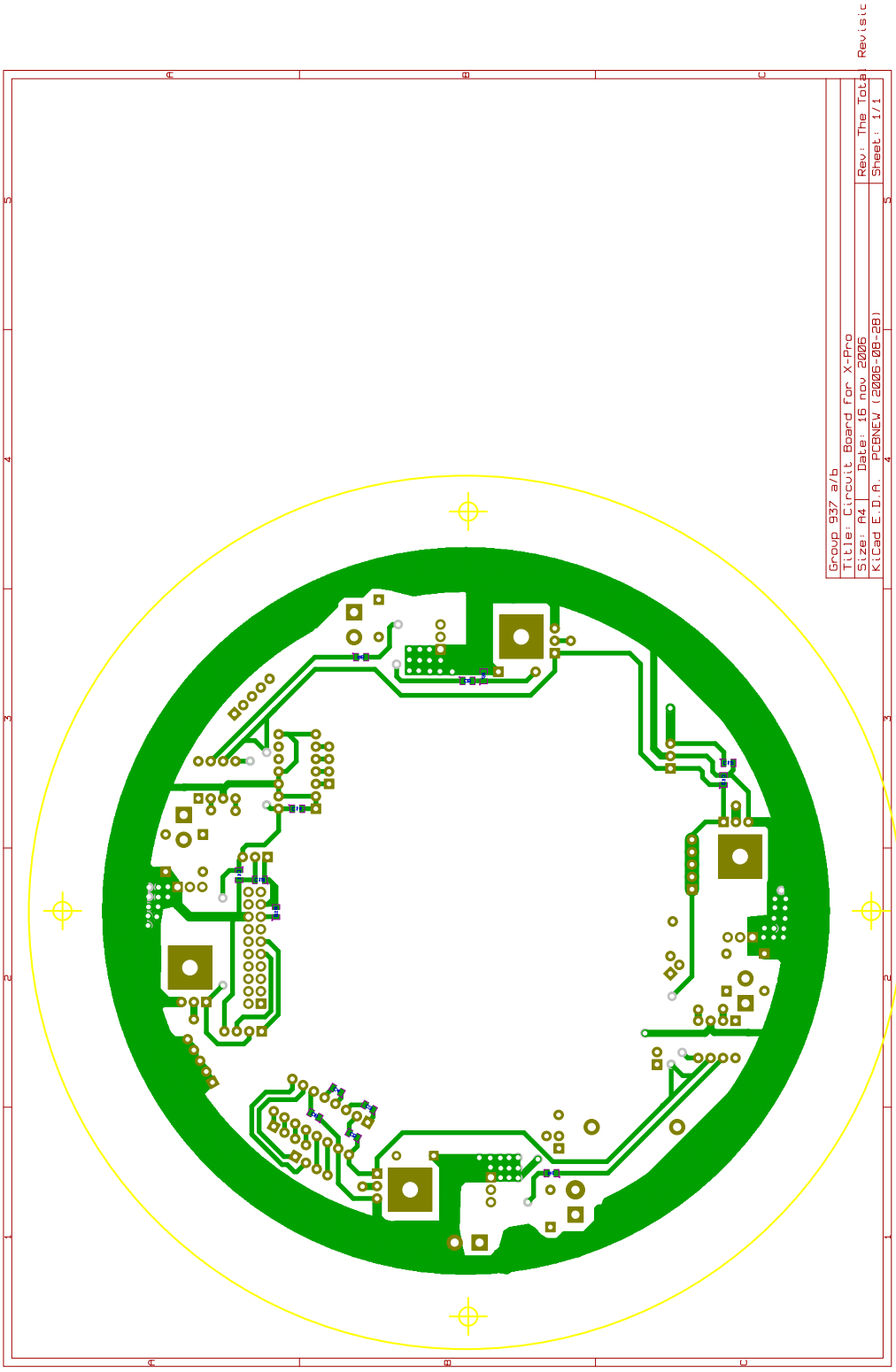

FIGURE 1.3: *The PCB bottom-layer seen from above.*

1.3 ADDITIONS MADE TO THE PCB

When designing the PCB all components were made as Surface Mounted Devices (SMD) components but because of problems with obtaining the desired rating of the SMD capacitors for decoupling odd-shaped components had to be inserted instead. In the power drives the SMD components were discarded because of the high current lead through the drives again odd-shaped components were used because they were rated to a higher current. The GPS module is running on a $3.3\,\mathrm{V}$ supply but the logic levels of the gumstix^{TM}STUART TX line is $5\,\mathrm{V}$ so a voltage divider is put added (resistor values of $3k\Omega$ to GND and $2k\Omega$ SMD on the TX line). Further more pull up resistors are placed on the input to all four PWM drivers, to make sure that motors does not run before Robostix board is initialized.

WORKSHEET XV: * CONSIDERATIONS ON EMI AND EMC AND CHANGES TO DESIGN

12/12-06

This worksheet is documentation the changes made to the Draganflyer X-Pro (X-Pro) to accommodate the noise mainly induced by the four power drives and the fluctuating current consumption by digital components.

CONTENTS OF WORKSHEET XV

1 Problem areas **150**

2 The accommodations **151**

PROBLEM AREAS 1

The group had some worries on the design of the PCB, thus a contact to Hans Ebert, from Department of Communication Technology. His general opinion was that our original design was not perfect, although many traps had been avoided. In the following a list of problems areas will be presented which can be minimized by relatively simple means. The topic described are the outcome of a discussion with Hans Ebert.

1. Poor decoupling of the $14.8\,\mathrm{V}$ power supply will mean that noise can superimpose a voltage on top of the power supply. This could lead to noncontinuous operations, and cause destruction of $5\,\mathrm{V}$ power supply or even worse the Lithium-Polymer (Li-Po) battery could get damaging voltage spikes back into the battery.

2. Poor decoupling of the digital $5\,\mathrm{V}$ power supply. A quick burst from Wifistix could cause a local voltage drop on the power supply to this module, which could case an unintended reset of the Gumstix or the Robostix.

3. The separation of the GND and $5\,\mathrm{V}$ connectors. These connectors are put on opposite ends of the PCB. Changes in the current consumption by the Robostix and the Wifistix will introduce a magnetic disturbance to others, even more it is possible that peak voltages can be superimposed on the wires to the Robostix and the Wifistix.

4. Power wires from the PCB to the motors are acting like large antennas, emitting noise due to the $300\,\mathrm{Hz}$ Pulse Width Modulation (PWM) signal.

5. Noise emitted from the power drives. The PWM signal causes the opening and closing of the driving MOSFET - transistor, this will without a doubt produce some noise but to what extend this noise will be damaging to the digital signals near by is not known.

THE ACCOMMODATIONS 2

The following accommodations are carried out in accordance with Hans Ebert to the extend that the solutions where feasible. To do a new print layout is not considered as feasible at the current project phase. On Figure 2.1 on the next page improvements are shown.

