Aleksi Olkkonen

A Quadcopter Flight Controller

VAASAN AMMATTIKORKEAKOULU

Tietotekniikka
2018

TIIVISTELMÄ

Tekijä	Aleksi Olkkonen
Opinnäytetyön nimi	A Quadcopter Flight Controller
Vuosi	2018
Kieli	Englanti
Sivumäärä	45
Ohjaaja	Jani Ahvonen

Opinnäytetyö tehtiin oppimis- ja harrastustarkoitukseen ja se sisältää keskeisimmät asiat jotka tulisi ottaa huomioon nelikopterin rakentamisessa ja ohjelmoinnissa. Työssä tutkitaan myös mitä rajoitteita ja mahdollisuuksia LoRa-teknologia tarjoaa nelikopterin ohjaamiseen. Opinnäytetyössä suunniteltiin ja rakennettiin nelikopteri sekä nelikopterin ohjain, ohjelmoitiin lennonohjausjärjestelmä Arduino-järjestelmille ja hyödynnettiin LoRa-teknologiaa ohjaimen ja nelikopterin välisessä kommunikaatiossa. Työssä rakennettiin piirilevy, johon asennettiin mikrokontrolleri, anturit ja muut oheislaitteet.

Lennonohjausjärjestelmä on ohjelma joka laskee antureiden mittausarvoista nelikopterin asennon, vastaanottaa ohjaimelta komentoja ja käyttää PID-säädintä ohjausasennon laskemiseen. Moottoreiden nopeutta säädetään ohjausasennon perusteella.

Lennonohjausjärjestelmän tämänhetkinen toteutus on vajavainen. Laitteisto, LoRa-kommunikaatio ja suurin osa lennonohjausjärjestelmän ominaisuuksista toimivat, mutta järjestelmä ei pysty vakauttamaan nelikopteria asianmukaisesti.

Avainsanat Nelikopteri, sulautetut järjestelmät, säätötekniikka, LoRa

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Bachelor of engineering (IT)

ABSTRACT

Author	Aleksi Olkkonen
Title	Quadcopter flight controller
Year	2018
Language	English
Pages	45
Name of supervisor	Jani Ahvonen

The thesis was made for learning and hobby purposes. It can be used as a reference what things should be considered when building and programming a quadcopter. The thesis also covers what limitations and possibilities LoRa technology introduces to controlling a drone. The scope of the thesis was to design and build a quadcopter and a joystick, implement a flight controller on Arduino systems and use LoRa technology to establish communication between the quadcopter and the joystick. A custom circuit board was built to connect the microcontrollers, sensors and other devices together and to provide them a supply voltage.

The flight controller is a program that reads user input and sensor measurements to calculate the current and target orientation of the quadcopter. It uses a PID controller to calculate the control orientation of the quadcopter and controls the motor speed accordingly. The joystick communicates to the quadcopter using LoRa technology and one element of the thesis was to find out if it is a suitable technology for this use case.

The current implementation of the flight controller is incomplete. The hardware, communication and most of the flight controller's features are working, but the flight controller is not able to stabilize the quadcopter properly.

Keywords Quadcopter, embedded systems, control theory, LoRa

TIIVISTELMÄ
ABSTRACT
INDEX OF FIGURES
INDEX OF TABLES

TABLE OF CONTENTS

1 Terms and abbreviations....8
2 Introduction....9
3 Requirements specification....10
3.1 Hardware....10
3.2 Software....11
4 Flying the quadcopter....12
4.1 Flight modes....12
4.2 Quadcopter maneuvers....13
5 Hardware....14
5.1 Component list....14
5.2 Motors....15
5.3 Electronic speed controllers....15
5.4 Battery....16
5.4.1 Safety....17
5.5 Propellers....17
5.6 Mass / thrust ratio....18
5.7 Sensors....18
5.7.1 Accelerometer....19
5.7.2 Gyroscope....20
5.7.3 Magnetometer....20
5.7.4 Barometer....20
5.7.5 Sensor configuration....21
5.8 Microcontroller....21
5.9 Usage temperatures....22
5.10 Frame....23
6 Flight controller....25
6.1 Overview....25
6.2 Flight controller circuit board....25
6.3 Software Architecture....26
6.3.1 Flight controller....26
6.3.2 LoRa joystick....26
6.3.3 Communication....26
6.3.4 Joystick....27
6.3.5 IMU....27
6.3.6 Motor controller....27
6.4 Serial peripheral interface....27
6.5 Calculating the control orientation....28
6.5.1 Complementary filter....28
6.5.2 PID-controller....29

6.6 Controlling quadcopter orientation....31
6.6.1 Pulse width modulation....31
6.6.2 Calculating motor rpm....32
6.7 Flight controller commands....32
6.7.1 Flight controller configuration....32
6.8 Flight controller safety....33
7 Joystick....34
7.1 Joystick circuit board....35
7.2 Joystick functionality....35
7.2.1Interpreting the joystick state....36
8 Communication....37
8.1 Lora protocol....37
8.2 Joystick and quadcopter communication....37
8.3 Packet validity and security....37
8.4 Performance....38
9 Drone flying and the Finnish law....39
10 Current state of the project....40
10.1 Current implementation....40
10.2 Required improvements....40
10.2.1 Flight controller loop frequency....40
10.2.2 Yaw calculation....41
10.2.3 PID controller tuning....41
11 Conclusions....42
12 References....43

INDEX OF FIGURES

Drawing 1: Squence diagram of the system.....10
Drawing 2: Pitch, roll and yaw [1].....12
Drawing 3: Quadcopter flight configurations [3].....13
Drawing 4: Brushless DC motor [4].....15
Drawing 5: 3-phase brushless DC motor steps [6].....16
Drawing 6: Two bldaded propeller [8].....18
Drawing 7: Accelerometer [12].....19
Drawing 8: 850 mAh LiPo battery discharge at different temperatures [17].....23
Drawing 9: Drone's components from side view.....24
Drawing 10: Flight controller circuit board.....25
Drawing 11: SPI protocol [19].....27
Drawing 12: Complementary filter [20].....29
Drawing 13: PID controller configured with different porpotional term values [22] 30
Drawing 14: PWM signal at different duty cycles [25].....31
Drawing 15: Joystick viewed from inside.....34
Drawing 16: Joystick circuit board.....35

INDEX OF TABLES

Table 1: X configured quadcopter maneuvers.....13
Table 2: List of the components.....14
Table 3: Sensors' measurement ranges [13].....21
Table 4: Components' usage temperature ranges.....22
Table 5: All of the IMU's SPI signals utilized by the flight controller.....28
Table 6: Joystick commands.....33

1 TERMS AND ABBREVIATIONS

BEC	Battery Elimination Circuit
(C)CW	(Counter) Clockwise
CRC	Cyclic Redundancy Check
Drone	A general name for different type of multicopeters
ESC	Electronic Speed Controller
FPV	First Person View
IMU	Inertial Measurement Unit
Kv	rpm/V – rounds per minute per volt
LiPo	Lithium Polymer battery
LoRa	Long Range wireless communication technology
PDB	Power Distribution Board
Pitch	Rotation around X-axis
PWM	Pulse Width Modulation
Roll	Rotation around Y-axis
SPI	Serial Peripheral Interface
Yaw	Rotation around Z-axis

2 INTRODUCTION

The scope of the thesis was to design and build a quadcopter and a joystick, implement a flight controller on Arduino systems and use LoRa technology to establish communication between the quadcopter and the joystick.

