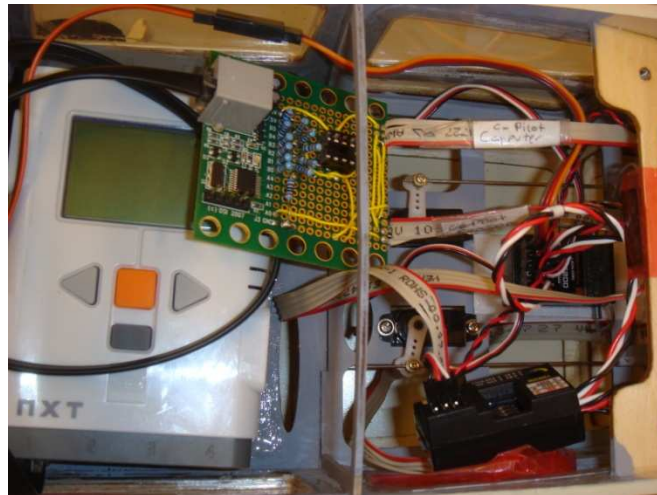

November 2009

Assembly of a Unmanned Aerial Vehicle (UAV)

Bachelor Thesis in
Electrical Engineering

Ugur Bozkurt & Mustafa Aslan



School of Information Science, Computer and Electrical Engineering

Halmstad University

Assembly of a Unmanned Aerial Vehicle (UAV)

Bachelor Thesis in
Electrical Engineering

School of Information Science, Electrical Engineering
Halmstad University
Box 823, S-301 18 Halmstad, Sweden

November 2009

Acknowledgement

*“Only those who will risk going too far can possibly
find out how far one can go.”
T. S. Eliot*

This project was performed in CERES laboratory in Halmstad University with two groups; one was responsible for the interface with a GPS and for the development of software to navigate the UAV in the air, and our group was responsible for assembling the hardware system, providing interfaces to the system actuators and the autopilot in order to allow the autonomous flight.

We first would like to thank our supervisors at Halmstad University Edison Pignaton de Freitas and Kristoffer Lidström because of their supports and ideas helping us during this process. We also thank the other group assigned in the rest of the project for their labour and collaboration, Denny A Toazza and Tae Hyun Kim.

You always guided us with your e-mails and meetings from the beginning to the end, thank you Kenneth Nilsson.

Furthermore a special thanks to our families to believe and support us.

*Halmstad, November 2009
Ugur Bozkurt & Mustafa Aslan*

Preface

Title: Assembly of an Unmanned Aerial Vehicle (UAV)
Authors: Ugur Bozkurt & Mustafa Aslan
Degree program: Electrical Engineering
University: Halmstad University
Year: 2009

Supervisors: MSc Edison Pignaton de Freitas,
Lic D Kristoffer Lidström

Abstract

This bachelor thesis is dedicated to assemble the hardware system of a UAV (Unmanned Aerial Vehicle) in order to prepare the platform for an autonomous flight in the air for a given path through the pre-programmed check points. A UAV is an aircraft that contains sensors, GPS, radio system, servomechanisms and computers, which provide the capability of an autonomous flight without a human pilot in the cockpit. A stable flight requires sensing the roll, pitch, and yaw angles of aircraft. Roll and pitch angles were ensured by a sensor system of FMA Direct Company called co-pilot flight stabilization system (CPD4), which allows controlling ailerons and elevator manually.

An autopilot is required for steering the aircraft autonomously according the GPS data and the establish waypoints that the airplane have to pass by. The GPS gives heading information to the autopilot, and this uses the information of the next waypoint to decide which direction to go. Hereby an autonomous flight is provided. In this project a lego mindstorm NXT was used as an autopilot that is product of LEGO Company [1]. The output of the autopilot is used to control the airplane servos to fly in the desired direction. A software and hardware interface was designed to allow the autopilot to receive the data from the co-pilot sensor and to transmit data to the co-pilot processor, which will finally steer the actuator servos. Experiments were performed with different parts of the system and the results reported.

Contents

ACKNOWLEDGEMENT	I
PREFACE	II
ABSTRACT	II
CONTENTS	III
1 INTRODUCTION	1
1.1 BACKGROUND.....	1
1.2 THESIS OVERVIEW	4
1.3 LIMITATIONS.....	4
1.4 RELATED PROJECTS	5
2 HARDWARE COMPONENTS	7
2.1 LIST OF THE COMPONENTS USED IN THE PROJECT	7
2.2 AIRCRAFT	8
2.2.1 STEPS FOR ASSEMBLY	9
2.2.2 CONTROL THROWS	9
2.2.3 CENTRE OF GRAVITY	9
2.3 RADIO SYSTEM	10
2.3.1 TRANSMITTER AND RECEIVER.....	10
2.3.2 BIND PLUG	11
2.4 BRUSHLESS DC MOTOR.....	12
2.5 BRUSHLESS ESC.....	13
2.6 RC SERVO.....	13
2.7 FLIGHT STABILIZATION SYSTEM	14
2.8 AUTOPILOT	16
2.8.1 NXT-G SOFTWARE	18
2.8.2 LEJOS NXJ.....	19
2.9 NXT PROTOTYPE BOARD	20
3 METHODOLOGY	21
3.1. PROJECT STEPS.....	21
3.2. CO-PILOT SENSOR CHARACTERISTICS AND USAGE	21
3.3. MINDSTORM NXT AS AN AUTOPILOT.....	22
3.4. UAV CONTROL.....	23
4 EXPERIMENTS AND RESULTS	24
4.1. CO-PILOT MEASUREMENTS	24
4.2. UNDERSTANDING THE MINDSTORM NXT HARDWARE	27
4.3. PROVIDING AN INTERFACE FROM NXT TO CO-PILOT.....	29

4.4. TROUBLESHOOTING	33
4.5. HARDWARE INSTALLATION.....	34
4.6. PROJECT INTEGRATION	35
4.7. FLIGHT TEST	36
5 CONCLUSIONS	38
REFERENCES.....	39
APPENDIX A	41
APPENDIX B	43

1. Introduction

1.1. Background

UAVs are aircrafts do not need a pilot to be navigated in the air. Remote controlled systems or autopilots are used instead of a pilot on board. The merit of autonomous flight makes UAVs preferable beside remote controlled ones. UAVs have a great potential of using in real life applications such as reconnaissance, fire-fighting, surveillance, and many other civilian and military applications.

For purposes of basic understanding, flight dynamics principals are described in 3-dimensional coordinate system (see Figure 1).

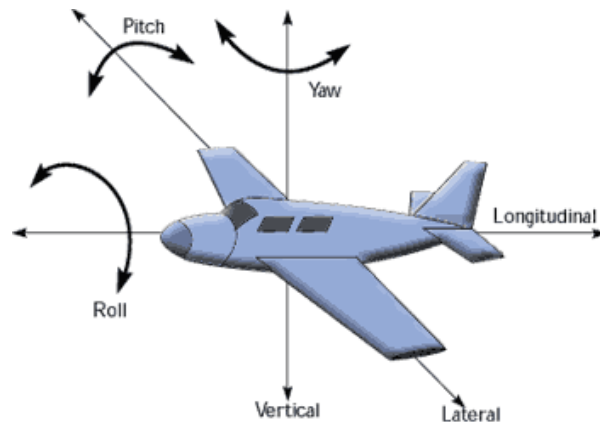


Figure 1 Flight dynamics [2]

The centre of gravity of airplane represents the origin of this coordinate system. The axes are called lateral, longitudinal, and vertical which correspond to pitch, roll, and yaw respectively. Following table explains how to control airplane.

Control in Transmitter	Control in Autopilot	Control panel	Motion
Aileron stick (Left &Right)	Left &Right sensors	Ailerons	Roll
Elevator stick (Up &Down)	Front &Back sensors	Elevator	Pitch
Rudder stick (Left &Right)	Autopilot	rudder	Yaw

Table 1 Controlling the airplane.

According to the data presented in Table 1, an airplane can make 6 motions in the air by both manually or autonomously, and these motions can be described with three dimensional coordinate system.

As ailerons were used instead of rudder in the project, it is important to understand how they work to produce rolling and yawing motions.

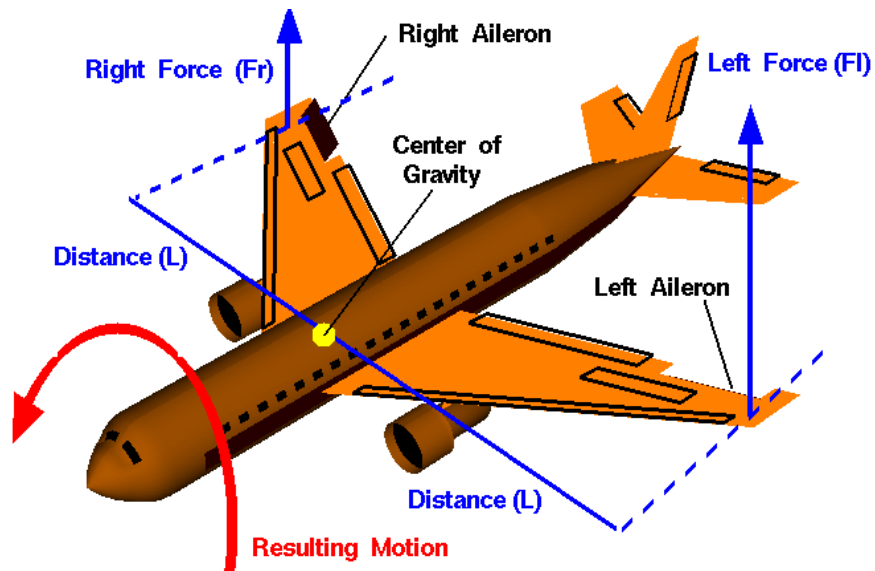


Figure 2 An illustration for the ailerons about how they work [24].

The use of ailerons is needed to be able to bank an aircraft to produce a rolling motion (or yawing motion). If the ailerons are moved as in the Figure 2 (right panel up and left panel down), the resulting force causes the right wing tip to move down and the left wing to move up. It causes the aircraft to turn in a direction which is opposite to the turn generated by the ailerons. In other words in the case of using ailerons instead of rudder to turn the aircraft the yawing motion is called “adverse yaw”.