1. Poor decoupling of the $14.8\,\mathrm{V}$ power supply: The battery VCC and GND are placed in circular PCB tracks on the top side and on the bottom side respectively. Eight ceramic $100\,\mathrm{nF}$ decoupling capacitors are placed between VCC and GND around the whole board. Where the battery wire connects to the PCB a relatively large electrolytic capacitor of $100\ \mu F$ is placed to even out spikes that could harm the battery.

2. Poor decoupling of the digital $5\,\mathrm{V}$ power supply: As near to the Wifistix and the Robostix, that is in the DC-connectors two sets of decoupling capacitors are placed. A $22\ \mu F$ electrolytic capacitor and a $100\,\mathrm{nF}$ ceramic capacitor. The two capacitors will even out noise over a wide frequency spectrum. The capacitors can only even out the influence of the noise, but with the use of the ferrite beads on the power supply wires to the Wifistix and the Robostix, the energy in the high frequency noise is absorbed. This is due to the key characteristic namely the hysteresis of the ferrite. I the setup they will act as a low pass filter although there is no cutting in the supply wires. With these two components the noise emitted back out of the Wifistix and Robostix are kept to a minimum.

3. The separation of the GND and $5\,\mathrm{V}$ connectors: The simple solution is to remove the problem. The GND connector is moved to other end of the board, near the $5\,\mathrm{V}$ which make it possible to twist the GND and +5V wires. This will reduce the amplitude of the magnetic field due top the changes in current consumption. Further more the immunity to noisy environments are increased cause of the twisting of the wires.

4. Motor wires: The control signals to the DC-motors are not only a DC voltage. There is also a fair amount of higher frequencies on these wires. Again ferrite beads is put on the wire to a low pass filtering with dissipation.

5. Noise emitted from the power drives: The solution for this problem is to shield the power drives, reducing the electro magnetic noise to the surroundings. The

shield should be without gabs and should be connected to GND. Hans Ebert did not evaluate that noise could be directly damaging to any of the components, thus it is chosen to test the influence of the noise on a running system, before further noise accommodations are taken.

The additions 1 to 4 have all been carried out and can be seen on Figure 2.1. None of the additions should add negative side effects besides add to the total weight of the X-Pro.

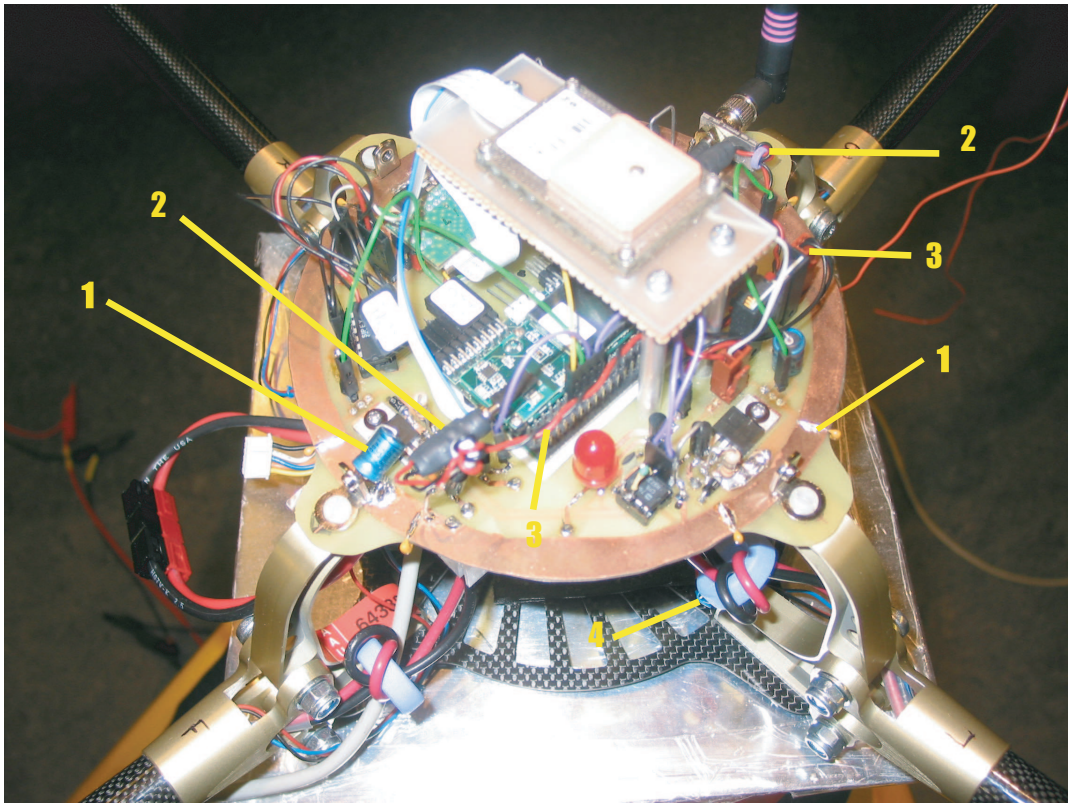

FIGURE 2.1: *Picture showing the improvements to the design. The numbers are corresponding to the enumerations in the text.*

WORKSHEET XVI: DESIGN OF MANUAL CONTROL

18/12-2006

The Draganflyer X-Pro (X-Pro) is controlled via Remote Control (R/C) with help of four different controlling signals: Throttle, Yaw, Pitch and Roll. The Throttle signal is comparable to a gas pedal, meanwhile the other signals are steering signals.
The steering signals describe a rotation about each of the three main, locally X-Pro, axes (z,y,x). To accomplish a rotation about any of those axes, the rotors have to behave in a particular manner relative to each other. This means that the controlling signals have to be interpreted to appropriate current flow to each rotor with a help of an algorithm, which is the main topic of this chapter.

CONTENTS OF WORKSHEET XVI

1 Design of manual control **154**

1.1 Incoming signals 154

1.2 Mapping of steering signals to rotor Thrust 155

1.2.1 Single axis manipulation 155

1.2.2 Multiple axis manipulation 157

1.3 Range of T_{max} 159

1.4 Block Diagram of system 160

DESIGN OF MANUAL CONTROL 1

1.1 INCOMING SIGNALS

Incoming signals from the R/C are the steering signals Yaw, Pitch, Roll and the amplitude signal Throttle, these are illustrated in Figure 1.1.