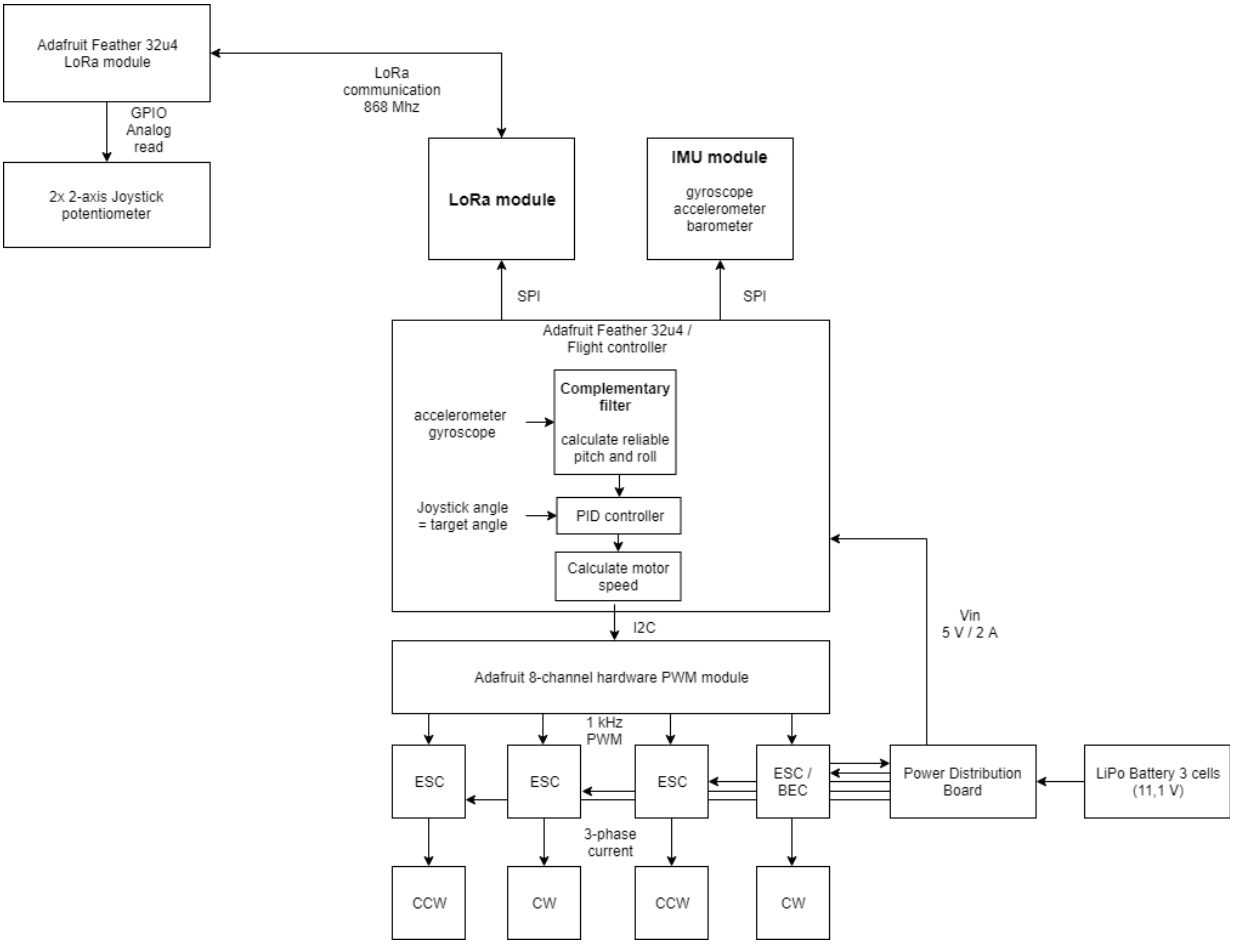
A drone is an unmanned flying vehicle. The term is generally used to describe all kinds of multicopters. A quadcopter is a multicopter that has four motors. A drone is usually controlled with a joystick, but it can also use GPS or other technologies to navigate from a location to another. A drone needs a flight controller that can calculate and control motor speed based on user input and sensor measurements. The flight controller can provide different features such as normal and assisted flight modes, GPS navigation and safe landing.

LoRa was chosen as the communication technology because it is a new and popular technology in Internet of Things applications. Part of the thesis was also to find out if it is a suitable technology for quadcopters, which require fast packet transmission and low delay. Due to long range communication capabilities of the LoRa technology, it could also be used to control self-sustaining drones.

I would like to thank VAMK, University of Applied Sciences for supporting the component purchases and allowing me to do this interesting thesis project. I would also like to thank Jani Ahvonen, who is my thesis supervisor and especially my father who helped me a lot with understanding hardware design.

3 REQUIREMENTS SPECIFICATION

The Drawing 1. defines the main components of the system. The joystick and the drone communicate by using LoRa technology. A RFM95 LoRa module is integrated to Adafruit Feather 32u4, which is the microcontroller used in both devices.



Drawing 1: Squence diagram of the system

The joystick reads 3-axis potentiometer values to determine the joystick state and sends it to the quadcopter. The flight controller parses the LoRa packet as a joystick state and uses the state as the target orientation. Sensor measurements are read to estimate the current orientation of the drone. The control orientation is calculated with a PID controller and the output is used to generate a PWM signal for the electronic speed controllers. The ESCs will control the speed of the motors according to the PWM signal. The ESCs and motors are powered by a LiPo battery. One of the ESCs has an integrated battery eliminaction circuit that provides a supply voltage for the quadcopter.

3.1 HARDWARE

All of the hardware components need to be compatible with each other. The

battery needs to be able to ouput enough current for the electronic speed controllers and motors. A circuit board and power distribution board are needed to provide a supply voltage for all of the components. The microprocessor needs to be able to communicate with the sensors and LoRa module and output a PWM signal for the ESCs. The flight controller's update frequency should be high enough to control the quadcopter in a smooth manner.

3.2 SOFTWARE

The flight controller will be implemented on Arduino systems. It should be able to communicate with the joystick using LoRa technology, read sensor measurements to estimate the current orientation and control the speed of the motors.

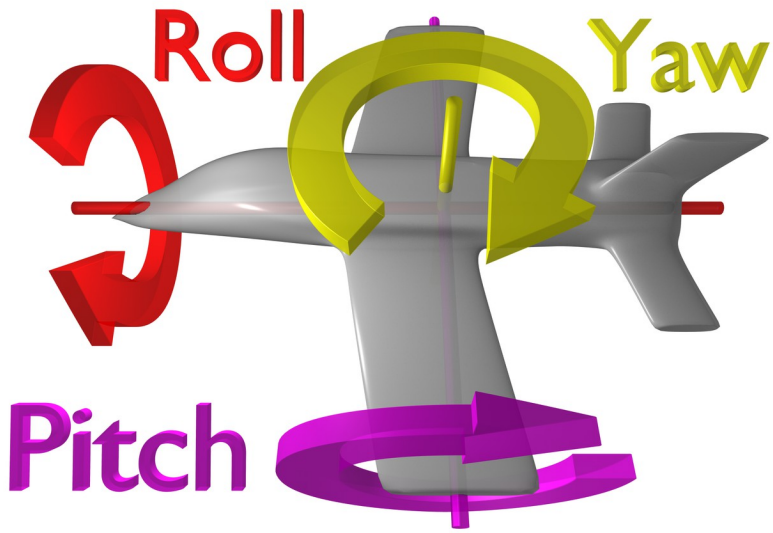
The flight controller implements a self-leveling flight mode. The user should be able to configure PID controller terms and other parameters with the joystick. The flight controller needs to communicate with the sensors and LoRa module using a Serial Peripheral Interface bus. The LoRa module needs to support the legal hobbyist frequencies used in Finland.

4 FLYING THE QUADCOPTER

4.1 FLIGHT MODES

Quadcopters usually support at least two flight modes: acro and self-level mode. The self-level mode is implemented in the thesis because it is easier to control and helps with the limitations of the LoRa communication.

A quadcopter can rotate around X, Y and Z axes. Rotation around the X-axis is called the pitch, rotation around the Y-axis is called the roll and rotation around the Z-axis is called the yaw.



Drawing 2: Pitch, roll and yaw [1]

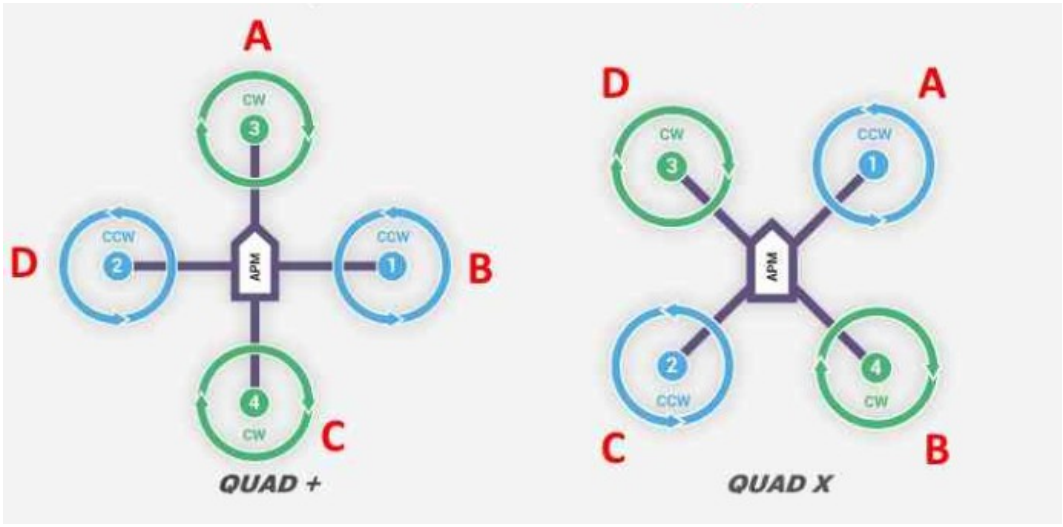
In the acro mode the pitch, the roll and the yaw control angular velocity of the quadcopter. The angular velocity is measured with a gyroscope in degrees per second. User input will increase or decrease the quadcopter's angular velocity and the user needs to manually bring the quadcopter to the neutral orientation. [2]

In the self-level mode the user controls the quadcopter's orientation directly. If the user does not input any pitch, roll or yaw the flight controller will attempt to bring the quadcopter into the neutral orientation. Pulling the joystick to some direction will change the quadcopter's target orientation on that axis.

4.2 QUADCOPTER MANEUVERS

A quadcopter has four different maneuvers which the user can control with the left and right joysticks of the controller. Maneuvers are achieved by spinning the motors at different speeds.

There are two different configurations for a quadcopter - a plus and a X configuration [3]. The plus configuration has one motor on each side. The X configuration has two front motors and two back motors. The shape of the Spendix S250Q frame used in this thesis does not allow implementing a plus configuration, because the motors are closer to each other on the Y-axis than they are on the X-axis (drawing 3.).



Drawing 3: Quadcopter flight configurations [3]

Table 1. describes motor control in the X configuration.