The hardware development for UAVs is mostly based on choosing of the autopilot and the flight stabilization systems that make it possible to control the possible motions described above.

An autopilot is an electronic platform that controls the aircraft to operate autonomously by the outputs of the software that runs on it. In the project the autopilot was assigned to manipulate the data coming from GPS and actuate sub micro servos to be able to navigate the aircraft in the air via pre-programmed points (waypoints). This can be achieved by controlling either aileron or rudder panels. However aircraft is levelled by flight stabilization systems sensing the temperature difference between ground and sky. According the sensor data ailerons and elevator servos are actuated. A fail safe system is required to handle the control when autopilot is failed. So hardware system designed with remote control system consists of receiver and transmitter parts. The following figure illustrates commonly used UAV hardware architecture, which can be used as an overview of the hardware system developed in this work.

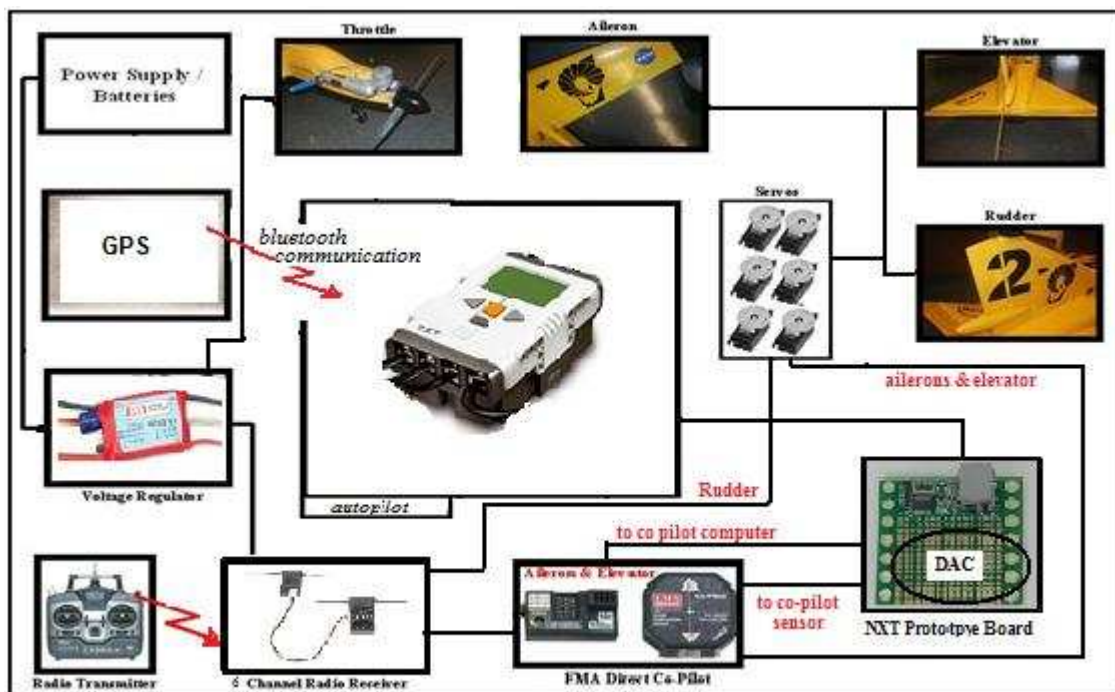


Figure 3 An Overview to UAV System Hardware Architecture [3].

Figure 3 depicts the hardware system installed in the UAV that provides autonomous flight through the actuators which reacts to infrared sensors and GPS fairly placed in the UAV.

The following thesis was intended to design a hardware system for UAVs, which can guide itself by raw GPS data processed in the autopilot. Problems were inevitable while hooking up the hardware parts to each other. For instance NXT outputs do not support a direct connection

with the components such as RC servos and co pilot sensor due to incompatibilities of the signals. The solutions to such problems are presented in this report.

1.2. Thesis Overview

The report has been written based on the following titles.

- Assembling an aircraft model.
In this project a Taylorcraft450 model has been used.
- Measurements on hardware platform.
Co-pilot sensor outputs for sub micro servos were measured.
- Experiments on autopilot.
Autopilot (mindstorm NXT as an autopilot was chosen in this project). It does not support digital (PWM) or analog outputs. It only allows I²C (serial data bus) communication. So experiments were performed and researched on mindstorm NXT output ports.
- Making an interface between autopilot and co pilot.
To actuate the sub micro servos with manipulated GPS data, an autopilot is required to interact with the co-pilot.
- Troubleshooting.
Problems encountered during the previous steps were clarified.
- Hardware installation.
After the problems were resolved, the hardware was installed in the aircraft model.
- Combining the projects.
The autopilot was programmed by the other group responsible for the navigation control. The two projects were combined to be able to deliver the complete UAV system.
- Flight test.

1.3. Limitations

In this project the UAV was allowed moving only in 2 dimensions during autonomous flight. Takeoff and landing are controlled by remote control. Otherwise the autopilot is required to control throttle channel according the data, which can be provided by an altitude sensor. Namely hardware system gets more complex since NXT ports cannot support PWM signals for

throttle. I²C to PWM converter can be used but it increases the cost and complexity of the UAV system.

1.4. Related projects

UAVs were used firstly in military applications. But they can also be used for many different purposes. These promising aspects make them very desirable. Many projects are available related to unmanned aircrafts. Some of them are presented as follows:

- GeoCrawler1, 2, 3, and 4 [\[4\]](#).

Created by Chris Anderson. In GeoCrawler2 lego mindstorm NXT was used as an autopilot which controls only rudder servo for navigation as we inspired in this project, and IMU based flight stabilization system was used. In GeoCrawler2 touch sensor was used for switching between autopilot and remote control and a mechanical connection between autopilot and the rudder servo was provided to control yawing. The disadvantages of GeoCrawler2 are that it requires more space in the airplane, the turn value is not very sensitive, and an extra component was used for switching. In this project the NXT Prototype Board was used to control yawing that does not require additional space in the UAV, has the sensitivity of 64 values for yawing, and the ailerons were used for yawing through the co-pilot computer, which provides a channel for switching the autopilot on. The UAV presented in this project is superior to GeoCrawler2 in these aspects. The only feature that represents a plus in the GeoCrawler2 project is that it has a gimbal camera to observe the environment, which was not provided in this project. However, a camera can be added to the aircraft platform in a future work.

- Güventürk [\[5\]](#).

Product of METU (Middle East Technical University). FCS (Flight Control System) platform was uniquely designed. Güventürk provides fully autonomous flight that means takeoffs and landings can be performed autonomously. The hardware system does not occupy a big space on board because of that most of the components were designed originally and placed on a single board such as GPS module, autopilot, flight stabilization system, etc. Therefore Güventürk is very advantageous in all mentioned aspects in relation to

the UAV developed in this project. However, it is important to highlight that in our project only components of the self were used. It is an advantage if compared to system developed from the scratch in Güventürk project.

- Dragon Eye [\[6\]](#).

Developed by Naval Research Laboratory and Marine Corps Warfighting Laboratory, it has a GPS-IMU based navigation system. It is used for reconnaissance and surveillance. The features of Dragon Eye are that fully autonomous operation, ability of updating the waypoints during the flight, image capturing, laptop mapping, and long endurance (45-60 minutes). In this project the developed UAV is much simpler than the Dragon Eye.

- Bayraktar [\[7\]](#).

Product of Turkey's Baykar Machine Inc. It is designed for short range reconnaissance and surveillance applications. It has a day/night camera and ground based monitoring system. Takeoffs and landings are autonomous as well. In case of lost communication it returns home. With these features it is superior to the one presented in this project.

- EMT Aladin (**Abbildende Luftgestützte Aufklärungsdrohne im Nächsbereich**) [\[8\]](#)

Product of Bundeswehr (German Army). It is used for reconnaissance by German Army. It has a day and night camera for observing the environment. It can fly fully autonomously, and has daylight and IR video sensors and ground control station for mapping, mission planning, and flight controlling. These features were not included in the UAV presented in this project yet.

2. Hardware Components

2.1. List of the Components used in the Project

Adhesives

6 minute epoxy
30 minute epoxy
Canopy glue
Thin CA 28gr
Medium CA 28gr
Threadlock

Tools

Propeller & propeller adapter
Felt-tipped pen
Drill bit (1.5 and 2mm)
Medium grit sandpaper
Cardstock
Square
Brush set

Electronic components |

Lego Mindstorm NXT
Spektrum DX6i transmitter & AR6200 receiver
FMA Direct Co Pilot™ (PN CPD4)
Nippy Black 1210/103 BLDC motor
HiTechnic's NXT Solderless Prototype Board
NXT servomotor
GPS module
Sub micro servos
Brushless ESC (JETI Model ECO 18)
Li-Poly battery
Module cable RJ12 6/6
Servo cables
Panel potentiometer 1k Ω
Battery LR6/AA
LM741DIP-8 op-amp
0,25W 1% 1.0 K
0,25W 1% 2.0 K

The kit/parts included in airplane model (Taylorcraft 450 ARF)

Wing set w/Ailerons
Fuselage
Tail set
Cowling
Landing gear & covers
Wheel pants
Pushrod set
Fuselage hatch
Window set
Wing strut set

2.2. Aircraft

Taylorcraft450 ARF used in the project is a hobby aircraft model created by E-flite and very suitable for beginners (see Figure4).



Figure 4. Taylorcraft 450 ARF [9]

It is a small type of aircraft and contains internal servo mounts concealed from view.

Endurance of 12 minutes is provided by 11.1V, 3 cell Li-Poly battery in the air. However Taylorcraft can fly in the winds up to 10 mph.

In the manual specifications are introduced as in the following table [10].

Specifications	
Wingspan:	46 in (1170mm)
Length:	36 in (915mm)
Wing Area:	370 sq in (23.87 sq dm)
Weight w/o Battery:	29–31 oz (680–740 g) Weight w/
Battery:	24–26 oz (820–880 g)

Table2. Taylorcraft specifications

2.3. Steps for Assembly

Assembly consists of the following steps. Detailed information about most of the steps can be found in user's manual [10].

- Joining the Wing Panels
- Tail Installation
- Landing Gear Installation
- Aileron Servo Installation
- Motor and Cowling Installation
- Rudder and Elevator Servo Installation
- Radio system (receiver) and FCS (Flight Control System) Installation
- Window and Strut Installation

2.4. Control Throws

Using a ruler throws of ailerons, elevator and rudder are adjusted to obtain the desired rates by adjusting the positions of pushrods.