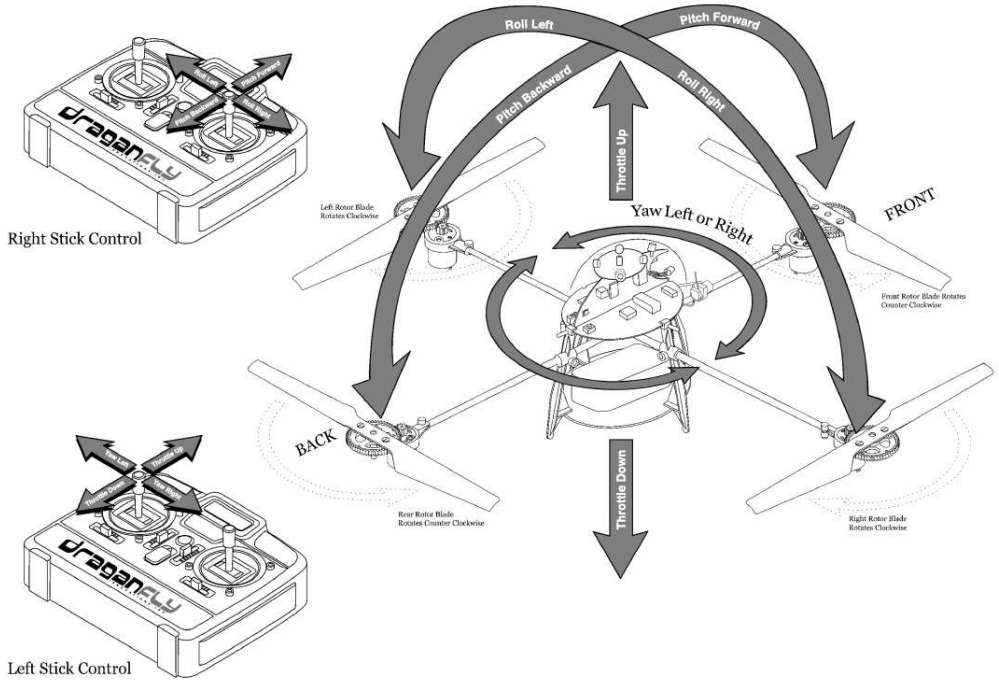

FIGURE 1.1: *The X-Pro response to the R/C as illustrated by the producer. [inc]*

The incoming signals from the R/C are scaled in terms of their duration, 0 to 1 ms, and must be adjusted according to their working area. The working area of the incoming signals is defined as follows. Throttle: $T\epsilon[0,1]$, Yaw: $Y\epsilon[-0.5,0.5]$, Pitch: $P\epsilon[-0.5,0.5]$ and Roll: $R\epsilon[-0.5,0.5]$. The range of the R/C is illustrated in Figure 1.2

This adjustment of the working area of the incoming signals is implemented in the following section where each rotor is been estimated relative to Throttle and appropriate steering signals.

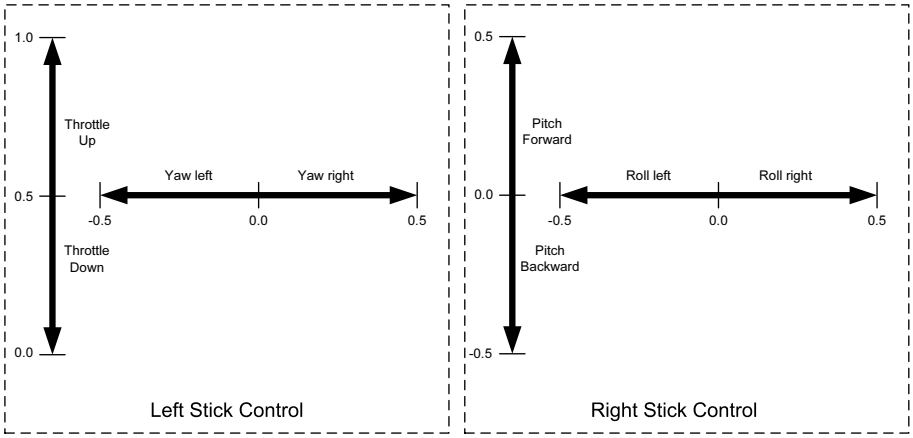

FIGURE 1.2: *The range of the R/C sticks.*

1.2 MAPPING OF STEERING SIGNALS TO ROTOR THRUST

The input is known and it indicates "what" needs to be done, the "how" factor has to be designed in form of a mapping (from-"what"-to-"how").

It has to be made sure of that the same total Lift is maintained when the Throttle control is kept constant, although the steering signals are changed in manoeuver. A total angular momentum in both magnitude and direction about an axis is conserved, as long as all external torques acting on the system is zero. This is made sure of, by internally "balancing" the appropriate rotor pairs about the current Throttle amplitude value $T_B \epsilon T$. The arriving steering signals need to be translated to appropriate motor signals. T_B is the point the motors balance about, when the steering signals are adjusted.

1.2.1 Single axis manipulation

Figure 1.3 shows how the front and back motors are balanced for the pitch signal for two cases:

1. This case has ”Nose-up-Tail-down”. The pitch value has to be negative, where where $B_1 < T_B$ on the left side and $F_1 > T_B$ equally on the right side.

2. This case has ”Nose-down-Tail-up”. The pitch value has to be positive, where $F_2 < T_B$ on the left side and $B_2 > T_B$ equally on the right side.

The Figure 1.3 shows two examples where the slope has been defined as positive at all times, which will be used from now on. The control signals for the back and front motors are calculated by including the pitch and the slope value $\alpha(T_B)$ (relative to T_B) as shown

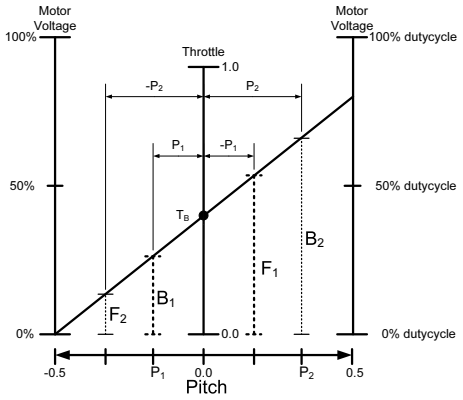

FIGURE 1.3: *The Thrust of the rotors balanced about the Throttle amplitude value.*

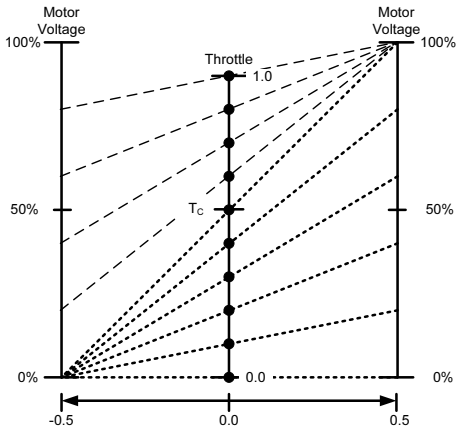

FIGURE 1.4: *An algorithm, graphically illustrates how to keep motor voltages balanced and within range, by dividing it about its balance point, at 50% motor voltage.*

in following algorithm:

$$B_P = T_B + P \cdot \alpha(T_B)$$
$$F_P = T_B - P \cdot \alpha(T_B)$$

The pitch mapping algorithm does not have to be changed much to be used as algorithm for roll and yaw mapping as well.