Command	Joystick	Motor speed	Maneuver
Throttle	Left joystick Y-axis	Each motor spins at equal speed	Increases or decreases altitude
Roll	Right joystick X-axis	Motors A and B spin at different speed than motors D and C	Move left or right
Pitch	Right joystick Y-axis	Motors A and D spin at different speed than motors B and C	Move forwards or backwards
Yaw	Left joystick X-axis	CW motors spint at different speed than CCW motors	Rotate left or right around drone's local Z-axis

Table 1: X configured quadcopter maneuvers

5 HARDWARE

5.1 COMPONENT LIST

In Table 2. there is a list of the most important hardware components used in the project. The total price of all components is 338,07 €. Most of the purchases were kindly supported by VAMK, University of Applied Sciences.

The drone's are trending and there are various products at drastically different price points. Manufacturers build toy, racing and professional video drones. The toy drones can be as cheap as 15 €. The drone built in this thesis is a racing drone and their price can range from 100 to 800 €. Home built drones usually cost more than prebuilt drones because it is more expensive to buy components separately than to manufacture professionally using optimal components. However, it is easier to fix or replace broken parts after building the drone by yourself and having every replacement component available.

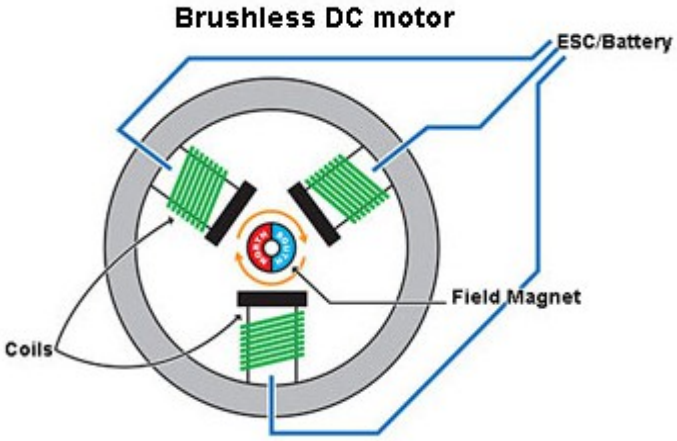
Part	**Description**	**Price (€)**
Motors	2x CCW and 2x CW EMAX MT2204	68,00
Electronic speed controllers	4x Emax 20A ESC with BEC	44,00
Propellers	4x 6"4 propellers	4,40
Battery	2x Gens ace 2200 mAh 3 cell, 11.1 V, 25C LiPo battery	59,80
Charger	GT Power B3 2-3s LiPo charger	14,00
Microcontroller	2x Adafruit feather 32u4 RFM95	57,32
Sensors	PmodNAV 9-axis IMU	27,13
Frame	Spedix S250Q	26,13
Circuit board	A custom made circuit board to connect PmodNav, microcontroller and other devices	12,30
Power distribution board	4-axis Electric Connection Plate Quad Power Distribution Board	5,70
PWM module	Adafruit 8-channel servo PWM module	11,20
Joystick battery	ValueLine power bank 2200 mAh	8,09
		338,07

Table 2: List of the components

5.2 MOTORS

A quadcopter requires two clockwise motors and two counter clockwise motors. Having motors that spin in different directions and are positioned properly will eliminate torque in a direction which would cause the quadcopter to yaw to the opposite direction.

Brushed and brushless DC motors are two common types of motors used in quadcopters. Brushless motors are more popular because they are smaller, offer more power, higher efficieny and have less mechanical wear. However they are more expensive and they require an electronic speed controller.



Drawing 4: Brushless DC motor [4]

In a 3-phase brushless DC motor the electronic speed controller controls the motor by rapidly switching the current input for each of the coils. This creates a rotating magnetic field around the coil and spins the magnet. The magnet is attached to a shaft and causes it to spin. A propeller is attached to the same shaft and the thrust generated by the spinning propeller lifts the quadcopter in the air [4].

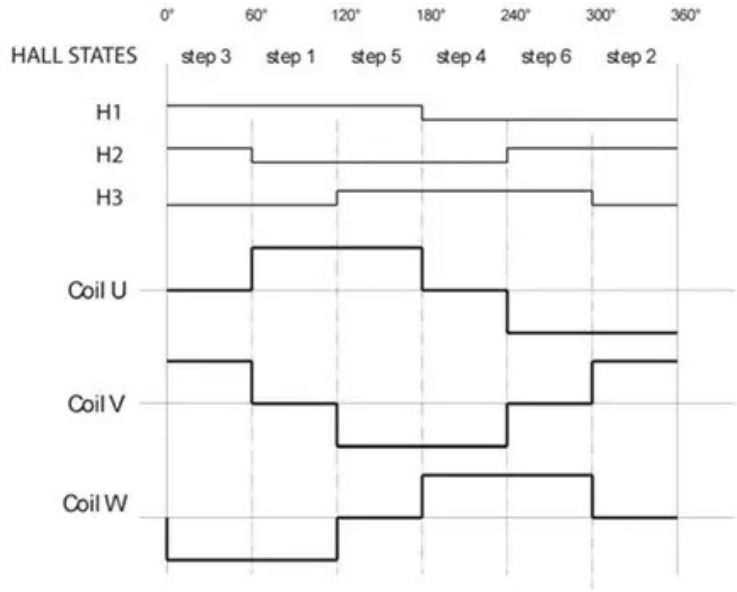
A motor with lower Kv rating can produce higher torque and is able to spin a bigger propeller than a motor with higher Kv rating. Racing size quadcopters use small propellers, motors with higher Kv rating and therefore they maneuver faster but can lift less mass than heavy drones. Kv is equivalent to rpm/V, that is rounds per minute per voltage when there is no load attached to the motor. The more input voltage is supplied to the motor the faster it spins. Emax MT2204 motors are used in this project. Their Kv rating is 2300. Drawing full voltage output from a 11.1 V battery results into 25 530 rpm with no load attached. In practice their rpm is slower due to drag and properties of the propellers being used.

5.3 ELECTRONIC SPEED CONTROLLERS

Electronic speed controllers control the current flow from the battery to the motors according to a pulse width modual signal. The PWM signal determines the

voltage and current flow to the coils which affects the speed and torque of the motors. Hall-sensors are used to determine position of the rotor and accurately control rotation of the magnetic field by energizing three coils of a brushless motor in a specific order. [5]

The drawing 5. shows how coils U, V and W are energized in different phases according to the states of Hall-sensors H1, H2 and H3. In each step one coil is in neutral state, one is in positive state and one is in negative state. This forces the motor to spin into a specific position. The next step coils are energized in a different order and the motor spins to the next position. Total of six steps are required to spin the motor 360 degrees.



Drawing 5: 3-phase brushless DC motor steps [6]

ESC's maximum current rating must be equal to or higher than the maximum current draw of the motor. Emax BLHeli 20 A ESC has high enough current rating for Emax MT2204 motor that requires 11.5 A current at the maximum thrust. It also has a built in battery elimination circuit that can be used to get a supply voltage for the microcontroller and other devices in the quadcopter. Emax BLHeli ESC provides a supply voltage of 5 V / 2 A.

Most ESC firmware can be configured with parameters such as brake type, motor timing mode, motor control frequency and low voltage protection. These will affect flight properties of the quadcopter. Configuration can be done by using the joystick or an ESC programming toolkit and software.

5.4 BATTERY

Emax MT2204 motor requires a 11.5 A current output at the maximum thrust. Spinning all of the four motors at the maximum thrust requires 46 A current output from the battery. Gens Ace 3S LiPo battery is used in this project. It has

three cells, each having nominal voltage of 3.7 V, which results into total of 11.1 V. Its 2200 mAh capacity provides roughly 5-12 min flight time.

LiPo batteries are used because they have a high discharge rate and are relatively compact in size. The battery's discharge rate, C, describes how fast it can discharge and how much current it can output [7]. If the battery's capacity is 2200 mAh and the discharge rate is 1 C, the battery will deliver its capacity in an hour outputting 2.2 A. Therefore, the battery used in this project has a discharge rate of 25 C, so that it can output atleast the maximum current of 46 A required by the motors. A 2200 mAh capacity with a discharge rate of 25 can output 55 A and discharge the battery in 2.4 minutes. However most of the time the motors do not draw the maximum current and estimated flight time is a few minutes longer.

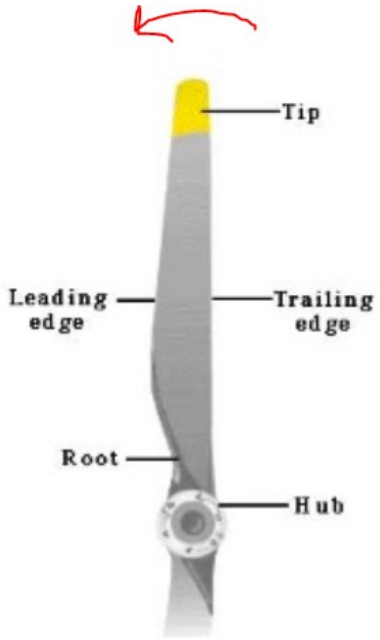
5.4.1 Safety

A user needs to be careful when charging, discharging and storing a LiPo battery because improper use of the battery can be dangerous. A damaged battery can start a fire and it burns at hot temperatures. The battery should be kept in a LiPo safety bag when it is being stored or charged. If there is visible expanding on the battery's cells or the battery does not charge, it should be disposed properly. The battery charger should be a high quality charger designed for LiPo batteries. It should indicate when the battery is fully charged and stop charging when the batteries are fully charged or damaged.