For example following measurements are given in user's manual [10].

<i>Ailerons</i>
Low Rate: 1/2-inch (13mm) with 15% Expo (Up/Down)
High Rate: 3/4-inch (19mm) with 25% Expo (Up/Down)
<i>Elevator</i>
Low Rate: 1/2-inch (13mm) with 15% Expo (Up/Down)
High Rate: 7/8-inch (22mm) with 25% Expo (Up/Down)
<i>Rudder</i>
1 1/2-inch (38mm) (Left/Right)

Table 3. Measured throws of ailerons, elevator and rudder

2.5. Centre of gravity

After hardware installation is completed, centre of gravity (CG) of aircraft must be balanced before the first flight. Taylorcraft user's manual recommends 60-73mm back from leading edge of the wing for CG location (see Figure 5).

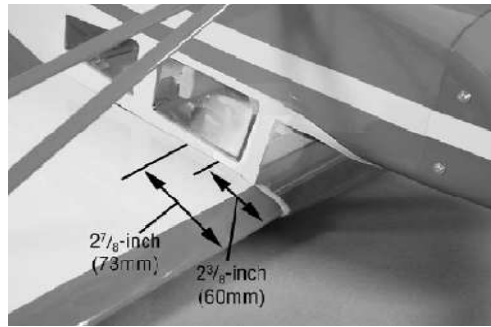


Figure 5. Centre of gravity of Taylorcraft [10]

CG can be changed after first flight according to the user's choice. It is important to achieve recommended rates not to lose control of aircraft in the air.

2.6. Radio System

2.7. Transmitter and Receiver

Spektrum's DX6i 6-channel radio system [11] (see Figure 6) used in this project includes 2.4 GHz AR6200 DSM2 6-channel receivers and supports full range for all types of model aircraft. Advantages of spread spectrum RC technology are fast communication between Rx and Tx that provides instantaneous servo response, two additional channels on receiver is useful for switching between autopilot and radio system, etc. However programming interface through the LCD display allows adjusting many features (see Table 4).



Figure 6. Spektrum's DX6i transmitter, AR6200 6-channel receiver and bind plug [12]

Features	
10-model memory	Graphic throttle curve
Dual and exponential rates	Graphic pitch curve
Dual aileron / flaps / V-tail / Delta/ P mixes	Rudder dual rate
Servo reverse	Revolution mix
Sub-trims	2 swash types 1 servo/3 servo CCPM
Travel adjust	Throttle cut
Timer / Alarm	Model copy
Trainer mode	Servo monitor
Gyro adjust	Modelmatch
Range checking	DSM2 DualLink@ Technology

Table 4. Features of Spectrum's DX6i transmitter [11]

Most beneficial aspect of the receiver, which consists of two parts (main receiver and remote receiver), is to provide more precision while getting the signal coming from the transmitter (i.e. even if the main receiver cannot catch the signal, the remote receiver can handle it or vice versa).

2.8. Bind Plug

This item (see Figure 7) is required for binding operation that ensures Tx to accept signals sent from only the bounded Rx. Tx can recognize different receivers up to 10 models. Thus Tx does not confuse with another signals in the same frequency range. This operation should be performed before each flight.

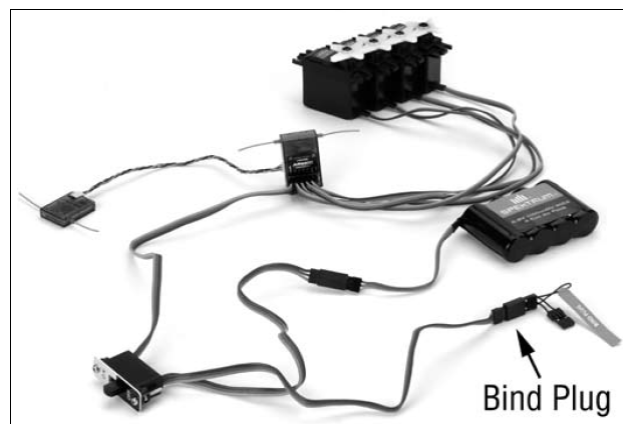


Figure 7. Binding operation [11]

Bind plug is inserted corresponding channel of receiver. Then receiver is powered by battery via ESC which provides 5V from the throttle channel. Leds on the receiver starts blinking that shows system is ready for binding operation, which is performed as follows: (1) Turn on the transmitter while pulling and holding trainer switch. LCD display indicates 'BINDING'. (2) After 3 seconds leds on receiver go solid, hereby, system is connected. (3) Removing bind plug the binding operation is completed.

2.9. Brushless DC Motor (Outrunner)

Brushless DC motor [13] (BLDC) is an electric motor which operates by DC electric. The difference from brushed motors is that commutation is controlled electrically instead of mechanical control as in the brushed motors. Nippy black 1210 /103 model used in the project is depicted in the Figure 8.



Dimensions	Ø 40,2 x 48 mm
Shaft diameter	4 mm
Weight	48 g
No. of cells	Li-Poly 2 ÷ 3 NiCd, NiMH 8 ÷ 10
Maximum sustained current	12 A
Maximum efficiency	79%
Current for efficiency > 72%	5 ÷ 9 A
Current for efficiency / 9,6V	1,2 A
Internal resistance	110 mΩ
No load rpm / V	1.030

Figure 8. Nippy Black 1210 /103 Brushless and Sensorless DC Motor [14]

BLDC motors are more advantageous than brushed motors with the aspects of high efficiency, reliability, reduced noise, longer lifetime, eliminated brush erosion, and more power. Applied power is only limited by heat that can cause harm on magnets.

Requirement of electronic speed controllers (ESCs) to drive BLDC motors makes their cost higher while brushed motors can be controlled by simple controllers such as potentiometer.

2.10. Brushless ESC (Electronic Speed Control)

Electronic speed control [15] (ESC) consists of an integrated circuit to drive motor via radio control. Latest version of ESC incorporates battery eliminator circuit (BEC) that removes more battery need. In many such ESCs support Li-poly, and NiMH batteries. It is important to consider the required ratings before choose a battery. At this point which type of battery should be used is another issue to think about. JETI model ECO 18 used in this project can be seen from the following figure.



Figure 9. Brushless ESC (JETI model ECO 18) [16]

ESC takes the signal from throttle channel of receiver, and supplies 3 phased AC voltages to motor. Polarization of three leads from ESC to motor is important to achieve desired rotation of motor.

An important benefit of ESC for model airplanes is to cut the power of propeller motor when the battery is low and keep the controls for ailerons, elevator and rudder on. Hereby, airplane can glide down in safe.

2.11. RC Servo

RC servos [17] are devices which can be controlled remotely. They are employed as actuators in many applications such as steering of a car, flaps on model airplane, active parts on a robot (i.e. moving the arms and legs), etc. In this project sub micro servos were used to control ailerons, elevator and rudder panels on the airplane (see Figure 10).



Figure 10. Sub-Micro RC Servo [18]

RC servos include a motor which cooperates with a potentiometer through the gear box. PWM signals are translated to achieve desired position from servo arm through the electronic circuit inside it. Motor is actuated up to potentiometer attains desired angle from the servo arm. RC servos consist of three wires which are for ground, supply voltage, and PWM signal, respectively.

Angle of servo arm is controlled by PWM signals. For example 1.5ms width of a pulse corresponds to 90° and commonly assumed as neutral position for a servo. Servo arm can be rotated from 0° to 180° .

Most servos are employed under the DC voltage of 4.8V-6V, and NiCd, NiMH, or LiPo batteries can be used after provide the correct ratings for servo through ESC, or any regulator.

2.12. Flight Stabilization System

Co-pilot [19] is a module composed by a sensor and a computer device. The sensor device provides flight stabilization with respect to difference in infrared signals between earth and CO_2 in the atmosphere. Double side sensors in two directions provide day or night stabilization for both ailerons and elevator (see Figure 11).

Co-pilot allows intervening flight system through receiver at any time and this makes co-pilot very advantageous if compared with other flight stabilization systems. This feature of this co-

pilot is achieved by the co-pilot computer, which takes signals from both receiver and co pilot sensor and translates them for aileron and elevator servos.



Figure 11. Co-pilot flight stabilization system (CPD4) of FMA Direct Company [19]

Co-pilot provides only one output for elevator and one output for aileron while two outputs for ailerons are needed. To achieve two outputs for ailerons from one output Y servo harness are used which consists of leads that are connected in parallel (see Figure 12).

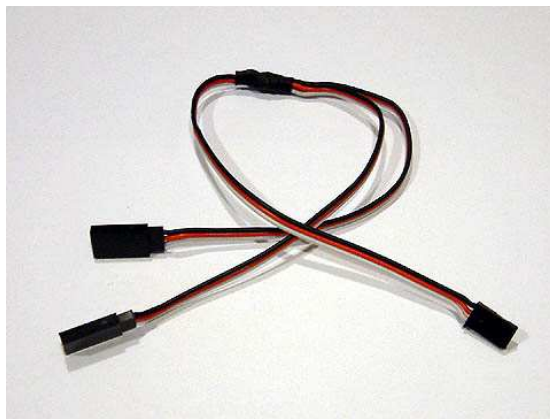


Figure 12. Y Servo Harness [20]

Weather conditions affect the co-pilot operation, and if there are strong winds, the co-pilot may not provide a good stabilization of the aircraft. In such cases the user may want to take the control over, at this point co-pilot offers one channel more for remote switch, thus the user can handle the control of the aircraft.

Specifications are given as follows as presented by manufacturer [19].

Inputs: aileron, elevator, throw, and aux for remote switching.

Outputs: pitch (elevator), roll (aileron).

System weight: 1 oz (28.35 gram)

Sensor dimensions: 1.35" (octagonal), 0.53" thick

Microprocessor: 1.50"x0.89"x0.60"

Power consumption: 5 ma

2.13. Autopilot

UAVs become popular with many kinds of applications in different areas, recently. UAVs used for surveillance, war fighting, fire fighting, reconnaissance, and similar military and civil applications require complex hardware platforms, which include image processing, take-off/landing systems, GPS navigation, flight stabilization systems, autopilot, etc. All of them are called as flight control system (FCS). Autopilot is definitely the most important part of FCS which takes the information provided to it and makes a decision about what to do next.