It has to be made sure of that the motor voltages do not get out of range $[0\%, 100\%]$. This is taken into account, where the range of the motor voltages is adjusted when approaching the limits of the working area. Figure 1.4 shows that the maximum Throttle signal is scaled lower than the maximum motor voltage, which is loosely explained in section 1.3. It also illustrates how to estimate the motor voltage. The motor voltage is estimated differently on either side of the T_C (the crossover point), which is defined 50% relative to the motor voltage maximum. The idea is illustrated where the slope value $\alpha(T_B)$ is defined as follows:

$$\alpha(T_B) = \begin{cases} 2 \cdot (T_B - 0) & \text{if } T_B \leq T_C \\ 2 \cdot (1 - T_B) & \text{Otherwise} \end{cases}$$

Rotor	Roll	Yaw	Pitch
F:		$+Y$	$-P$
L:	$+R$	$-Y$	
B:		$+Y$	$+P$
R:	$-R$	$-Y$	

TABLE 1.1: *Relation between angular rotation and motors applied.*

Table 1.1 shows the relation between angular rotation and what motors should be used to perform these. The table shows that each rotor can be influenced by two different angular rotations which needs to be taken into consideration when estimating the influence on each of the rotors.

1.2.2 Multiple axis manipulation

The R/C allows the pilot to activate all rotations at the same time and that needs to be taken into consideration. To ensure that all of the rotations stay within their maximum

working area at all times, these need to be weighed against each other.

This is a balancing problem can be visualized as a weight where: Yaw is balanced about Throttle, based on Roll and Pitch. Roll and Pitch are then balanced about each endpoints of Yaw. It has to be ensured that non of the endpoints cross outside the borders of the working area. The ide of this algorithm is illustrated in Figure 1.5.

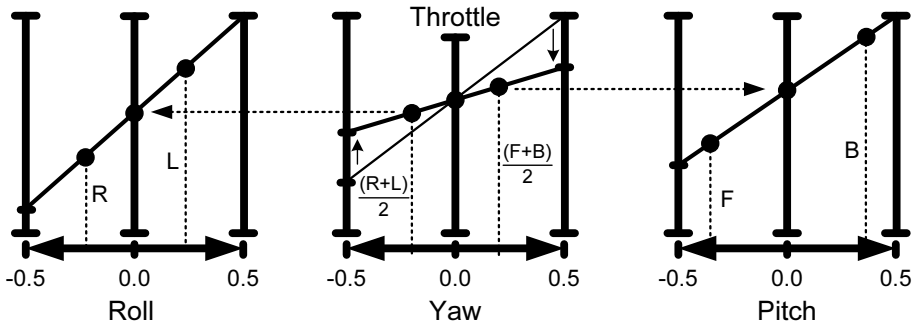

FIGURE 1.5: *The balancing problem: The slope $\alpha(T_B)$ scaled to ensure that each rotation can be controlled fully within previously defined working area.*

The functions mentioned in Table 1.1 are described by the following algorithm, where the weight balancing of the steering signals is calculated:

1. Throttle is scaled, to ensure the possibility of further manipulation in maximum throttle:

$$T_{scaled} = T_B \cdot T_{max} \tag{1.1}$$

2. The maximum slope for yaw α_Y is calculated based on the respective algorithm:

$$\alpha(x) = \begin{cases} 2 \cdot (x - 0) & \text{if } x \leq 0.5 \\ 2 \cdot (1 - x) & \text{Otherwise} \end{cases} \tag{1.2}$$

   The calculation is based on T_{scaled}.

$$\alpha_Y = \alpha(T_{scaled}) \tag{1.3}$$

3. The maximum slope for yaw is scaled, based on whether roll and/or pitch are manipulated:

$$Y_{scale} = \min\left(\frac{|Y|}{|P| + |Y|}, \frac{|Y|}{|R| + |Y|}\right) = \frac{|Y|}{\max(|P|, |K|) + |Y|} \tag{1.4}$$

4. The yaw and throttle contributions for R & L and F & B motors are then calculated as:

$$T_{F+B} = \alpha_Y \cdot Y_{scale} \cdot Y + T_{scaled} \tag{1.5a}$$

$$T_{R+L} = -\alpha_Y \cdot Y_{scale} \cdot Y + T_{scaled} \tag{1.5b}$$

5. The maximum slopes for pitch and roll are then calculated as:

$$\alpha_P = \alpha\left(T_{F+B}\right) \tag{1.6a}$$

$$\alpha_R = \alpha\left(T_{R+L}\right) \tag{1.6b}$$

6. Finally, the motor voltages / duty cycles can be calculated:

$$B = \alpha_P \cdot P + T_{F+B} \tag{1.7a}$$

$$F = -\alpha_P \cdot P + T_{F+B} \tag{1.7b}$$

$$L = \alpha_R \cdot R + T_{R+L} \tag{1.7c}$$

$$R = -\alpha_R \cdot R + T_{R+L} \tag{1.7d}$$

1.3 RANGE OF T_{max}

There is a need to estimate if the X-Pro is capable of taking off to flight and in that matter to break down speed before landing. The working range of T_{max} under these working conditions is a key-element in this matter.

The lower limit T_{max_lo} gives the idea when hover takes place and the higher limit T_{max_hi} is the maximum permitted Throttle for the pilot to apply. In the neighborhood of T_{max_hi}, the steering signals are scaled to their minimum in order to preserve the same total Thrust and moments.

T_{max_lo} is estimated as follows:
By inserting total weight of X-Pro into Equation [ABG+06, Eq. D.2] the entire Lift force needed to hover can be estimated. The Lift force is then applied to Equation [ABG+06, Eq. C.5c] along with the total mass value. The equation is then solved for, Ω (angular velocity) where the X-Pro maintains hover condition.
Changes in the R/C apply changes in input voltage to the motor model which returns angular velocity to the rotor, this is shown in Figure [ABG+06, Fig. A.6].

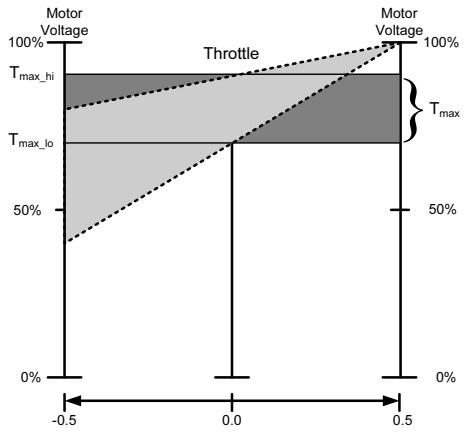

FIGURE 1.6: *The working area of the Throttle in flight.*

T_{max_hi} is scaled lower than the absolute maximum motor voltage, to approximately 90% of its capability. The reason for this scaling is to ensure manoeuvrability in the neighborhood of T_{max_hi}.