5.5 PROPELLERS

The motor's datasheet is the best reference for choosing an optimal propeller for the motor. The propeller length must be suitable for the quadcopter's frame size so that the propeller's tip does not get too close to another propeller or the frame. The propeller's length is measured from tip to tip. Emax MT2204 motor is designed for 5 to 6 inch propellers. A longer propeller or a propeller with higher pitch produces more thrust but requires more torque. Pitch determines the angle of the curve between the leading and trailing edges.

It is also important to choose a propeller according to the motor's direction. When the motor is spinning, the leading edge of the blade must be leading the trailing edge to the direction of movement. The leading edge is higher than the trailing edge and therefore the propeller pushes air downwards and lifts the quadcopter in the air.



Drawing 6: Two bldaded propeller [8]

A propeller can have different number of blades. Most commonly two or three bladed propellers are used. Propellers with two blades are more efficient than three bladed propellers. They require less torque but produce less thrust and therefore do not maneuver the quadcopter as fast.

5.6 MASS / THRUST RATIO

Emax MT2204 product description [9] defines that the maximum thrust for a 6”3 propeller is 440 g. All four motors combined the thrust is 1760 g. As a rule of thumb the maximum mass of the quadcopter should be half of the total thrust. The mass can be even lower to achieve faster maneuverability. The mass of the quadcopter built in the thesis is roughly 650 g which is lower than the suggested maximum of 880 g.

5.7 SENSORS

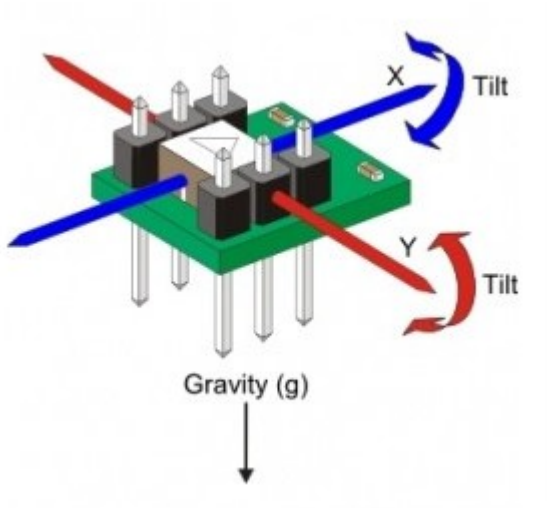
Different types of sensors are required to estimate the quadcopter's orientation. The self-leveling mode requires an accelerometer, gyroscope and magnetometer. Other sensors can be used to provide various types of measurements such as atmospheric pressure or distance which can be used to maintain the altitude.

PmodNav [10] is used in the thesis. It has a LSM9DS1 chip that contains a 3-axis accelerometer, gyroscope and magnetometer. Additionally it has a LPS25HB chip that contains a barometer.

5.7.1 Accelerometer

Accelerometer provides linear acceleration on three axes in g. Value of g is 9.81 m/s^2 and it is equivalent to acceleration caused by earth's gravitational force. Due to the gravitational force the accelerometer measures 1 g when it is not moving. If the orientation of the sensor is neutral, 1 g is measured on Z-axis. Once it rotates around X or Y-axis, acceleration can be measured also on other axes. From this data it is possible to calculate the accelerometer's pitch and roll – the rotation around the X and Y axes. [11]

Rotation around an axis that is pointing to the same direction as the gravitational force vector cannot be measured. For example in the neutral orientation the Z-axis is pointing to the same direction as the gravitational force vetor. Rotation around the Z-axis does not affect the acceleration measured on X or Y axes and therefore it results to 1 g on Z-axis.



Drawing 7: Accelerometer [12]

The quadcopter's orientation can be calculated from the accelerometer data using the following equations:

```
pitch = atan(accelerometer.Y, accelerometer.Z) * 180 / Pi
roll = atan(accelerometer.X, accelerometer.Y) * 180 / Pi
```

Rotation around the X-axis is called the pitch and it can be measured on the Y-axis. Rotation around the Y-axis can be measured on the X-axis and it is called the roll. Both values are converted from radians to degrees. Accelerometer is easily affected by noise, such as the quadcopter's motors causing the frame to shake. Therefore it is important that the accelerometer is installed on a noise reduction material such as rubber, and that it is used together with a gyroscope and a complementary filter.

5.7.2 Gyroscope

The gyroscope measures angular velocity on three axes in dps (degrees per second) [13]. The quadcopter's orientation cannot be calculated from a single sensor measurement like it can be with the accelerometer. When the quadcopter has no angular velocity on an axis, the gyroscope measures 0 dps. When the quadcopter begins to rotate around an axis, the gyroscope will measure the angular velocity and it will be integrated by the microcontroller to estimate the quadcopter's orientation. [14]

```
pitch = pitch + gyroscope.X * deltaTime
roll = roll + gyroscope.Y * deltaTime
yaw = yaw + gyroscope.Z * deltaTime
```

This way the gyroscope can be used to measure the offset from its original rotation around an axis. Theoritically the original rotation is 0 degrees pitch and roll when the quadcopter is placed on a flat surface. Delta time is used as a multiplier because the quadcopter's flight controller loop executes n amount of times in a second, and the measured unit is degrees per second. The faster the angular velocity can be measured and integrated the more accurate the result is. However there will be always minor inaccuracies due to the sensor's refresh rate, measurement accuracy and floating point operations. This will result into drift which can observed when the quadcopter rotates away from the original rotation and retuns back to the original rotation - the calculated orientation will have some offset from the original position. When the integration is run for a longer time these inaccurasies can increase and decrease.

The gyroscope is not prone to noise caused by oscalliation of the quadcopter's frame and, therefore, it is used to compensate the noisy measurements of the accelerometer.

5.7.3 Magnetometer

A magnetometer measures the earth's magnetic field in G (gauss) and it is used to calculate the quadcopter's geographical heading. The magnetometer readings are converted into polar coordinates which behave similar to a compass. The polar coordinates measure rotation around the quadcopter's Z-axis in 360 degrees, where north is 0 degrees and south is 180 degrees. The magnetometer helps to compensate the gyroscope's drift when calculating yaw for the quadcopter. [13]

5.7.4 Barometer

A barometer measures atmospheric pressure in hPa (hectopascal). When the barometer is properly calibrated and has an accurate resolution it can measure even small changes in the atmospheric pressure when quadcopter's altitude changes. The barometer is used to maintain the quadcopter's altitude. [15]

However, changing weather conditions can change the atmospheric pressure drastically and in that case the barometer's measurement will drift from the original altitude and maintaining the altitude is not accurate. The barometer can provide accurate altitude information only short term. A rangefinder would provide much more accurate measurement of the quadcopter's altitude relative to the ground level. It can measure distance to the ground and therefore it can be used to adjust altitude when the terrain is not flat.

5.7.5 Sensor configuration

LSM9DS1 chip's sensors can be configured for different measurement ranges and refresh rates. The refresh rate is configured to its maximum of 952 Hz to provide fresh measurement data for the flight controller as often as possible. Each axis outputs 16-bit value to represent the measured value. A larger measurement range leads to decreased precision because capacity of a 16-bit variable is limited.

Each sensor is configured to use the smallest measurement range to achieve the best resolution. At least for the gyroscope the range should be more than enough, because the quadcopter would need to maneuver a 360 degree flip around one axis to get a measurement over 245 dps. High resolution is preferred when attempting to estimate quadcopter's position correctly. However, different sensor measurement ranges should be tested to verify how the configuration affects the flight performance of the quadcopter. Table 3. shows all available measurement range configurations for the sensors.

Sensor name	**Measurement range**
Accelerometer	±2, ±4, ±8 and ±16 g
Gyroscope	±245, ±500, ±2000 dps
Magnetometer	±4, ±8, ±12, ±16 gauss

Table 3: Sensors' measurement ranges [13]

LPS25HB barometer measurement range is from 260 to 1260 hPa. It can run in a high resolution or a low power consumption mode and it provides the measured values in 24-bits.

Configuring both chips and reading the measurements can be done by using either I2C or SPI interfaces. A open source library is available online and it allows easy usage of the sensors on the PmodNav IMU.

5.8 MICROCONTROLLER

The quadcopter and the joystick uses an Adafruit Feather 32u4 that has a LoRa RFM95 module preinstalled. This was the most simple way to establish a long range wireless communication between both of the devices. The microcontroller is ATMega 32u4 8-bit AVR clocked at 8 MHz. It has 3.3 V logic level, 32 KB

program memory and 2.5 KB SRAM. The chip can be programmed using a USB 2.0 interface. ATMega 32u4 has also other features required by the flight controller and the joystick [16]:

- 10x analog I/O pins for reading potentiometers or battery state
- Hardware SPI for reading and writing to sensors and LoRa module
- Hardware PWM generators to control signal for ESCs
- External interrupts and timer interrupts
- It is as light as a feather – mass of 5.5 grams adds nearly nothing to the total mass of the quadcopter

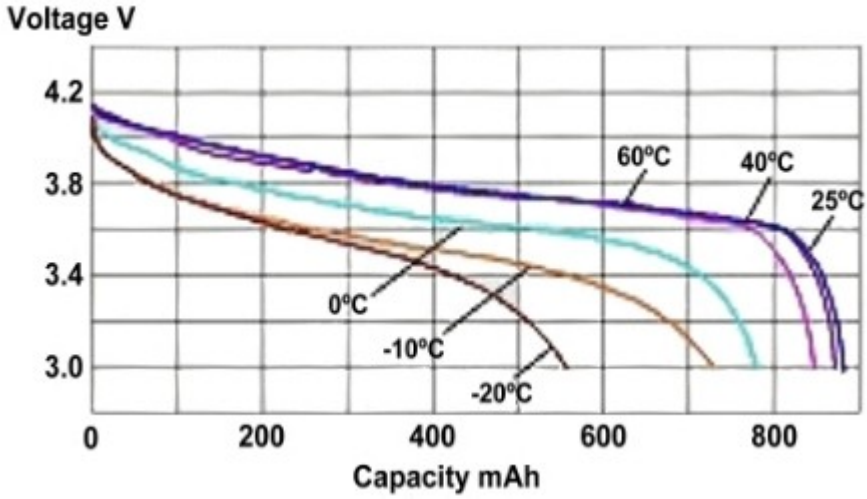
5.9 USAGE TEMPERATURES

Table 4. lists the most critical components and their usage temperature ranges.