There are many such autopilot platforms that can be used for UAVs. A list of some of these platforms is provided as follows.

- *ArduPilot: an autopilot based on the Arduino open-source hardware platform*
- *UAV DevBoard: 3 axis IMU-based autopilot platform*
- *Attopilot: an original and commercial autopilot platform*
- *Parallax Basic Stamp: based on commercial Parallax development board*
- *Lego Mindstorm NXT: a commercial so called NXT intelligent brick mostly used in robotic projects*

NXT can be used as an autopilot as mentioned above and it allows many robotic applications with its specific sensors and actuators (see Figure 13).

Communication with external hardware components are provided by I²C bus which is a multi-master serial computer bus that consists of serial data (SDA) and serial clock (SCL).



Figure 13. Lego Mindstorm NXT and its components [21]

NXT has 4 inputs (touch sensor, sound sensor, light sensor, ultrasonic sensor), and 3 outputs (Lego servo motors).

Hardware schematic is shown in the figure below.

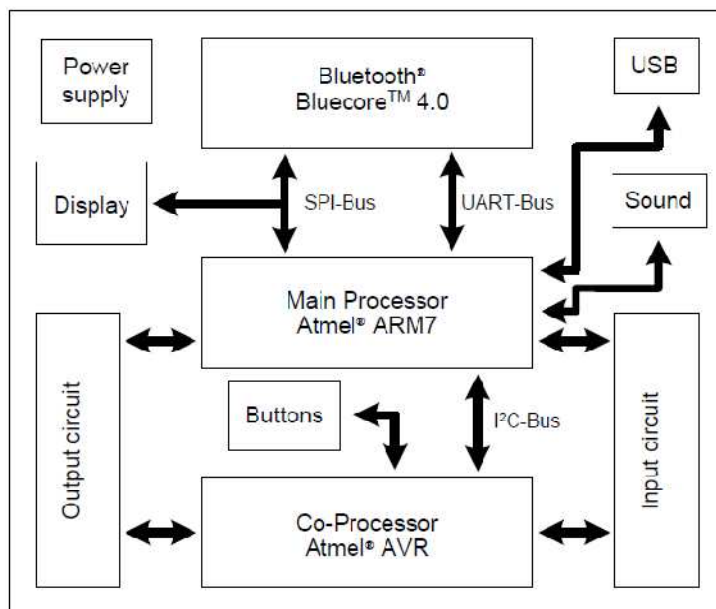


Figure 14. Block diagram of NXT hardware [22]

Technical specifications (cited from web page www.mindstormlego.com)

- 32-bit ARM7 microcontroller
- 256 Kbytes FLASH, 64 Kbytes RAM
- 8-bit AVR microcontroller
- 4 Kbytes FLASH, 512 Byte RAM
- Bluetooth wireless communication (Bluetooth Class II V2.0 compliant)
- USB full speed port (12 Mbit/s)
- 4 input ports, 6-wire cable digital platform (One port includes a IEC 61158 Type 4/EN 50 170 compliant expansion port for future use)
- 3 output ports, 6-wire cable digital platform
- 100 x 64 pixel LCD graphical display
- Loudspeaker - 8 kHz sound quality. Sound channel with 8-bit resolution and 2-16 KHz sample rate.
- Power source: 6 AA batteries

2.14. NXT-G Software

NXT-G is the block programming software for NXT brick (see Figure 15). Many blocks are ready to use are available inside the software. NXT-G allows dragging and dropping blocks over the screen, and uploading the software to NXT brick. So we can just call it as NXT intelligent brick.

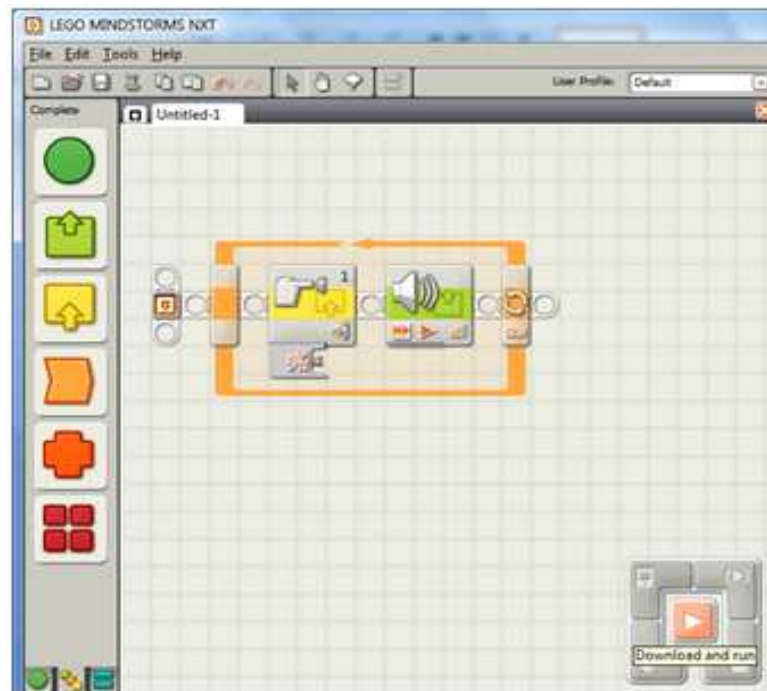


Figure 15. NXT-G retail version

As an example the figure above shows that whenever a touch sensor is pushed, NXT produces sound. Touch sensor block, sound sensor block and loop block were built by firm. So the user just drags the blocks, which he/she wants to use and drops them over the program properly. After NXT is hooked up computer via USB cable or Bluetooth, the program can be run and downloaded to NXT brick.

Here is another example to understand blocks more clearly.

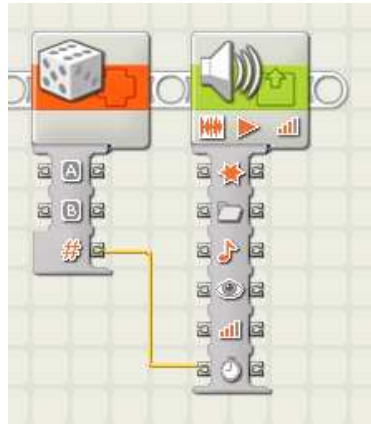


Figure 16. An example with NXT-G blocks

Figure 16 illustrates that blocks wired between them. With analog and digital I/O ports the data can be delivered through the wires. Thus more efficient programs can be created according what it is desired to NXT to do.

2.15. LeJOS NXJ

LeJOS NXJ is a firmware that is supported by NXT intelligent brick and an extension of Java. As it was extended by Java, LeJOS is an open source programming environment that consists of specific classes called LeJOS API. The use of this API makes it easier to program the NXT intelligent brick.

LeJOS allows concurrent programming with the use of threads. Thus the autopilot can execute two or more tasks at the same time such as processing the GPS data, controlling the NXT Prototype Board, etc.

The installation procedure can be performed as described in Appendix B. After the installations steps are followed, the programming IDE Eclipse provides an appropriate programming environment that allows creating new LeJOS codes and uploading them into the NXT intelligent brick.

2.16. NXT Prototype Board

NXT prototype board [23] is a platform developed by HiTechnic which allows someone builds his/her own design or communicating with other devices through NXT. NXT-G prototype programming block (see Figure 17) can be imported as presented by manufacturer and use to read 5 analog inputs and read and write 6 digital ports.

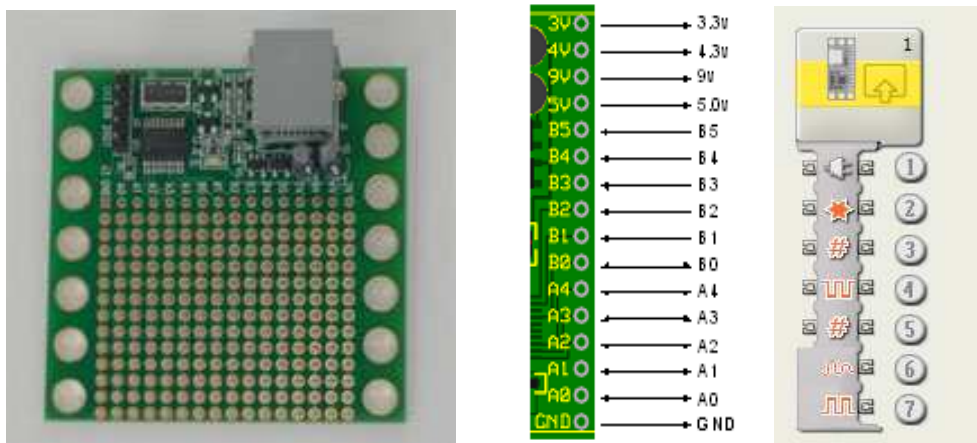


Figure 17. NXT prototype board, connections, and NXT-G block [23]

Since NXT requires I^2C communication, if it is desired to connect a device which uses analog and digital signals (besides I^2C bus which consists of SDA (serial data) and SCL (serial clock) signals), then this prototype board should be used to provide a suitable interface.

3. Methodology

3.1. Project Steps

In order to reach the goal of providing the hardware and the hardware-software interface support to the aircraft platform be able to perform an autonomous flight, this project was divided in the following steps:

1. Measure the co-pilot sensor outputs
2. Study the NXT I/O ports
3. Make the interface between co-pilot and NXT
4. Use a DAC (Digital-to-Analog-Converter) circuit with prototype sensor board
5. Develop a software interface to the NXT for I/O control
6. Study the airplane dynamics to provide the right output to control the servos via the prototype board
7. Combine this project with the navigation control provided by the other group
8. Make a real flight to test the UAV

3.2. Co-pilot Sensor Characteristics and Usage

The co-pilot mentioned in Section 2.7 is composed of two units: sensing unit (infrared sensors) and processing unit (computer). The sensing unit, which is called “co-pilot sensor”, contains infrared sensors that enable to measure roll and pitch angles and sends the information to the processing unit as analog signals through the 4 wires, which correspond to two analog signals, Vcc, and GND (see Figure 25).