The working range of $T_{max} \epsilon$ [T_{max_lo},T_{max_hi}], is the entire working area of the R/C from hover and up. Consequently it can be concluded that greater range gives greater controllability from the viewpoint of the R/C. The working range of T_{max} is shown in Figure 1.6. In case of X-Pro preparing for landing or falling to the ground, it may be assumed that the working area of such a Throttle level is $T_{min} \epsilon$ [0,T_{max_lo}].

1.4 BLOCK DIAGRAM OF SYSTEM

The controlling signals of the rotors are illustrated with the block diagram in Figure 1.7.

The block diagram illustrates the manual R/C control system, where the R/C provides the input to the system and the angular rate of the X-Pro is fed back into the system for controlling correction. The role of the blocks in the diagram is explained as follows:

Controller: Proportional feedback control (P) is made to be linearly proportional to the system error. That kind of control is simple to construct, but has the drawback that it may lead to instability in higher order systems with great error. It is essential to respond swiftly to feedback error in the system and ensure minimal response delay in the system due to controlling. A derivative control (D) has sharp effect on sudden changes (errors) and reacts faster to them than proportional control, whereas it depends on the rate of change of the error. The negative side about the deriva-

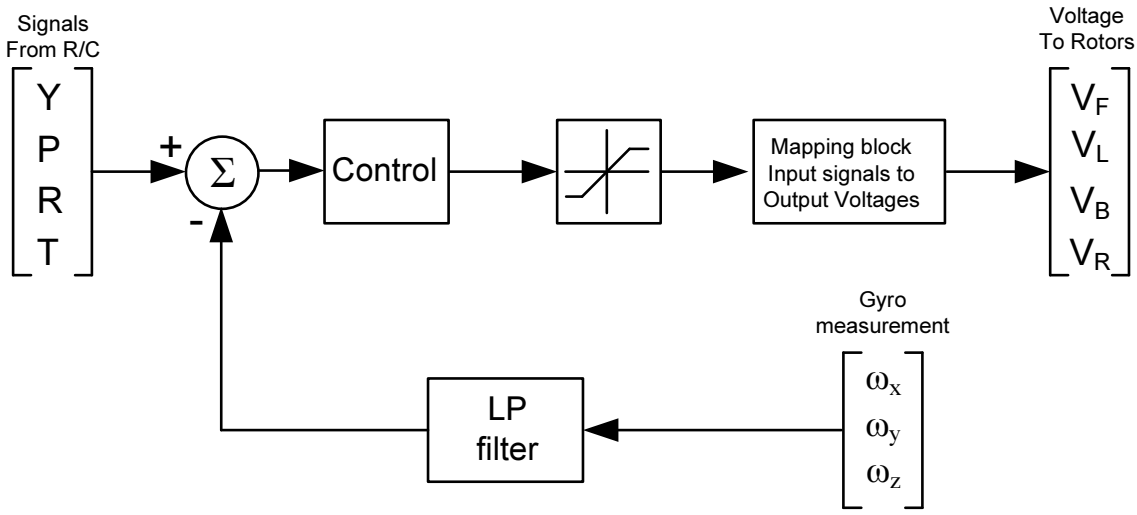

FIGURE 1.7: *The control signals corrected and converted to motor voltages.*

tive control is that high frequency noise from the system error would generate a big and incorrect control signal. By adding a pole to the controller it would become a lead compensator, which minimizes the effect from the high frequency noise from the sensors. The placement of the pole is a compromise between the effect of noise suppression which is achieved with the pole placed close to the imaginary axis and speed that is achieved for a pole placed into the negative half plane.

Mapping block: The mapping block includes a functionality which prevents saturation of the input signal to take place. The functionality of the mapping block has been explained in the previous section

LP-filter: In the previous project [ABG+06], it became apparent that the gyro measurements were distorted by signals with higher frequency compared to the measured angular rate of the X-Pro. An implementation of a low-pass filter on the gyro measurements would be appropriate, to prevent signal fluctuation to the motors as much as possible. It is previously known that the gyro measurements are sampled with $100\,\mathrm{Hz}$ [ABG+06], which gives a higher limit of the measured signal of interest. The LP-filter is therefore constructed with the corner frequency $f_c = 100\,\mathrm{Hz}$ and with the steady-state gain $A_0 = 1$. To keep the structure simple and as well to make the filter sharper (better damped at higher frequencies), it is constructed as 2^{nd} order low-pass filter.

The values between angular velocity from the gyros has to be scaled to the same level as the signals from the R/C.

WORKSHEET XVII: * CLOSURE OF THE 9'TH SEMESTER

3/1-07

CONTENTS OF WORKSHEET XVII

1 Current project status **163**

1.1 Preanalyses . . . 163

1.1.1 Dimensioning of IMU . . . 163

1.1.2 GPS investigation . . . 163

1.1.3 Analysis of magnetic disturbances . . . 163

1.1.4 Budgets . . . 163

1.2 Rebuilding of X-Pro hardware . . . 164

1.2.1 Change of R/C transmission frequency . . . 164

1.2.2 Dampening of vibrations . . . 164

1.2.3 New hardware . . . 164

1.2.4 New top-mount PCB, mountings, wiring . . . 165

1.3 robostix^{TM} SW . . . 165

1.3.1 Initial configuration . . . 165

1.3.2 Sensors . . . 165

1.3.3 Output . . . 166

1.3.4 Communication . . . 166

1.3.5 Safety feature . . . 166

1.3.6 Main . . . 166

1.4 gumstix^{TM} SW . . . 166

1.4.1 Initial configuration . . . 166

1.4.2 Sensors . . . 167

1.4.3 Communication . . . 167

1.4.4 Main . . . 168

1.5 Ground station . . . 168

1.5.1 Initial configuration . . . 168

1.5.2 Communication . . . 168

1.5.3 Data storage . . . 168

1.5.4 Main . . . 168

2 Incomplete tasks **169**

3 Plan for the 10'th semester **170**

3.1 Estimation group . . . 170

3.2 Control group . . . 170

3.3 Discussion of areas relevant for both groups . . . 170

CURRENT PROJECT STATUS 1

This chapter lists the work which has been conducted on the Draganflyer X-Pro (X-Pro), during the first semester. The listing is divided into three topics. The first is on the work done independ of the gumstix^{TM} and servers as a empirical analysis for some of the next work tasks. The next topic consers the changes made on the X-Pro. The last topic is on the status of the developed software.