Component	**Operating temperature range [°C]**
Gens Ace LiPo battery	-20 to +60
LSM9DS1TR	-40 to +85
LPS25HBTR	-30 to +105
ATMega 32u4	-40 to +85

Table 4: Components' usage temperature ranges

The LiPo battery can perform at more limited temperature range than the other components and its performance is affected the most by different temperatures. A cold temperature decreases the discharge rate of the battery and too warm temperature can decrease the battery's lifetime or even overheat it . The best operating temperature for a LiPo battery is room temperature. Drawing 8. shows a typical LiPo battery discharging at different temperatures.



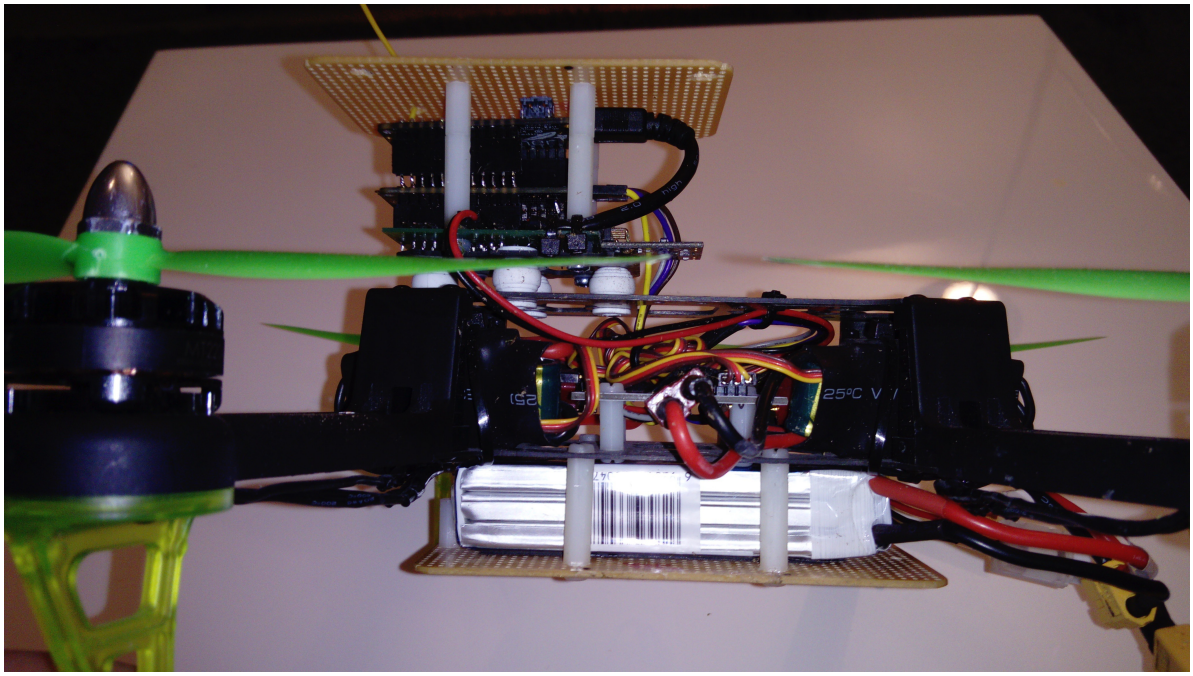
Drawing 8: 850 mAh LiPo battery discharge at different temperatures [17]

The colder the temperature is, the faster the LiPo battery's voltage output drops and all of its capacity is not discharged. At 25 celsius degrees the battery fully discharges and the voltage drop is slower than at high or low temperatures. Nonoptimal temperatures decrease the flight time and the thrust generated by the motors [17].

5.10 FRAME

All of the quadcopter's components are assembled on a frame. There are multiple sizes of frames for different types of quadcopters. Racing quadcopters have a small frame, small propellers, high motor rpm and fast maneuverability. The quadcopters that are used for photo shooting and recording videos usually have a larger frame, more mass, bigger propellers and lower motor rpm with higher torque. Generally they have slower maneuverability but can lift more mass.

Spendix Q250S racing size frame is used in the project. Its best feature is a board with rubber feet that can be used to install the microcontroller and sensors on the frame. The rubber feet help to reduce oscillation of the frame caused by the motors. The frame has also enough space for a power distribution board and electronic speed controllers, legs that support the quadcopter when it is on ground and various holes on its carbon fiber frame which allow installing different components on the frame. A battery slot and circuit board are installed on the frame with plastic spacers as demonstratedin Drawing 9.



Drawing 9: Drone's components from side view

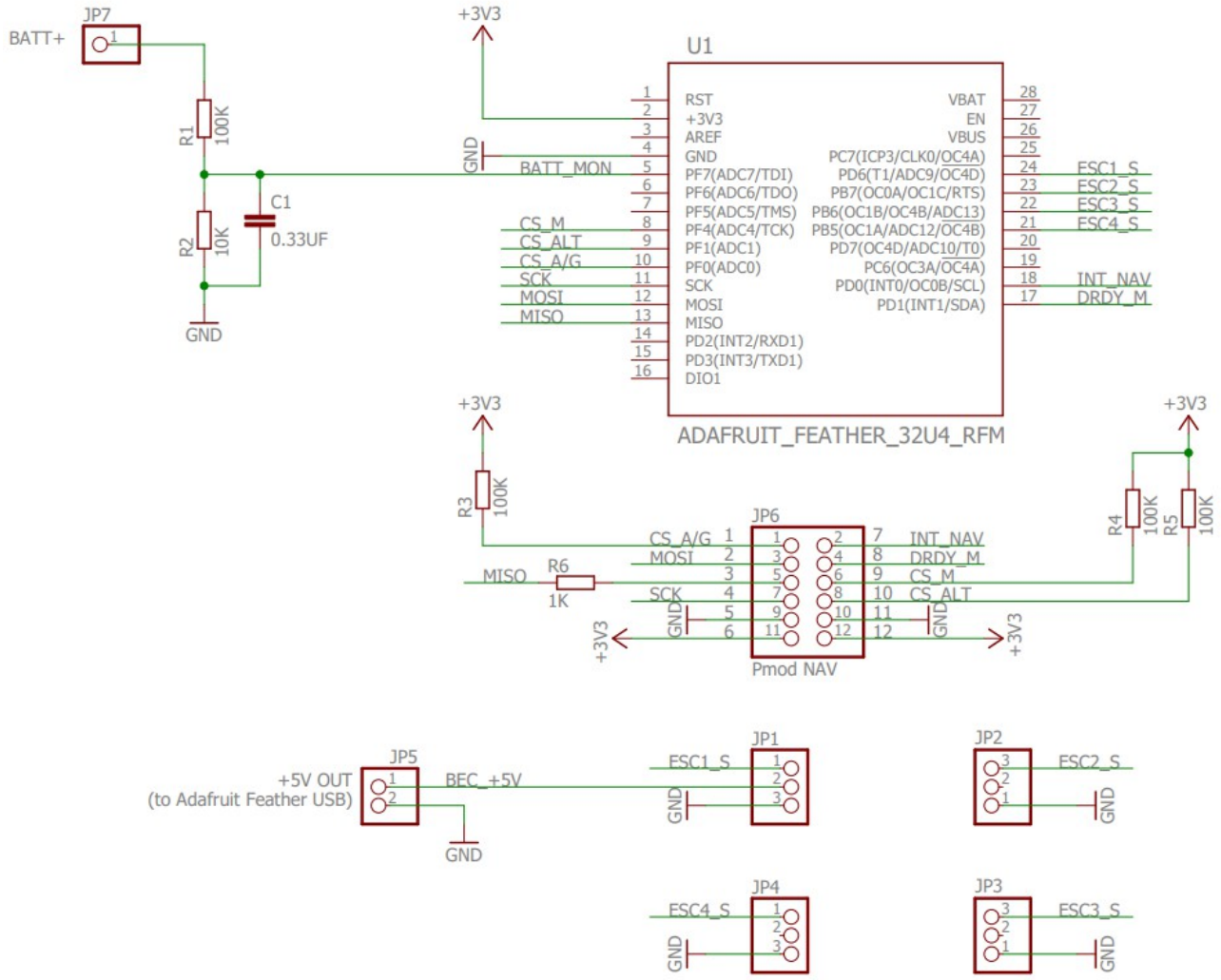
It is optimal to install components so that their mass is distributed evenly on the frame. However, it was not possible to install the board carrying the microcontoller and sensors at the center point of the frame, but fortunately their mass is relatively small compared to the mass of the battery, the electronic speed controllers and the power distribution board. A properly implemented flight controller should be able to correct its orientation regardless of slightly uneven distribution of mass.