The UAV can do both yawing and rolling motion with the use of only ailerons. The analog output of the co-pilot sensor for ailerons is needed to be changed according to the control algorithm implemented in the autopilot. The control algorithm requires the information that comes from the co-pilot sensor. It is necessary because the rolling angle varies by the time and the control algorithm needs to use this data to make a turn. Yawing motion can be provided by processing the data from the co-pilot sensor in the control algorithm implemented in the autopilot. To achieve this, an interface between the co-pilot sensor and the autopilot was required.

In the project, co-pilot sensor outputs had been measured with respect to the angle between co-pilot sensor unit and the earth surface. As the infrared co-pilot sensor provides analog signal as a function of angle to the surface, this information is manipulated by NXT (autopilot) in order to control the servos for the autonomously navigation of the UAV. This data is acquired by the NXT brick via NXT prototype board. The manipulated data is then sent to the co-pilot computer via the same board. The mechanism to manipulate the co-pilot sensor data and use the co-pilot computer with this data is explained in detail in the following of this report.

3.3. Mindstorm NXT as an Autopilot

Mindstorm NXT so called lego intelligent brick is an electronic platform developed and launched by Lego Company [21] and commonly used in robotic applications. NXT is very suitable for the beginners and allows learning robotic applications with its easy to use components easily.

The idea of using NXT in UAVs as an autopilot first came up with Chris Anderson [4] who is the founder of Diydrones.com. He developed 2 UAV autopilot platforms based on NXT. One with HiTechnic Prototype Servo Controller (manufacturer provided only one as a prototype) which both provides an interface from NXT to RC servos and functions as a mux (multiplexer) which enables to switch between autopilot and transmitter, and one with NXT servomotor (connected to RC rudder servo mechanically) and NXT touch sensor (for switching between autopilot and transmitter).

In this project, two solutions inspired from the projects mentioned above were presented with the use either of the HiTechnic NXT prototype board or of a potentiometer with NXT servomotor. With NXT prototype board a Digital-to-Analog Converter (DAC) circuit was developed to convert digital signal to analog since the co-pilot sensor sends analog signals to its computer. However, to be able to use the prototype sensor it was required uploading a software (programmed in LeJOS) to the NXT in order to provide I2C communication with the prototype sensor. The LeJOS programming primitives enable, among other things, to get the product specifications (Product ID, sensor type, and version) and data transferring.

The second alternative used a potentiometer and NXT servomotor to build the interface between NXT and the co-pilot. In this case the NXT brick determines how much the NXT servomotor will turn while the NXT servomotor turns the potentiometer, which is connected to the co-pilot sensor outputs as can be seen in Figure 26.

The autopilot takes the information regarding to UAV's current position via GPS, and compares with the waypoints (pre-programmed destination points), then calculates the heading value which UAV must head to achieve an autonomous flight to the next destination waypoint.

3.4. UAV Control

With the great potential of using UAVs in many applications which can be dangerous for human beings such as fire fighting, reconnaissance, war fighting, among other, the navigation algorithms that provide a smooth autonomous flight gets great importance. The waypoints are pre-programmed or updated in real-time, considering the task of the UAV and the area where the flight would be performed, while the current position is provided by GPS module. This project uses only pre-programmed waypoints.

The dynamics of the UAV is composed of the aileron, the elevator and the rudder panels which are needed for controlling of roll, pitch, and yaw, respectively. An autonomous flight can be provided by taking the controls over either aileron or rudder servos. The UAV is navigated in the air with the aid of autopilot. Autopilot consists of hardware and software components. Hardware is designed in a manner that provides an interface to rudder or ailerons, which are used for heading the airplane. External circuits can be necessary when the signals differ from the ones that Mindstorm NXT can read and provide, such as RC servos need PWM signals to be able to drive it, and autopilot may not supply convenient signals for servos. In such cases autopilot output signal (I²C is common) must be converted to PWM signals or any type of signals as needed. The software part is as important as the hardware one. Besides the navigation algorithm, an interface part has to translate the high-level navigation commands to the adequate values to be sent to the actuators.

4. Experiments and Results

4.1. Co-pilot Measurements

Co-pilot is a flight stabilization system as previously mentioned that operates with respect to infrared temperature difference radiated from the earth. Co-pilot's sensor module sends analog signal to its computer module and aileron and elevator servos are controlled with the signal that is converted to PWM in computer module. Angle of servo arm depends on the signal coming from the sensor module that represents the attitude of aircraft in the air. So that aircraft is intended to be levelled and keep the flight smooth.

Since co-pilot is sensitive to infrared radiation from earth, it doesn't provide correct results indoor applications because of that the walls prevent co-pilot sensors to sense environment. Considering this knowledge measurements were taken inside and outside respectively and following figures were obtained.

Inside measurements

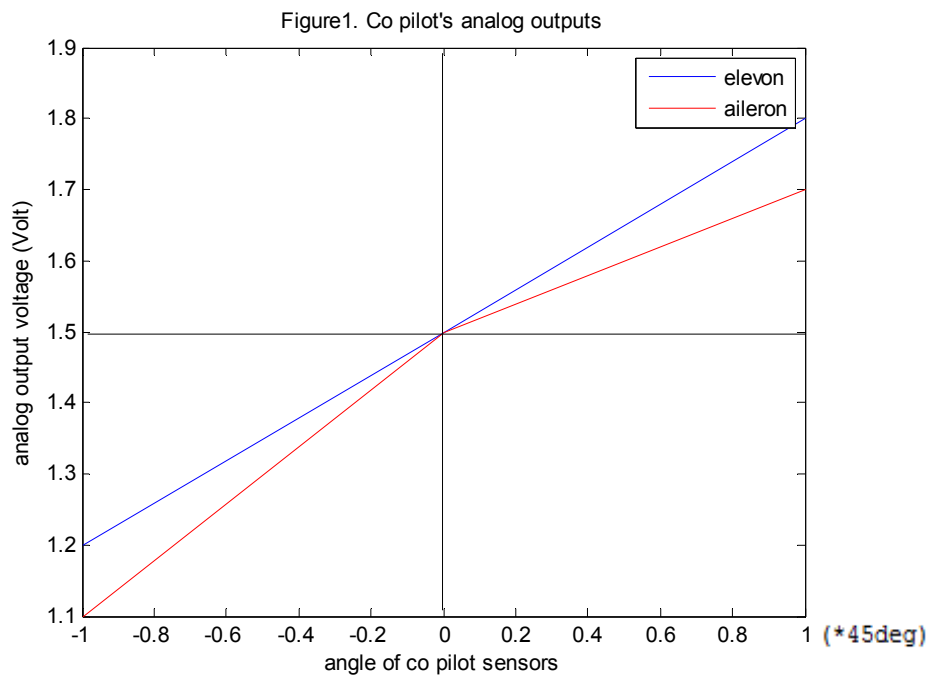


Figure 18. Analog sensor outputs of co pilot for rolling and pitching

Co pilot sensor outputs can be approximated from the figure above as it varies with angle of the sensor axles to the earth surface linearly. Because of that the sensors operate inside imprecisely, aileron servo output was diverged from linearity. However the effect of the walls to work of co-pilot's sensor is as illustrated in Figure 18. It depicts that analog output range of sensor outputs are constricted.

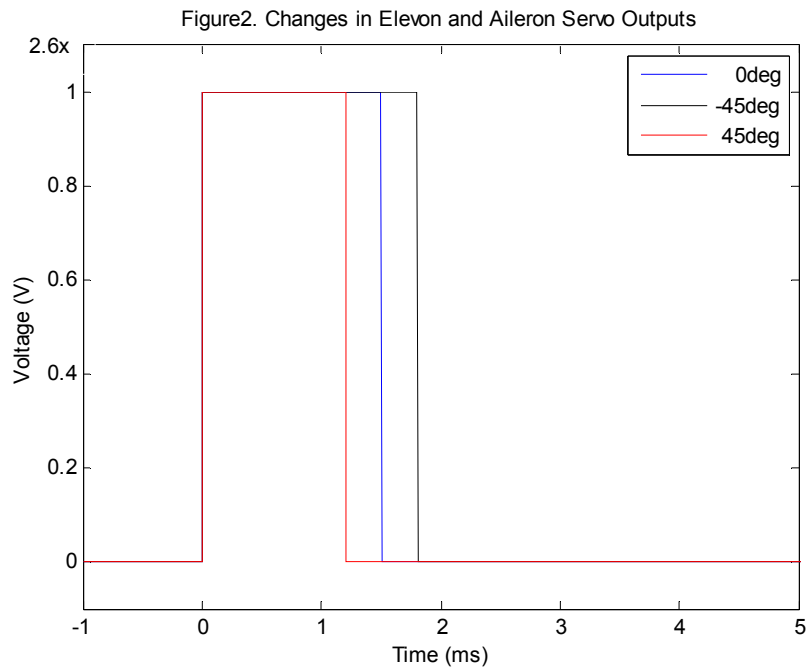


Figure 19. PWM servo outputs of co pilot's computer module

Servo outputs from co-pilot's computer were observed with respect to angle of rotation of sensor module and represented in time domain in the figure above. Black, blue and red lines correspond to 45° , 0° , and -45° respectively. However the pulse width of the PWM signal determines the rotation angle of servo arm either in positive direction or negative. Maximum range of rotation angle is 180° for a conventional RC servo as used in this project.

Outside measurements

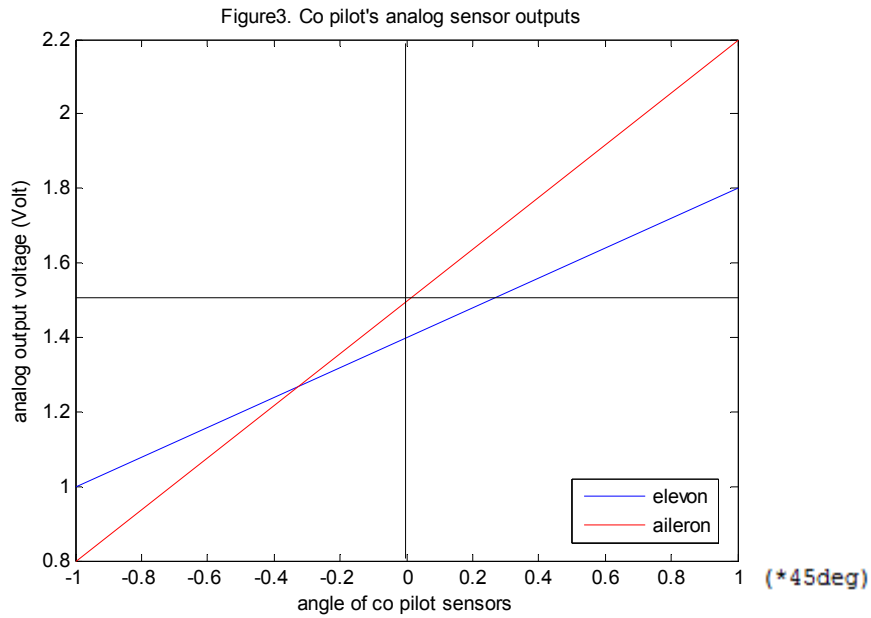


Figure 20. Analog sensor outputs of co-pilot vs. angle of rotation

When the experiment is performed outside, it can be seen that the range of output voltage rises and lies between 0.8-2.2V. Once again elevator servo was measured incorrectly due to the imperfectness of measurement system. 0.8V corresponds to -45° of sensor rotation angle as 2.2V corresponds to 45° .