1.1 PREANALYSES

This section lists the empirical analyses that was conducted mostly in the first part of the semester.

1.1.1 Dimensioning of IMU

From a previous project we knew that vibrations caused the accelerometers to saturate, thus the analysis was to find out what specifications was needed for a new Inertial Measurement Unit (IMU).

1.1.2 GPS investigation

The newly bought Global Positioning System (GPS) module has a high level protocol. Work has been put into getting the understanding of this protocol and the GPS module in general. This has been done mainly from a windows machine.

1.1.3 Analysis of magnetic disturbances

The analysis was conducted to estimate if it was possible to measure the magnetic field vector, or the magnetic disturbances from motors etc. were to great.

1.1.4 Budgets

Several budgets was made concerning economy, power consumption, space consumption by sensors and flight time vs. weight.

Economy Many of the sensors was relative expensive thus a budget was needed to verify if there was any sensors that we could not afford.

Flight vs. weight All the new hardware changes the weight of the X-Pro and the purchase of a new battery, made it necessary to estimate the weight at a early time in

the project period. Having this weight it was possible to estimate if the X-Pro will be able to fly and for how time on one battery charging. By this analysis the weight budget for the sensors was made tight, to minimise the extra weight added to the X-Pro.

Power All the electrical devices have relatively small current consumptions compared to the DC motors making the X-Pro hover. Even though this consumption was taken into consideration.

Space The space on the X-Pro main body are limited. The size of the hardware components though follows the weight, thus it has not been necessary to do a written budget concerning space, but it has been a part of the judging criteria for the choosing of components.

1.2 REBUILDING OF X-PRO HARDWARE

In order to prepare the X-Pro for autonomous flight, certain modifications were made:

1.2.1 Change of R/C transmission frequency

The Remote Control (R/C) module did not operate on a license-free frequency permissible in Denmark for remotely controlled airborne toys. The frequency was changed to one within the law in force.

1.2.2 Dampening of vibrations

Efforts have been made to reduce vibrations in the X-Pro, which in the previous project caused the linear accelerometers to saturate and thereby gave incorrect measurements.

1.2.3 New hardware

Battery: The battery was worn out, so it has been replaced. The new battery is lighter but has a lower energy content.

Motor drives: As the old Printed Circuit Board (PCB) mounted on the X-Pro was to be replaced, new motor drives have been designed, implemented and tested.

Onboard computers: A solution of two processors, gumstix^{TM} and robostix^{TM}, has been chosen as the platform for the project. Both processors have been mounted and

tested.

Communication: The communication possibility between a pilot / mission controller and the X-Pro has been expanded from the R/C module to also include a WiFi connection.

Sensors: The following sensor blocks have been obtained for the project: IMU, GPS sensor, SOund Detection And Ranging (SODAR) sensor and a magnetometer. Four tachometers were developed in the previous project.

Miscellaneous: The X-Pro has also been equipped with two status indicators; a buzzer and a Light Emitting Diode (LED).

1.2.4 New top-mount PCB, mountings, wiring

To be able to mount and connect the new hardware, a new PCB has been developed and tested. Sensor mountings have also been developed in co-operation with the metal workshop, and all hardware devices have been mounted on the X-Pro. A range of measures has been made in the process in order to accomodate electromagnetic disturbances, which could disturb especially the magnetometer and other hardware devices.

1.3 ROBOSTIX™ SW

The robostix™ handles communication and monitoring of most external hardware. It also has the safety feature of letting a human pilot take over the control of the X-Pro.

1.3.1 Initial configuration

The robostix™ has a Atmega128 processor as its CPU, the programing of this processor is done via the gumstix™, using a serial port. With the help of the gumstix™ communitys wiki page: `http://docwiki.gumstix.org/Main_Page` and the use of Atmels own C compiler it is possible to write programs in Standard C on any machine and transfer the hex file to the robostix™ via the gumstix™.

1.3.2 Sensors

Three sensors are connected to the robostix™ board for data collection:

Tachometer: Driver has been designed, implemented and tested.

IMU: Driver has been designed, implemented and tested.
Magnetometer: Driver has been designed, implemented and tested.

1.3.3 Output

The robostix^{TM} is supposed to set the duty cycles for the motor drives, and can also indicate certain parts of its status through status indicators.

Motor SW drivers: Driver has been designed, implemented and tested.
Status indicators: Integrated in system and tested.

1.3.4 Communication

gumstix^{TM} and robostix^{TM}: Driver has been designed, implemented and tested.

1.3.5 Safety feature

R/C: Driver has been designed, implemented and tested.

Human control: Human control is divided into two parts:

- *Mapping from R/C controls to motor duty cycles:* Concept has been designed, implemented and tested.
- *Simple dampening of angular rates:* Control has been designed and simulated.

1.3.6 Main

- Above-mentioned modules have been integrated, and the integration has been tested.
- A concept of switching between human, gumstix^{TM} and robostix^{TM} as pilot has been designed, implemented and tested.

1.4 GUMSTIX^{TM} SW

The gumstix^{TM} is intended to perform heavier tasks suchs as control, estimation, and communication through WiFi.

1.4.1 Initial configuration

The gumstix^{TM} configurations runs a small stripped linux OS, via a cross-compilation tool chain (overall denoted Buildroot) it has been made possible to add i2c kernel drivers

and otherwise recompile the linux image, that is booted on the gumstix^{TM} on startup. A full development environment has been set up on a 3GHz linux workstation. Connection to the gumstix^{TM} from the Development Host Machine (DHM) can be done by either main console, serial port or via WiFi. the software made on this platform are done in threads and to a wide extend made event driven in order not to consume all the CPU time in one thread.

1.4.2 Sensors

GPS The driver has been designed, implemented and tested to the extend that it works in the current software, running on the gumstix^{TM}. although since this is not the final software piece more work needs to be done on the interface towards the rest of the gumstix^{TM} software.

SODAR Driver has been designed, implemented, adjusted with respect to the desired needs and it has been tested on 400 kHz Inter-Integrated Circuit (I^2C) bus.

1.4.3 Communication

gumstix^{TM} and robostix^{TM} using the I^2C bus with the gumstix^{TM} as the master, sensor data is sent from the robostix^{TM}. The receiving of data packets is designed and implemented using calls to an I^2C kernel driver, but not fully tested. The speed of this connection has been tuned to 400 kHz on SCL from the original 100 kHz.

gumstix^{TM} and groundstation via WiFi the gumstix^{TM} and the groundstation are in contact. this connection is made by POSIX sockets and further more FIFO buffers are made on the gumstix^{TM} to ensure that every packets arrive to the groundstation even if a the connection is sometimes weak. The connection is ready in version 0.9, small corrections are needed to be able to loose the connection completely and then gain it again. The current implementation is tested, and fulfills the current needed functionality.