6 FLIGHT CONTROLLER

6.1 OVERVIEW

The flight controller is responsible for calculating the current quadcopter orientation based on accelerometer, gyrometer and magnetometer measurements, receiving a target orientation from the user and calculating the control orientation using a PID controller. The control orientation will be used to calculate the speed of each motor. The flight controller can support different flight modes such as acro and self-level modes. The self-level mode is implented in the thesis. It is harder to implement but easier to fly. The self-level mode allows implementing sophisticated features, such as using a GPS to locate the quadcopter and fly to a given location. It is also the better option due to limitations of LoRa communication.

6.2 FLIGHT CONTROLLER CIRCUIT BOARD



Drawing 10: Flight controller circuit board

The microcontroller, sensors and hardware module are connected together on a custom made circuit board. Supply voltage is provided by an electronic speed controller's battery elimination circuit through JP1. JP1-4 are connected to the ESCs PWM signal inputs.

The microcontroller communicates with an Inertial Measurement Unit (IMU) using a Serial Peripheral Interface (SPI) bus. This requires wiring of MOSI, MISO, SCK and chip select signals to the microcontroller. PmodNav has four sensors and each of them needs a chip select signal. The other pins are connected to interrupt signals which indicate data ready events. The microcontroller and PmodNav are connected to the same ground and supply voltage. There are pull-up resistors connected to the chip select signals, because the chip selects are zero active and the pin can be in a floating state.

JP7 can be used to read the battery voltage level, but it is not in use because the flight controller does not implement this feature yet. JP7 has voltage division to shift the incoming signal's logic level from 11.1 V to 5 V, which can be then connected to the microcontroller's analog input.

6.3 SOFTWARE ARCHITECTURE

The flight controller and LoRa joystick are two separate projects which both use the communication and joystick components. The flight controller also uses the IMU and motor controller components to perform all the required tasks.

6.3.1 Flight controller

The flight controller is the main component for the quadcopter implementation. It contains main function, the so called flight controller loop that will execute flight controller operations in a loop during the run time. Fresh sensor measurements are read and the communication component is checked for new received packets which are then parsed into a joystick state. The joystick state and sensor readings are passed to the motor controller as input parameters.

6.3.2 LoRa joystick

LoRa joystick reads 3-axis potentiometer measurements and uses joystick class to convert the values into throttle, pitch, roll, yaw and press button commands 1 and 2. A LoRa packet is constructed from this data and it is sent to the quadcopter using the communication component. The quadcopter will parse the payload into joystick state.

6.3.3 Communication

The flight controller and the joystick use communication class to perform LoRa communication. The communication class handles receiving and sending packets as well as validating them. Validation is required to ignore random packets that do not come from the joystick.

6.3.4 Joystick

The joystick class represents joystick state and contains functions to parse 10-bit analog input into the joystick state. It also fixes small accuracy errors in potentiometer measurements. Potentiometers never reach middle state accurately, so values close to the middle state are intepreted as middle state.

6.3.5 IMU

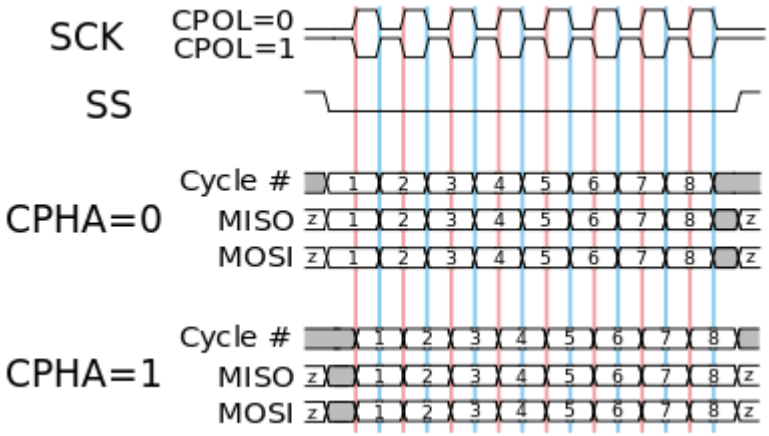
The IMU class handles configuring sensors and reading and converting sensor measurements from PmodNav. It uses SPI bus to communicate with the module.

6.3.6 Motor controller

Motor controller receives sensor readings and user commands as input. It uses a complementary filter to calculate accurate estimation of quadcopter's orientation. The complementary filter output and user input are passed to a PID controller as current orientation and target orientation. The PID controller outputs the control orientation and it is used to calculate speed for all four motors. PWM outputs are configured to generate signal for the ESCs which will then control the motor speed.

6.4 SERIAL PERIPHERAL INTERFACE

The Serial Peripheral Interface is used to communicate with PmodNav and SX1272 LoRa module. SPI is commonly used in embedded system for short distance communication. It is a synchronous protocol which means that the communication is synchronized using the master device's clock signal. A communication is initiated by the master device by setting a chip select signal low. Depending on the clock polarity, a data byte is transmitted either at low or high signal state. Once the master has transmitted its read or write command using the MOSI signal, the slave can respond similarly using the MISO signal [18]



Drawing 11: SPI protocol [19]

The LoRa module requires one chip select signal and PmodNav requires three chip select signals – one for each sensor. It is possible to communicate with one

device at a time, because the commands sent to them are different.

Signal	Master pin	Slave pin	Description
SCK	11	4	Clock signal
CS_AG	8	1	Accelerometer and gyroscope chip select
CS_ALT	9	10	Barometer chip select
CS_M	10	9	Magnetometer chip select
MOSI	12	2	Master Out, Slave In
MISO	13	3	Master In, Slave Out

Table 5: All of the IMU's SPI signals utilized by the flight controller

6.5 CALCULATING THE CONTROL ORIENTATION

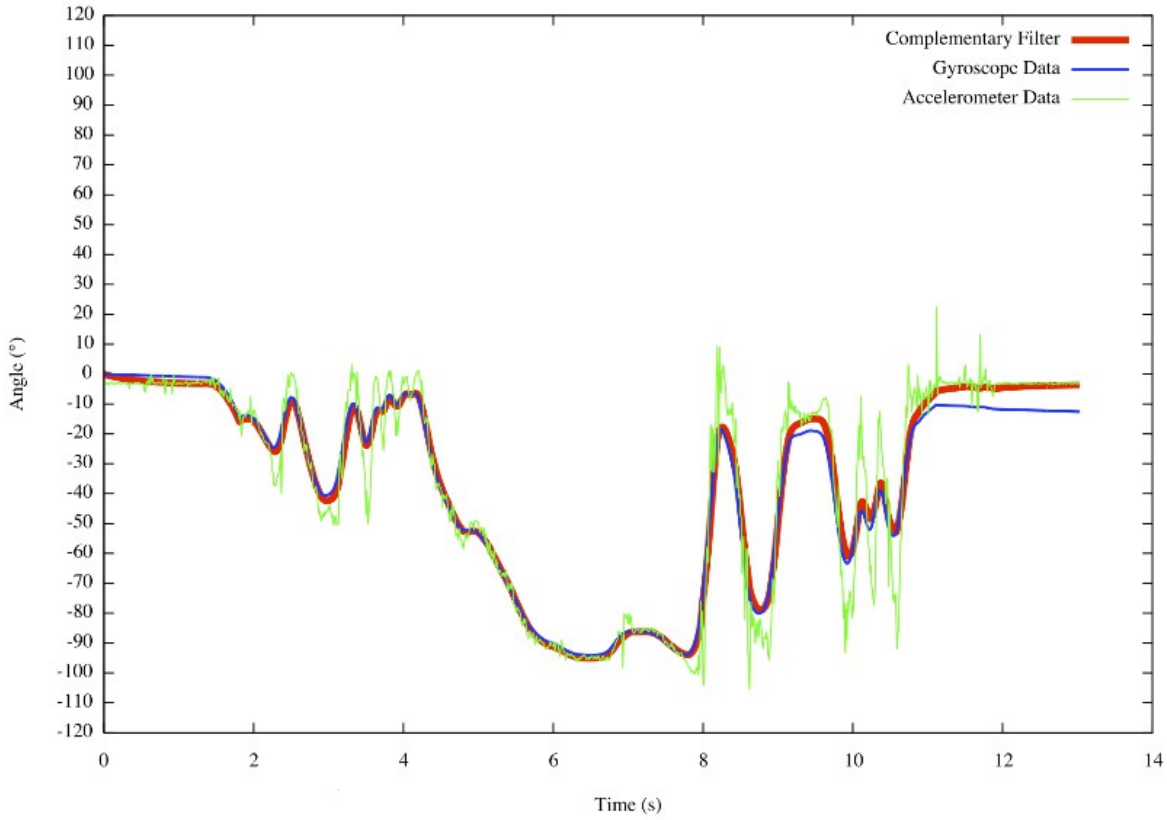
Implementing the self-leveling flight mode requires an accurate estimation of the drone's current orientation. The complementary filter is used to calculate the current orientation. It compensates the gyroscope's drift and the noise measured by the accelerometer.