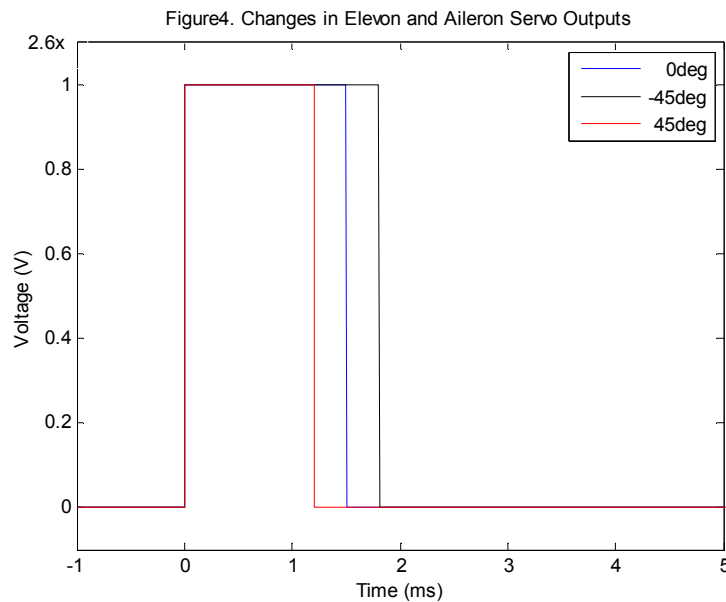


Figure 21. PWM servo outputs from co-pilot's computer in time domain

Aileron and elevator servo reactions are illustrated in figure 21. The pulse width of PWM signal varies between 1.2 and 1.8 with respect to co-pilot’s rotation angle. This pulse is used to control servo arm by changing the width of it.

Concluding, the signals coming from co-pilot’s sensor depend on the environmental factors such as buildings, trees, etc. Also weather conditions affect the co pilot’s performance. Namely if the weather is rainy or snowy, sensitivity of the sensors become more imprecise. So it is important to consider weather conditions before to perform it.

4.2. Understanding the Mindstorm NXT Hardware

Mindstorm NXT is an intelligent brick which allows many applications only functioning as a master in relation to I²C communication with 4 input ports and 3 output ports that are assigned to communicate with sensors, motors, and any external devices developed by the manufacturer, respectively.

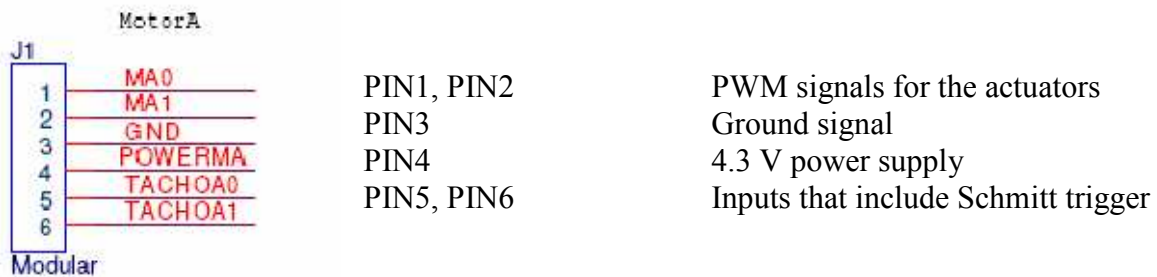


Figure 22. NXT output port [22]

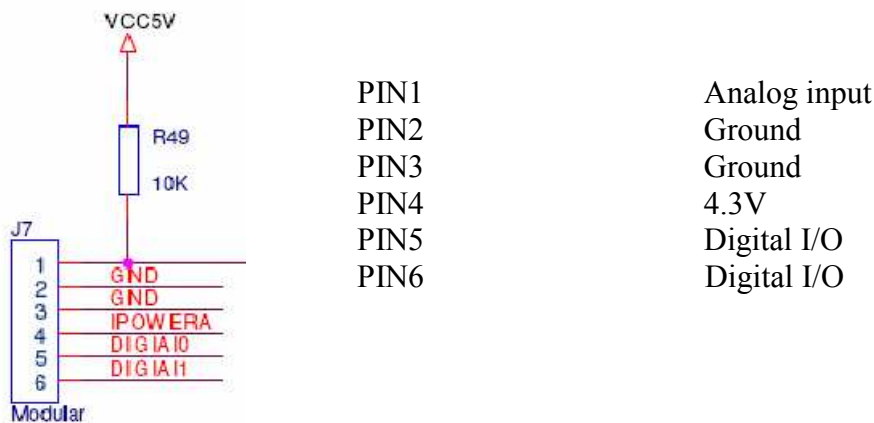


Figure 23. NXT input port [22]

The figures above (22 and 23) show the schematic details of output and input ports. NXT uses 6-wire digital communication interface. Input ports have an analog input which allows making both digital and analog sensors. And input ports 1-3 have a connection with a RS485 controller to manage bi-directional high-speed communication for future devices which require communication at higher speed rates.

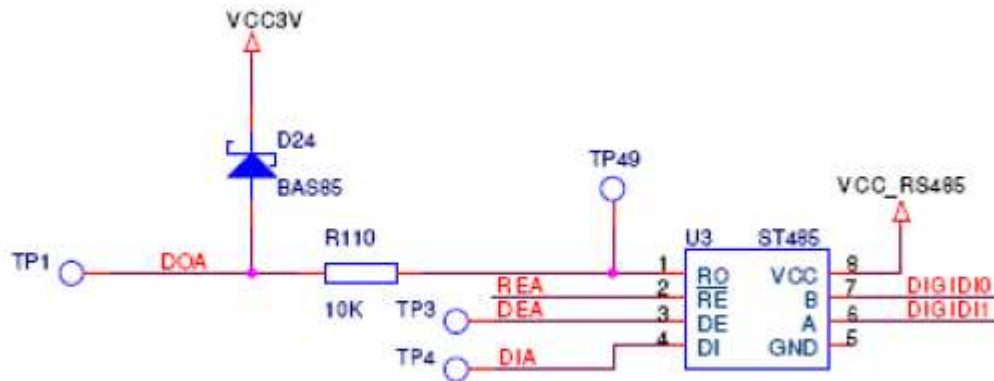


Figure 24. High speed communication hardware schematic

Since I²C communication works as a digital interface for the external devices, there are some advantageous aspects of communicating digitally. Most important one is that in digital devices it is possible to handle device name and various parameters such as start-up time, sensor calibration values, etc. To achieve this each bit or bits in the I²C bus assigned to handle a parameter is interpreted separately.

As a conclusion NXT uses digital interface for communication with external devices and their names must be included in the NXT. Otherwise NXT does not recognize such a device and cannot handle their control. For example, NXT prototype board was used in the project and an NXT-G block was implemented to introduce it to NXT. If another programming language is used beside NXT-G, programmer's code must be included with a few lines concerning the specifications of the device such as name, and so on.

4.3. Providing an Interface from NXT to Co-pilot

As mentioned in Section 2.6, the co-pilot sensor module has 4 pins which consist of analog1, analog2, supply voltage (2.5V), and ground, respectively. Analog1 provides rolling angle, whereas analog2 senses pitching angle.

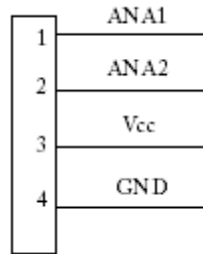


Figure 25. Schematic of co-pilot sensor ports

Since yaw rate can be controlled by both rudder servo and aileron servos, the aircraft can be guided in the air only with aileron servos as well, as it is the case in this thesis.

Co-pilot's analog1 output must be handled and used for yaw control. In order to do this, the mindstorm takes over the control of ailerons by manipulating the data coming from analog1 output. This is possible by the process described in following

This thesis project introduces 2 ways for mindstorm NXT to take over the control of the signal passing through the analog1 port. They are as listed below.

- By using a potentiometer and an NXT servomotor:
A potentiometer is to change the voltage coming from analog1, and an NXT servo motor which takes the control over potentiometer is used to give reaction according to the GPS data processed in the NXT. The connection that makes it possible is presented in Figure 26.

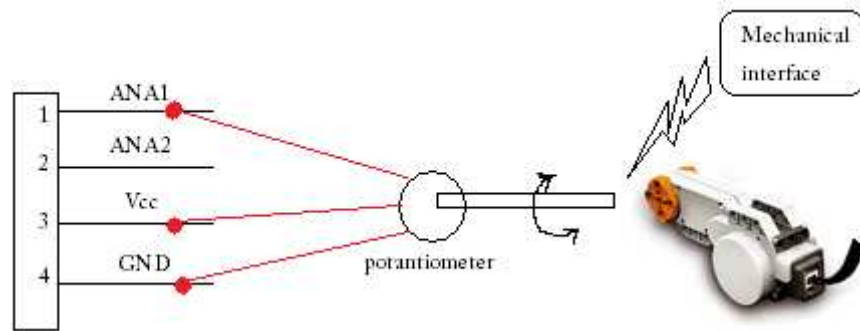


Figure 26. Interfacing with potentiometer

In Figure 26, a potentiometer determines how much the aileron servo arm is rotated. Bi-directional rotation is obtained by wiring potentiometer to Vcc.