Differential Global Positioning System (DGPS) Can be received on the gumstix^{TM} via a socket, from here is it relay further to the GPS module via a serial port. These features has been implemented and tested.

1.4.4 Main

The main on the gumstix™ has the assignment to start up all threads that can handle the above mentioned features. The current main is small and simple and is to be replaced in the next semester.

1.5 GROUND STATION

The purpose of the ground station is firstly to receive sensor data chunks from the gumstix™ module and relay Radio Technical Commission for Maritime services (RTCM) packages to the gumstix™. An augmentation of the ground station software is intended at a later stage.

1.5.1 Initial configuration

The ground station software can be used with any newer laptop or PC with a standard linux distribution and a wifi connection.

1.5.2 Communication

gumstix™ and ground station The data chunks are relayed from the gumstix™ over the wifi connection to the ground station.

DGPS RTCM packages are received from the correction antenna through http to the ground station. These are relayed over the wifi connection to the gumstix™.

1.5.3 Data storage

The data received is conveniently stored in csv files for easy access from a matlab environment.

1.5.4 Main

At present time the RTCM relaying and the receiving of data chunks are two separate programs. This means that the NtripLinuxClient which handles the connection to the Hals antenna is started separately from the data receiving program. This style of implementation has also been carried out on the gumstix™, except that the RTCM handling is started as a thread.

INCOMPLETE TASKS 2

The following tasks are yet to be completed before the start of next semester:

Magnetometer driver: Sometimes the magnetometer driver stops receiving new samples, but does not detect it as a communication error. It has been tested, that the magnetometer driver does detect when the cable to the sensor is removed, and considers this a communication error. The fault has not yet been isolated, but appears to be within the driver, or is a result of the fault in the tachometer three-buffer system.

I^{2}C connection: The gumstix^{TM} sometimes receives packages from the robostix^{TM}, which only consist of bytes with the value 256. The fault has not yet been isolated.

Ground station data reception: Some of the ground station software has not yet been finished due to gumstix^{TM} not receiving the packages from the robostix^{TM} correctly. This mainly concerns directing specific data into specific csv files etcetra.

R/C Control: The dampening control block as illustrated in Figure 1.7 on Page 161 has not been implemented.

Tachometer driver: The three-buffer system implemented in the tachometer driver does not function correctly. The origin of the fault has been identified and isolated, and is considered a minor fault.

robostix^{TM} acceptance test: The robostix^{TM} has not been tested with all elements fully implemented, so it has not been verified, that it is able to conduct all it's tasks within the required $1/100$ s at maximum load.

PLAN FOR THE 10'TH SEMESTER

3

This chapter gives an overview of ideas concerning what the two groups can work with on the 10'th semester. The ideas should be used in a discussion about what elements the master thesis of the two groups can consist of. The issue has been discussed at a supervisor meeting where two main areas of interest were discussed, i.e. estimation and control.

3.1 ESTIMATION GROUP

The estimation project could include some of the following areas:

- Sensor modeling
- Sensor Fusion
- Hybrid Estimator
- other areas?

3.2 CONTROL GROUP

The control project could include some of the following areas:

- Hybrid Control
- Robust Control
- Optimal Control
- Non-Linear
- other areas?

3.3 DISCUSSION OF AREAS RELEVANT FOR BOTH GROUPS

The following list includes areas of relevance for both groups:

Model revision/verification To what extent is revision and verification of the 8'th semester model necessary?

Adaptive control / Online par. estim. An area of interest?

Test pilot Should the test pilot be one or two from the groups or a certified pilot?.

Integration of control & estimator Should there be cooperation between the two groups at some level?.

ABBREVIATIONS

ADC Analog/Digital Converter

API Application Program Interface

CCD Conditioned Compass Data

CCW Counter-Clock-Wise

CEP Circular Error Probability

CM Center of Mass

CW Clock-Wise

DES Discrete Event System

DGPS Differential Global Positioning System

DHCP Dynamic Host Configuration Protocol

DHM Development Host Machine

ECEF Earth Centered Earth Frame

EEPROM Electrically Erasable Programmable Read Only Memory

EGNOS the European Geostationary Navigation Overlay System

EMC Electro Magnetic Compatibility

EMI Electro Magnetic Interference

FET Field Effect Transistor

FFC Flat Flex Cable

FFUART Full Function Universal Synchronous-Asynchronous Receiver/Transmitter

FSA Finite State Automata

G2R Gumstix-to-Robostix connection

GMT Greenwich Mean Time

GPIO General Purpose Input/Output

GPS Global Positioning System

HPR Heading, Pitch and Roll

I/O Input/Output

I^2C Inter-Integrated Circuit

ILED Infrared Light-Emitting Diode

IMU Inertial Measurement Unit

ISR Interrupt Service Routine

JTAG Joint Test Action Group

LAN Local Area Network

LED Light Emitting Diode

Li-Po Lithium-Polymer

MIT Massachusetts Institute of Technology

NMEA National Marine Electronics Association

Ntrip Networked Transport of RTCM via Internet Protocol

OBC On Board Computer

OS Operation System

PCB Printed Circuit Board

PDOP Position Dilution Of Precision

PWM Pulse Width Modulation

R/C Remote Control

RCD Raw Compass Data

RTCM Radio Technical Commission for Maritime services

SAB Smart Adaptor Board

SBAS Satellite Based Augmentation System

SEP Spherical Error Probability

SMD Surface Mounted Devices

SOA Safe Operating Area

SODAR SOund Detection And Ranging

SPI Serial Peripheral Interface

SRAM Static Random Access Memory

STUART Standard Universal Synchronous-Asynchronous Receiver/Transmitter

UART Universal Asynchronous Receiver Transmitter

UBX Ublox Binary Protocol

WAAS Wide Area Augmentation System

WAN Wide Area Network

X-Pro Draganflyer X-Pro

BIBLIOGRAPHY

[ABG+06] Mikael Berg Andersen, Ole Binderup, Sigurgeir Gislason, Jesper Haukrogh, Morten Kjærgaard, and Martin Sørensen. Draganflyer X-Pro Modelling and Control. Technical report, Aalborg University, 2006.

[Atm] Atmel. 8-bit AVR Microcontroller with 128K Bytes In-System Programmable Flash. URL http://www.atmel.com/atmel/acrobat/doc2467.pdf. Viewed 01-11-2006.

[AVR] AVR Freaks. Design Note #007, Little and Big Endian. URL http://www.avrfreaks.net/Tools/ToolFiles/239/DN_007.pdf. Viewed 01-11-2006.

[BVs04] Morten Bisgaard, Dennis Vinther, and Kasper Zinck Østergaard. Modelling and Fault-Tolerant Control of an Autonomous Wheeled Robot. Technical report, Aalborg University, Department of Control Engineering, Intelligent Autonomous Systems, 2004. URL http://www.control.aau.dk/farming/04gr1030a/report/report.pdf.