6.5.1 Complementary filter

The complementary filter is easy to understand and it is computationally effective. It uses a parameter that defines the ratio between the gyroscope's and accelerometer's measurements. The following equation is used to calculate the current pitch, roll and yaw.

```
A_RATIO = 0.02
angle = (1 - A_RATIO) * angleGyroscope + A_RATIO * angleAccelerometer
```

The paramater angleGyroscope is the pitch, roll or yaw which are the results of integrating the angular velocity measured by the gyroscope. The parameter angleAccelerometer is the pitch or roll which were calculated from the accelerometer's measurements. A_RATIO defines a multiplier that determines the ratio of the angleGyroscope and angleAccelerometer. In the above example 2 % of the angleAccelerometer is used and 98 % of the angleGyroscope is used.



Drawing 12: Complementary filter [20]

The Drawing 12. shows the angle calculated using a gyroscope and an accelerometer. The accelerometer measurement has a lot of noise when the drone performs quick maneuvers. The gyroscope does not measure noise, but it starts to drift and does not return back to the neutral angle. The angle calculated using complementary filter eliminates noise and drift and it gives an accurate estimation of the drone's orientation.

6.5.2 PID-controller

In addition to calculating the current orientation of the quadcopter, it is required to control the speed of the motors according to the current orientation and the target orientation given by the user. PID controller is used to stabilize the control orientation of the drone. The output orientation tries to follow the target orientation in a smooth manner by eliminating oscillation and drifting.

Separate PID controller channels are used for the throttle, pitch, roll and the yaw. The target angle and current angle are passed as parameters into the PID controller which uses three configurable terms to calculate the control angle.

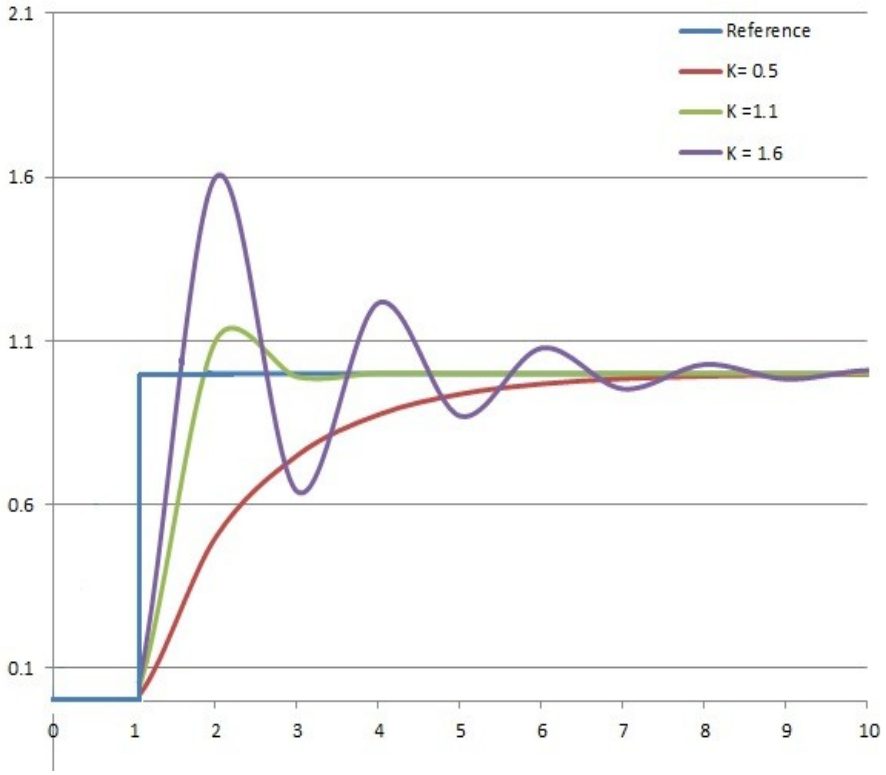
$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_0^t e(t')\,dt' + K_\mathrm{d} \frac{de(t)}{dt},$$

Kp, Ki and Kd are configurable parameters and they control how much the terms

porportional (P), integral (I) and derivative (D) affect the output u. If a multiplier is set to zero, the term won't affect the output. The parameter e(t) is the measured error margin between the current angle and the target angle at time t. [21]

```
e(t) = (targetAngle - currentAngle)
```

Porpotional term controls the error margin directly by applying the multiplier Kp. The following Figure 13. shows how changing the multiplier Kp affects the output of the PID controller. Terms I and D are constant.



Drawing 13: PID controller configured with different porpotional term values [22]

If Kp equals 1 and terms I and D are disabled, the output value will be exactly the same as the target value. A Kp multiplier lower than 1 outputs a control value that slowly reaches the target value. A Kp multiplier higher than 1 outputs a control value that quickly reaches the target value but tends to oscillate. Tuning the PID controller is usually started by configuring the P term. It should be increased until the drone can be controlled and it does not oscillate too much.

The integral term controls the cumulative error. The longer the drone is in a wrong orientation, the larger the intergral terms grows and tries to fix the orientation more agressively. Integral term helps to prevent drifting in a windy environment.

The derivative term will dampen the aggressive corrections made by the porpotional term. It estimates the trend of the error margin by comparing the current and the previous orientation. The larger the error margin is, the more the output will be dampened by the D term. [23]

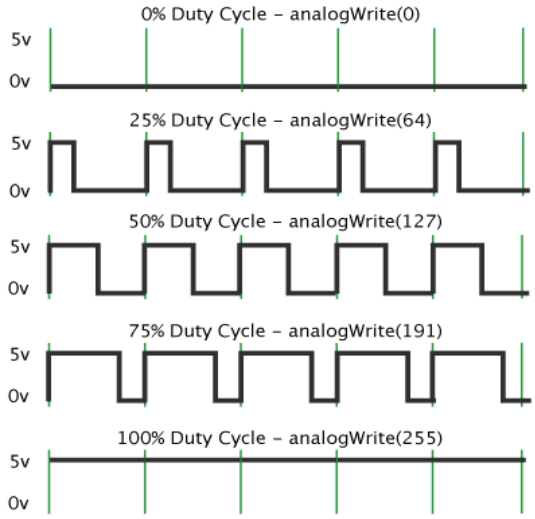
6.6 CONTROLLING QUADCOPTER ORIENTATION

6.6.1 Pulse width modulation

Pulse Width Modulation (PWM) is a common method to control servo motors and ESCs. Many ESCs implement a custom firmware and they support mostly two types of PWM signals.

The first one is the usual servo motor PWM signal at 50-500 Hz frequency where period is 2-20 ms. Usually the control signal would be from 1 to 2 ms, but BLHeli ESC uses a control signal from 1.2 to 1.8 ms. For this reason frequency cannot be higher that 500 Hz, because the period would be shorter than 2 ms. When the throttle is 0 %, the signal is set high for 1.2 ms and for 100 % throttle signal is set to 1.8 ms. If the control signal is high longer than 1.8 ms the motor will shutdown. This provides safety in situations where the signal would be erroneous due to problems in the program or hardware. A digital port in error state would not be able to generate this specific control signal, but rather be low or high all the time.

BLHeli can also be used with a 1 kHz or a higher frequency [24]. In this case the duty cycle represents directly the motor speed. The control signal is set high from 0 to 100 % of the period. For example when setting duty cycle time to 25 % of the period, the motor speed would be 25 %. If the frequency is 1 kHz, it is possible to change the motor speed every millisecond. That is 50 times faster than with 50 Hz signal. However, this is prone to error states where the digital port would be high all the time and command the motor speed to 100 %. To eliminate this problem custom hardware needs to be implemented.



Drawing 14: PWM signal at different duty cycles [25]

A 1 kHz PWM signal was chosen for the thesis because of the faster motor speed control and more important is the reason that hardware PWM modules were not able to generate 50 Hz signal with 1 to 2 ms control signal. This would have been possible with software timer interrupts but it would have been computationally more expensive.

6.6.2 Calculating motor rpm

The PID controller outputs the control throttle, pitch, roll and yaw for the quadcopter. These parameters will be used in an algorithm that calculates the actual PWM duty cycle for the ESCs. Each motor is configured to handle the pitch, roll and the yaw differently. For example, when the quadcopter is maneuvered to fly forwards, the back motors will be given more throttle than the front motors. When maneuvering to left, right front and back motor will have higher RPM than the left motors. To yaw right, right top and left back motors are given more throttle [26].

```
CW front motor = targetThrottle - controlPitch - controlRoll - Yaw
CCW front motor = targetThrottle - controlPitch + controlRoll + Yaw
CCW back motor = targetThrottle + controlPitch + controlRoll + Yaw
CW back motor = targetThrottle + controlPitch - controlRoll - yaw
```

The resulting motor speeds need to be clamped, because summing and subtracting the values can give results which overflow the maximum or minimum value. If one of the values overflow the maximum value 255, the amount of overflow is subtracted from all of the motor speeds. Values lower than zero are set to zero.

6.7 FLIGHT CONTROLLER COMMANDS

The flight controller supports active and inactive states. When the flight controller starts up, it is inactivated by default and the user can configure various parameters of the flight controller. To activate the flight the throttle must be set to 0. This allows to increase the throttle slowly to begin the flight. The flight is inactivated by pressing the right joystick for one second and letting the throttle joystick go from the bottom to the center.

6.7.1 Flight controller configuration

When the flight mode is inactive the quadcopter's PID values can be configured with the right joystick. Pulling the joystick in the right position will cycle through the configurable parameters. Pulling the joystick up and down will increment and decrement the selected parameter's value. It is also possible to reset the parameters to their default values. Default values are defined as macros in the source code. The quadcopter blinks a red led when a command is executed. Commands are executed when once the right stick is brought back to the center position.