- By using NXT prototype board (see Section 2.9):
NXT prototype board provides I²C communication to NXT, as mentioned in Section 2.9. It provides 5 analog inputs, 6 digital outputs, and various supply voltages (3V, 4V, 5V, and 9V). The manufacturer supplies a programming block supported by NXT-G. This block allows taking the control over the I/O ports on the NXT prototype board. So any other software is required to programme the NXT if this block is used. Figure 27 illustrates the connection diagram of the interface between the NXT prototype board and co-pilot and between the board and the co-pilot sensor.

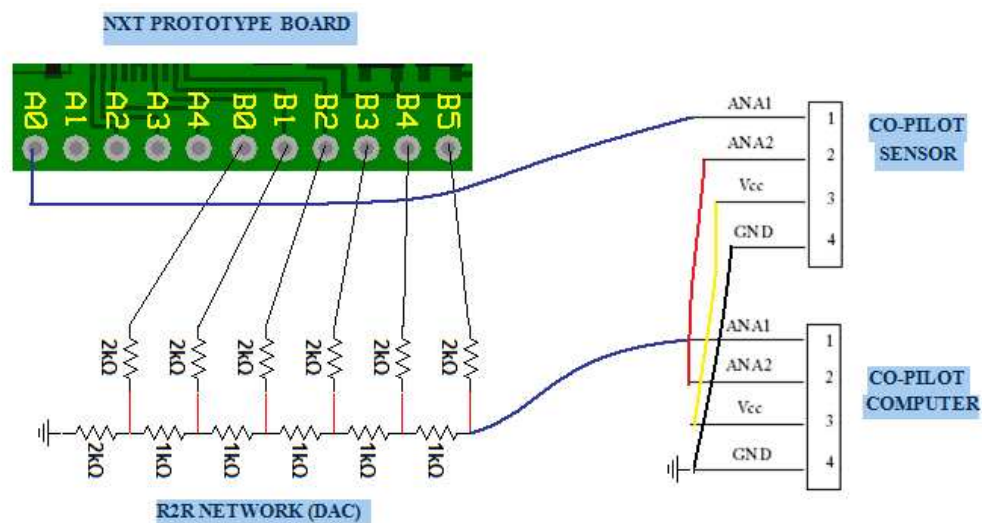


Figure 27. Connections of NXT prototype board as an interface

NXT prototype board takes analog signal from co pilot sensor and gives digital outputs. A DAC (digital to analog converter) is used to change the digital signal to analog signal again as an auxiliary circuit.

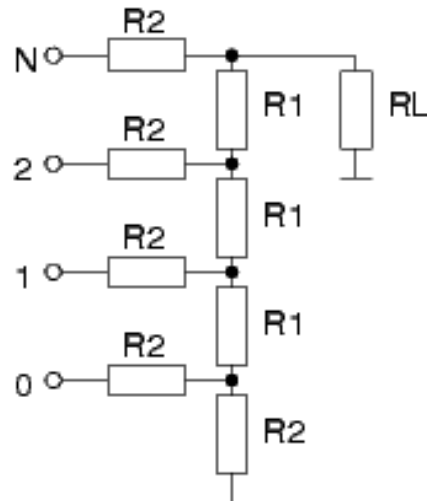


Figure 28. R2R Network ($R2=2*R1$)

Figure 28 illustrates a 4 bit digital-analog converter, which is composed by a R-2R resistance network. When an input either high or low voltage is applied through the inputs from 0 to N (for N bit DAC), the analog output can be observed on the load resistance (R_L represents the co-pilot's internal resistance).

Finally the co-pilot computer can use the analog signals provided to control the aileron servos.

The NXT can be programmed with the NXT-G block, which is supplied by the manufacturer, but also with other software programmed in other languages. In this project, LeJOS (a Java extension for Mindstorm) was used to develop the NXT software used to navigation control, so the software-hardware interface was also developed in this language. This interface, which allows the high-level navigation commands, be translated in signals that will control the servos as already presented above, is detailed in the following.

```

...
4 public class PrototypeSensor extends Thread{
...
11 I2CSensor s = new I2CSensor(SensorPort.S1);
...
16 s.setAddress((byte) 0x01); // address is set here.
...
53 getData(0x42, buf, 1); /* analog input corresponds to upper 8 bits
*/
54 getData(0x43, buf1, 1); /* analog input corresponds to lower 2 bits
*/
56 buf2[0] = (byte) (Math.floor(((buf[0]<<2) | buf1[0])/16.24)+1);
59 sendData(0x4E, ctrl, 1); /* digital I/O control register (ctrl = 0x3f)
...
98 sendData(0x4D, buf2, 1); /* digital output */

```

Listing 1. PrototypeSensor Source Code

The *PrototypeSensor* constructor shown in line 4 of Listing 1 has to be extended using the *I2CSensor* constructor from the LeJOS API to be able to use the commands such as *getData*, *setData*, among other established in the API.

The first action to be done is to set the address to 0x01 in line 16 of Listing 1 to be able to communicate with HiTechnic's prototype board.

The prototype board provides analog inputs that can be handled by reading from 2 registers as 10 bits and 6 bit digital outputs are written via the specific register (as shown in lines 53, 54 and 56 of Listing 1).

Before the data is sent, the digital I/O ports must be assigned as output. This can be achieved writing 0 or 1 bit to the 0x4E register. The value 1 for this bit assigns the I/O ports as outputs whereas value 0 assigns as inputs. (as shown in lines 59 and 98 of Listing 1).

The code presented in Listing 1 enables the NXT to get analog data from co-pilot sensor and send to the digital outputs as a data of 6 bits. The R-2R resistance network gets the bits from

digital outputs (B0-B5) on the prototype sensor board and sends as analog signals through the wire, which is hooked to ANA1 input of co-pilot computer in Figure 27.

4.4. Troubleshooting

In this section the problems encountered during the working process are presented and the tried solutions to them are outlined. Some of the problems were: mounting the propeller motor, interfacing to co-pilot from NXT and some bugs in the software.

The following list summarizes the main problems encountered in this project and the adopted solutions.

- How to provide a stable mounting for the propeller motor?

The motor has a round shaped motor mount whereas it must be an X-shaped motor mount. This specific shape is required because the Taylorcraft450, which was the model airplane platform used in this project, allows only X-shaped motor mount since the manufacturer provides a mounting surface that can only be used for X-shaped models. The way to solve this “interface” problem was to screw the motor mount on the mounting surface of airplane providing a tough assemble.

- Manipulate the co-pilot’s signal for steering with aileron servos, is it possible?

The airplane needs to GPS data to be able to determine its current position and a navigation calculation done by the NXT to decide which direction to point the heading angle, in order to reach the next waypoint. Since yaw control can be achieved by aileron servos, and co-pilot takes the control over the ailerons, an interface between the co-pilot and NXT must be provided.

- What are the possibilities to provide such interface?

Two ways were introduced and explained in Section 3.3. One is provided by a potentiometer and NXT servomotor and the other is by NXT prototype board.

- What are the constraints in two cases?

With a potentiometer, programming of NXT is quite easy to implement. It is done by using a NXT servomotor, which is a specific component of NXT. However, the NXT servomotor only allows a mechanical interface with the potentiometer, which is not very reliable as an electronic interface and requires more space inside the airplane. When NXT prototype board is used, a problem appears regarding how to program it. The manufacturer only offers a program package supported by NXT-G. A Java extension for NXT called LeJOS was used in this project, so it is not possible to explore the manufacturer's program package. The NXT is programmed to be able to communicate with the HiTechnic's prototype board as described in manufacturer's instructions [23].

4.5. Hardware Installation

After the all hardware components experienced in laboratory environment, they were installed inside the airplane considering the centre of gravity of the airplane, which must be within the range given in Section 2.2.3.

Centre of gravity is very important parameter in order to avoid balance problems during the flight. That is why the aircraft can only fly when its weight is balanced and the centre of gravity is right place (see Section 2.2.3). Otherwise it cannot keep flying smoothly, and undesired dangerous conditions may happen.

To be able to make the airplane well-balanced the heavier parts must be located closer to elevator and rudder servos since CG corresponds to this field. So the heaviest hardware part, mindstorm NXT, and the second heaviest component, LiPo battery, were placed as close as it allows to the elevator and rudder servos. Figure 29 shows the complete hardware installation.



Figure 29. Hardware installation inside the aircraft model (Taylorcraft450)

In Figure 29 it is possible to notice the importance of the size of the hardware components used in the project. There is not much place inside the airplane to install too much additional equipments. So the system should be designed considering the airplane model, and how much free space it provides. For instance, looking at the Figure 29 is possible to see that the Lego Mindstorm NXT covers a great part of the free space inside the airplane. Even interfacing the other components with the NXT is not easy, because it fits tight to the free space.

Because of the NXT is heaviest component in the system, it determines the centre of gravity (CG) of the UAV. So it is important to place it considering the change that its addition represents in the CG position.

4.6. Project Integration

This project, which consists of two parts, was carried out by the labour of two groups. The project was divided in so that one part was more oriented to the hardware components, while the other focused in the software components. Each part then was assigned for a group. The work was performed by each group separately during a certain period of time, but the final phase of the project required a co-work of both groups in order to integrate the two parts.

During this collaborative work, two alternatives of interfacing the co-pilot with the NXT Mindstorm were tried, which were explained and had the related problems described in Section 3.4.

First, two different programming codes written in NXT-G and LeJOS were merged according to a version completely programmed in LeJOS.

In the case of using NXT prototype board, it was not trivial to adapt NXT-G code to LeJOS. In the other case, namely the one using NXT servomotor and a potentiometer to take control over the NXT servo in LeJOS, it works fine.

Before the flight test, in order to make sure the system works fine, the software was tested on the ground. This was done in order to tune the system in relation to the ratings of servo rotation, which was done by adjusting the servo sensitivity with the aid of the adjust menu in the transmitter. The sensors should be checked if they need to be calibrated. Then, after checking the system tuning, the airplane would be ready to fly autonomously.

4.7. Flight Test

It is time to enjoy Taylorcraft450's flight. With the aid of multi functional DX6i transmitter, the model airplane takeoffs and landings can manually be performed. Taylorcraft450 allows many basic aerobatic maneuvers with its original design as well as the instructor can. To take controls with transmitter over Taylorcraft450 also is easy only pushing the sticks on it in the desired direction is enough. With LiPo batteries the Taylorcraft450 can keep flying for around 12 minutes. The Taylorcraft450 is constrained by the weather conditions (e.g. it can fly in the winds up to 10 mph) and the maximum takeoff weight (MTOW) which is determined taking into account the factors; wing flap setting, condition of the runway, and the weather conditions [28]. Landings are easily accomplished with a three point state (corresponds to wheels) in safe.