[DB] Columbia Md. David Bourner, National Semiconductor. Capacitor Characteristics Impact Power Supply Decoupling. Power Electronics Technology magazine. URL http://powerelectronics.com/mag/power_capacitor_characteristics_impact/. Viewed 25-10-2006.

[fGR98] Center for Global and Regional Environmental Research. GPS Glossary. Technical report, University of Iowa, Center for Global and Regional Environmental Research, 1998. URL http://www.isprs.org/istanbul2004/comm2/papers/97.pdf. Viewed the 1/12-06.

[Gum06a] GumstixDocsWiki. Frequently asked questions/IO. Technical report, Gumstix, 2006. URL http://docwiki.gumstix.org/Frequently_asked_questions/IO. Viewed 20-11-2006.

[Gum06b] GumstixDocsWiki. Robostix gumstix ISP. Technical report, 2006. URL http://docwiki.gumstix.org/Robostix_gumstix_ISP. Viewed 18-10-2006.

[Gum06c] GumstixDocsWiki. Robostix I/O pins. Technical report, 2006. URL http://docwiki.gumstix.org/Robostix_I/O_pins. Viewed 16-10-2006.

[Han06] Hans. Tips og oplysninger om Li-Po og Li-Ion celler. Technical report, www.el-fly.dk, 2006. URL http://www.el-fly.dk/cell-tek.htm. http://www.control.auc.dk/ 06gr937a/CDROM/datasheet/Electric%20Flight%20Equipme

[Hon] Honeywell. HMR3000 Digital Compass Solution User's Guide.

[i2c00] The I^2C-Bus Specification, January 2000. URL http://www.semiconductors.philips.com/acrobat_download/literature/9398/39340011.pdf. Viewed 10-11-2006.

[inc] Draganfly Innovations inc. Draganflyer Flight Control Diagram. Technical report. URL http://www.rctoys.com/pdf/df5-flightcontrol.pdf. Viewed 06-12-2006.

[Mic] Microstrain, Inc. 3DM-GX1 Hard Iron Calibration. URL http://www.microstrain.com/pdf/3DM-GX1%20Hard%20Iron%20Calibration.pdf.

[mot] Mabuchi motor. RS-545SH. Technical report, Mabuchi motor co. ltd. URL http://www.control.auc.dk/~06gr937a/CDROM/datasheets/rs_545sh.pdf.

[O-N] O-Navi. Falcon/GX - Digital Tri-Axial 6 Degree-of-Freedom MEMS IMU Module. URL http://www.o-navi.com/falcongx.htm. Viewed 10-10-2006.

[Pet04] Martin Peterzon. Distribution of GPS-data via Internet. Technical report, LANTMÄTERIET, 2004. URL http://www.lantmateriet.se/upload/filer/kartor/geodesi_gps_och_detaljmatning/Rapporter-Publikationer/LMV-rapporter/Lmv_Rapport_2004-01_exjobb_Peterzon.pdf. Downloaded to CDROM/gps/DGPS_litterature.

[PNI06] PNI Corp. PNI MicroMag 3 - 3-Axis Magnetic Sensor Module, June 2006. URL https://www.pnicorp.com/downloadResource/c40c/manuals/110/MicroMag3+3-Axis+Sensor+Module_June+2006.pdf. Viewed 25-10-2006.

[PSV04] A. Pala, G. Sanna, and G. Vacca. Real Time Mapping With DGPS-Enabled Navigation Equipment. Technical report, University of Cagliari Department of Structural Engineering, 2004. URL http://www.isprs.org/istanbul2004/comm2/papers/97.pdf. Downloaded to CDROM/gps/DGPS_litterature.

[Rao04] Singiresu S. Rao. Mechanical Vibrations. Prentice Hall, 2004. ISBN 0-13-120768-7.

[Ras04] Muhammed H. Rashid. Power Electronics, Circuits, Devices and Applications. Prentice Hall, 2004. ISBN 0-13-122815-3.

[Roba] Robot Electronics. SRF08 Ultra sonic range finder Technical Specification. URL http://www.robot-electronics.co.uk/htm/srf08tech.shtml. Viewed 16-10-2006.

[Robb] Robot Electronics. Using the I2C Bus. URL http://www.robot-electronics.co.uk/htm/using_the_i2c_bus.htm. Viewed 16-10-2006.

[SB00] Raymond A. Serway and Robert J. Beichner. Physics - For Scientists and Engineers with modern physics. Harcourt College Publishers, 2000. ISBN 0-03-022657-0.

[Sur05] U. S. Geological Survey. Maps for the Earth's Surface. Technical report, USGS National Geomagnetism Program, 2005. URL http://geomag.usgs.gov/movies/. Viewed 25-10-2006.

[The03] The Phaeton Project, Hardware Purchase Document. Technical report, Massachusetts Institute of Technology, 2003. URL http://www.mit.edu/~16.82/html/16.82_fall03/old%20html/Purchase%20Document.pdf.

[UB90] J. Zweig (UIUC) and C. Partridge (BBN). TCP Alternate Checksum Options. Technical report, Network Working Group, March 1990.

[Ubla] Ublox. SAM-LS, GPS Smart antenna adapter board data sheet. Technical report. URL http://www.u-blox.com/products/Product_Summaries/SAB_Prod_Summary(GPS-X-04005).pdf. Viewed 12-11-2006.

[Ublb] Ublox. SAM-LS, GPS Smart antenna module, ANTARIS Positioning engine. Technical report. URL http://www.u-blox.com/products/sam_ls.html. Viewed 12-11-2006.

[Ublc] Ublox. SAM-LS, GPS Smart antenna module data sheet. Technical report. URL http://www.u-blox.com/products/Data_Sheets/SAM-LS_Data_Sheet(GPS.G3-SA-03002).pdf. Viewed 12-11-2006.

[Ubld] Ublox. SAM-LS, GPS Smart antenna module data sheet extended. Technical report. URL http://www.u-blox.com/customersupport/gps.g3/TIM-Lx_Sys_Int_Manual(GPS.G3-MS3-01001).pdf. Viewed 12-11-2006.

[Uble] Ublox. SAM-LS, GPS Smart antenna module data sheet extended. Technical report. URL http://www.u-blox.com/customersupport/gps.g3/ANTARIS_Protocol_Specification(GPS.G3-X-03002).chm. Viewed 12-11-2006.

[Vit95] Joseph Vithayathil. Power Electronics, Principles and Applications. McGraw-Hill Inc., 1995. ISBN 0-07-067555-4.

[Wol] Axel Wolf. I2C (Inter-Integrated Cirtuit) Bus Technical Overview and Frequently Asked Questions (FAQ). ESAcademy. URL http://www.esacademy.com/faq/i2c/. Viewed 16-10-2006.