In the case of failure of the Taylorcraft450's Mindstorm NXT autopilot, the user can easily take the control over it through the co-pilot's computer, which enables to control aileron and elevator

servos at any time; also the RMT channel from the co-pilot computer can give full control to transmitter. This provides a safe way to perform the flight.

A real flight could not be performed due to bad weather conditions in the final of the project. However, tests on the ground were performed and they provided good results showing that the commands issued by the navigation program were correctly translated in the actuation of the servos to steer the plane.

5. Conclusions

This thesis reports a hardware system that was designed for a UAV (Unmanned Aerial Vehicle), which enables the control of the airplane actuators in order to assist the autonomous flight. Takeoff and landing systems were not included in the scope of this project. All the hardware components used in the project are introduced in the report.

Off the shelf platforms such as FMA co-pilot as a flight stabilization system, Lego Mindstorm NXT as an autopilot, and Hi-Technic NXT prototype board were used. While using Mindstorm NXT, a problem concerning its I/O ports appeared in relation to the interface with the RC servos. The design provided in this thesis gives a solution for this issue.

The navigation software, the other part of the whole project, was provided by another group. This fact required collaboration between the two groups in order to integrate the navigation software with the software interface that control the servo-mechanisms. Non-trivial problems appeared during this integration, which were solved given as a final result that the UAV was prepared to make an autonomous level flight (takeoff and landing have to be accomplished manually).

However, practical problem during the integration phase made the real flight test unsuccessful, due to some problems in the navigation algorithm. It was identified that those problems were issued by thread coordination timing issues, which were not completely solved by the group responsible for the navigation. Tests on the ground were performed, walking with the airplane and observing if the commands from the navigation program were correctly transformed in movements of the actuators. These tests were enough to show that the control part of the whole work, the goal of this thesis, was successfully achieved.

References

- [1] *NXT Technology Overview* [Online], LEGO Group Corporation, 1932. Available from:
URL: <http://mindstorms.lego.com/eng/Overview/default.aspx>
- [2] *Aeronautics* [Online]. [2005 Jul 27] [cited 2009 Jul 23]. Available from:
URL: <http://virtualskies.arc.nasa.gov/aeronautics/tutorial/motion.html>
- [3] Robert H K, Abhishek H, Jeffrey EQ, Hung HV, Brittiany W *An FPSLIC-based UAV Control Platform* Virginia Commonwealth University, Richmond, Virginia, 2004
- [4] Anderson C. *Original DIY Drones UAV projects* [Online]. Available from:
URL: <http://diydrones.com/profiles/blogs/original-diy-drones-uav>
- [5] Alemdaroglu N. *Unmanned Aerial Vehicles Projects* Middle East Technical University, Ankara, Turkey, 2004.
- [6] *RQ-14 Dragon Eye* [Online]. Available from:
URL: http://en.wikipedia.org/wiki/AeroVironment_Dragon_Eye_RQ-14
- [7] *Bayraktar Mini UAV* [Online]. Available from:
URL: <http://www.baykarmakina.com/MiniUAV>
- [8] *EMT Aladin* [Online]. Available from:
URL: [http://en.wikipedia.org/wiki/Aladin_\(UAV\)](http://en.wikipedia.org/wiki/Aladin_(UAV))
- [9] Available from:
URL: <http://www.lineu.info/blog/wp-content/uploads/2008/09/t-craft8925-05.jpg>
- [10] E-flite Taylorcraft 450 ARF: Assembly Manual, 2007, Horizon Hobby, Inc.
- [11] Spectrum DX6i: Radio Programming Guide, 2007, Horizon Hobby, Inc.
- [12] DX6i radio system, AR6200 receiver and accessories [Online]. Available from:
URL: <http://www.hobbyzone.com/>
- [13] *Brushless DC electric motor* [Online]. Available from:
URL: http://en.wikipedia.org/wiki/Brushless_motor
- [14] Brushless Motor Nippy Black NB 1210/103 [Online]. Available from:
URL: <http://www.uberallmodel.cz/index.php?id=12&lang=en>
- [15] *Electronic Speed Control* [Online]. Available from:
URL: http://en.wikipedia.org/wiki/Electronic_speed_control
- [16] Jeti ECO 18 [Online]. Available from:
URL: <http://www.sloughrc.com/asp/default.asp?itemid=P-JESBECO18>
- [17] *Servomechanism* [Online]. Available from:
URL: http://en.wikipedia.org/wiki/Rc_servo
- [18] Servos [Online]. Available from:
URL: http://www.2dogrc.com/ecommerce/os/catalog/images/blue_servo_72dpi_web.jpg
- [19] Co-Pilot Flight Stabilization System [Online]. Available from:
URL: <http://www.fmadirect.com/detail.htm?item=1489§ion=20>

- [20] *Futaba Servo Y-Harness, 22ga, 12 inch* [Online]. Available from:
URL: <http://www.innov8tivedesigns.com/images/Fut22ga12Y.jpg>
- [21] Lego Mindstorms NXT [Online]. Available from:
URL: <http://news.cnet.com/i/ne/p/2006/16leggo550x322.jpg>
- [22] *Lego Mindstorms NXT Hardware Developer Kit: Figure 1. Hardware block diagram of the NXT brick*, 2006, The LEGO Group.
- [23] *NXT Prototype Board - Solderable* [Online]. Available from:
URL: <http://www.hitechnic.com/cgi-bin/commerce.cgi?preadd=action&key=NPT1050>
- [24] Ailerons [Online]. Available from:
URL: <http://www.grc.nasa.gov/WWW/K-12/airplane/Images/alr.gif>
- [25] Download [Online]. Available from:
URL: <http://www.lejos.org>
- [26] Download [Online]. Available from:
URL: <http://libusb-win32.sourceforge.net/#downloads>
- [27] Download [Online]. Available from:
URL: <http://www.eclipse.org>
- [28] Maximum Takeoff Weight [Online]. Available from:
URL: http://en.wikipedia.org/wiki/Maximum_Takeoff_Weight

Appendix A

Prototype sensor class written in NXT LeJOS was given as follows.

```
import lejos.nxt.*;

public class PrototypeSensor extends Thread{
    private static LRDataBridge db;
    private int sh;
    int count = 0;

    I2CSensor s = new I2CSensor(SensorPort.S1);

    public PrototypeSensor(LRDataBridge tLBDB, I2CPort sensorPort) {

        s.setAddress((byte) 0x01);
        sensorPort.setMode(SensorConstants.TYPE_LOWSPEED_9V);
        s.sendData(0x4F, (byte) 10);

    }

    public void run(){

        while(true){

            db = new LRDataBridge();

/*
            public void getSpec(){

                String id = s.getProductID();
                String type = s.getSensorType();
                String ver = s.getVersion();

                LCD.drawString(id, 0, 0); /* Product ID: HiTechnic
                LCD.drawString(type, 0, 1); /* Sensor type: Proto
                LCD.drawString(ver, 0, 2); /* Version: V1.0

            } */

/* the function provides I2C communication with HiTechnic Prototype Sensor
 * buf1 argument is to manipulate the input value to send to outputs
 */
```

```
byte[] buf = new byte[8];
byte[] buf1 = new byte[8];
byte[] buf2 = new byte[8];
byte[] ctrl = new byte[8];

s.getData(0x42, buf, 1); /* analog input corresponds to upper 8 bits */
s.getData(0x43, buf1, 1); /* analog input corresponds to lower 2 bits */

buf2[0] = (byte) (Math.floor(((buf[0]<<2) | buf1[0])/16.24)+1);

ctrl[0] = 0x3F; /* All digital outputs are assigned as outputs */
s.sendData(0x4E, ctrl, 1); /* digital I/O control bit */

sh = db.getjump();

    if ( sh == 1){
        LCD.drawString("LEFT", 0, 6);
        buf2[0] += 10;
        s.sendData(0x4D, buf2, 1);

        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {}
    }

    else if (sh == -1){
        LCD.drawString("RIGHT", 0, 6);
        buf2[0] -= 10;
        s.sendData(0x4D, buf2, 1);

        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {}
    }

    else if (sh == 0){
        LCD.drawString("Straight", 0, 6);
        s.sendData(0x4D, buf2, 1);
    }

}

}
```


Appendix B

The following steps describe the instructions to use LeJOS.

1. Install the Java Software Development Kit (SDK) of Sun Microsystems (version 1.5 or later)
2. Download the latest version of LeJOS from [\[25\]](#), unzip the contents into a directory and install it.
3. Create the environmental variables via Start > Control Panel > System. Click the Advanced tab, then Environment Variables.
4. Create a new environmental variable by clicking New. Type NXJ_HOME as variable name, and add the LeJOS directory to the path as variable value.
5. Download and install the USB drivers from [\[26\]](#).
6. Hold the reset button underside of the NXT brick to delete the current firmware and put it into the firmware upload mode. When NXT produces a pulsing sound, plug in USB and type `nxjflash` in the command line. After a short time NXT brick will be ready to use.
7. Download the eclipse from [\[27\]](#).
8. Unzip the files into a directory and run eclipse.exe.
9. Create a new project by clicking on File > New > Project > Java Project. Type a name for the project and click on Finish.
10. Add the LeJOS classes to the Eclipse classpath. Select Project > Properties. Click on Java Build Path on the left panel and select Libraries tab. Add the classes.jar which was located in LeJOS directory as external jars.
11. Create the tools for compiling, transferring the code, and exploring the files. For compiler tool select *Run > External Tools > External Tools Configurations*. Double click on the Program in the opened window. Click on *Browse File System* under *Location* and browse to the `\bin\nxjc.bat`. Under Working Directory click *Browse Workspace* and select the current project. Click the variables under *Arguments* and select `java_type_name`. Do the same for the other tools used for transferring the code and exploring the files. Only create a new configuration for each tool and browse to the `\bin\nxj.bat` and `\bin\nxjbrowse.bat` (there are no arguments for this tool), respectively.

12. Select *Run > External Tools > Organize Favorites*, click Add on the new window and select all three of the LeJOS tools and click OK.