Table 6. defines all joystick commands and their functions. The user inputs the commands using the right joystick.

Command	Function and signal
Pull the joystick right	Cycle between PID controllers terms, user input dampening and gyroscope sensitivity. Led blinks n times for each parameter: 1: Porpotional (P) 2. Integral (I) 3: Derivative (D) 4: user input dampening 5: gyroscope sensitivity
Pull the joystick up	Increment value of the selected parameter. Led blinks twice.
Pull the joystick down	Decrement value of the the selected parameter. Led blinks once.
Hold the joystick at left position for 2 seconds	Reset all settings to default. Led blinks for 2 seconds.
Press the joystick	Inactivate flight mode

Table 6: Joystick commands

6.8 FLIGHT CONTROLLER SAFETY

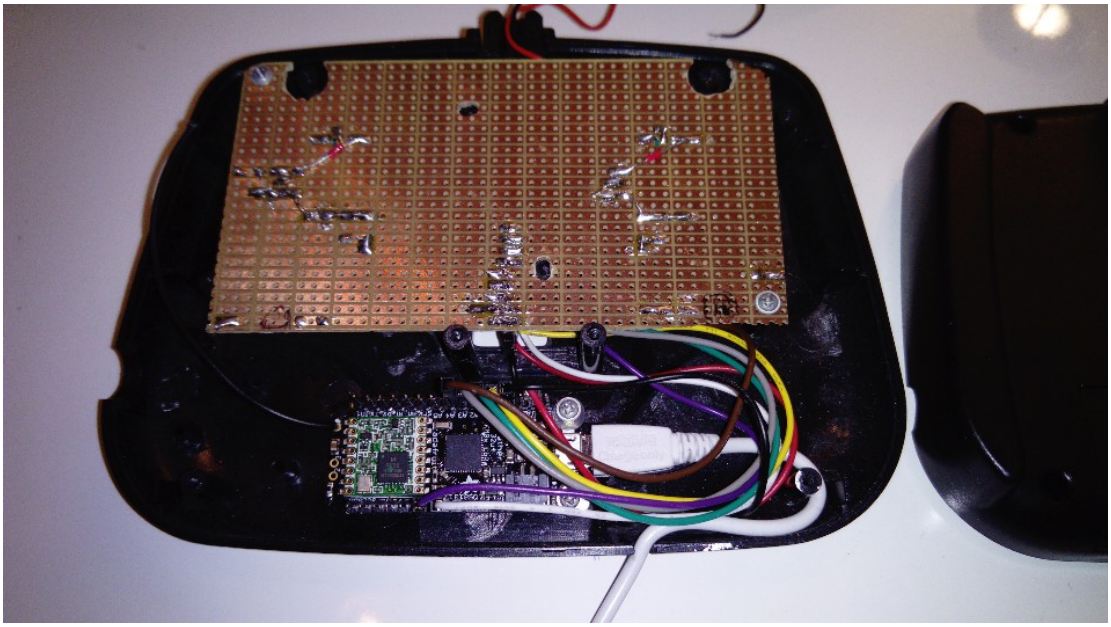
The flight controller's flight mode can be activated intuitively and safely, because it allows increasing the throttle slowly. If something goes wrong, the user can always inactivate the flight by pressing the right joystick.

If the communication between the quadcopter and joystick fails for some reason and no packets are received the quadcopter will inactivate the flight mode and shutdown the motors after 200 ms. The user can activate the flight by setting the throttle to 0. If communication is restored before the delay, the quadcopter will give the control back to joystick.

This functionality could be improved so that the drone stabilizes itself and decrease the throttle to land safely when the communication fails. After the landing it should inactivate the flight and shut down the motors so that the user can unplug the battery.

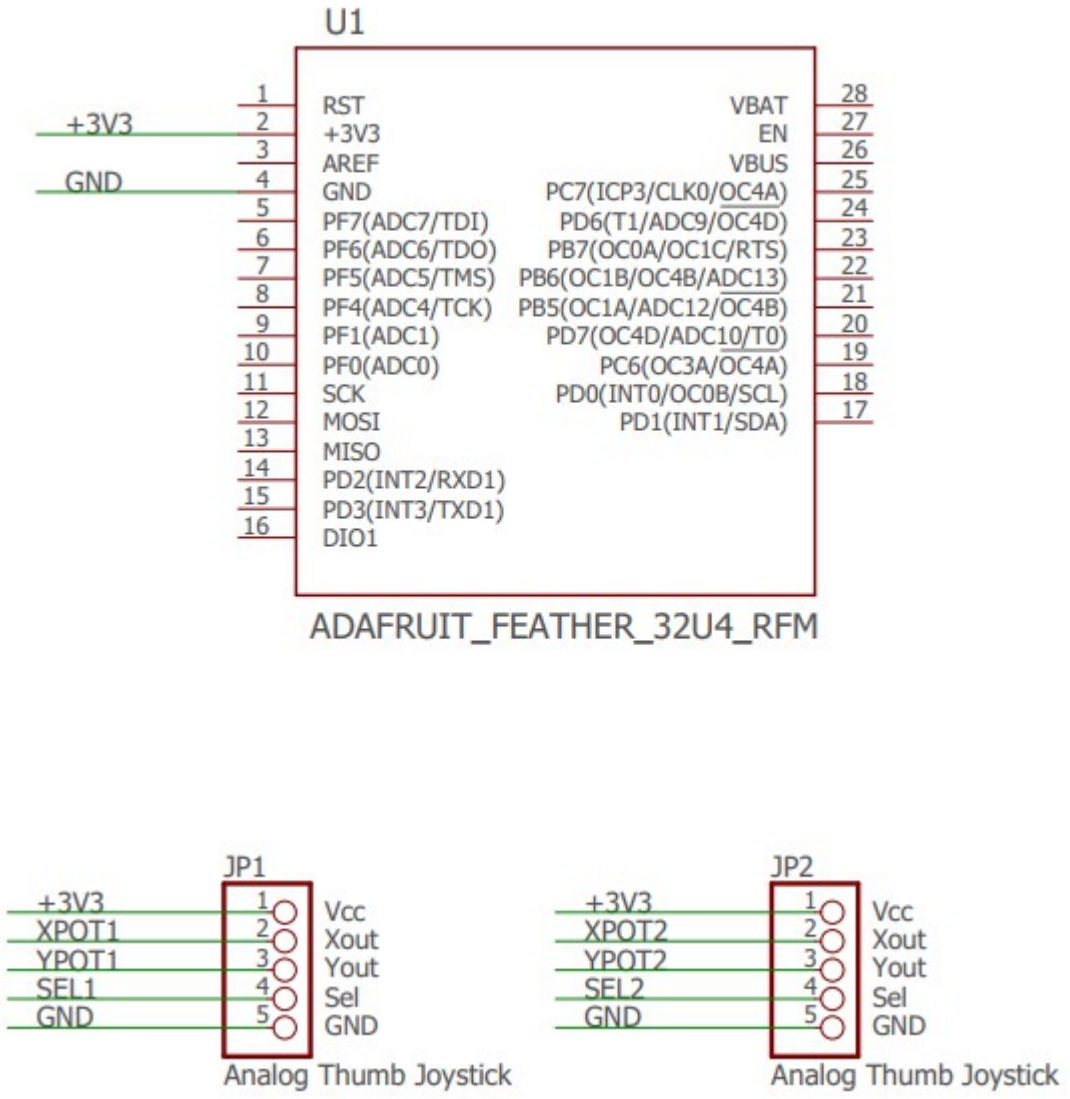
7 JOYSTICK

The joystick is built inside an old RC car controller case. It uses an Adafruit 32u4 with an interagted LoRa RFM95 module to read joystick positions and to send the commands to the drone. A 2200 mAh mobile phone power bank is attached to the backside of the controller case and it is used as a power supply as shown in Drawing 15.



Drawing 15: Joystick viewed from inside

7.1 JOYSTICK CIRCUIT BOARD



Drawing 16: Joystick circuit board

3-axis potentiometers are connected to the analog inputs of the microcontroller. A potentiometer changes its resistance linearly when the joystick moves on the X or Y-axis. This causes the voltage level to change between 0 and 5 V and the value can be read with an analog input and intepreted as joystick position. The Z-axis is read using a digital input because its value can be only low or high. The Z-axis behaves as a press button.

7.2 JOYSTICK FUNCTIONALITY

The joystick's functions are executed in a loop. On each loop the joystick reads the potentiometer measurements and parses them into a joystick state. The throttle, pitch, roll and yaw are handled as bytes and therefore their value range is from 0 to 255. When the joystick is in the center position its X and Y-axis values are 127. The joystick state is sent as a six byte payload using the communication

component.

The joystick loop is lightweight and currently the LoRa communication sets a limit for the packet send ratio. Ten packets per second is the average rate of packets sent by the joystick.

7.2.1 Interpreting the joystick state

The quadcopter will interpret the target throttle differently than the target pitch, roll and yaw. The throttle is 0 % when the joystick is at the bottom position and 100 % when it is at the top position. Therefore it will use the analog input reading directly. The following equations are used to calculate the target pitch, roll and yaw.

```
angleLimit = 25
scale =  angleLimit * 2 / 255
dampenInput = 0.8

targetAngle = dampenInput * (joystickValue * scale – angle_limit)
```

The parameter angleLimit controls the maximum target angle for pitch, roll and yaw. When the angle is limited to 25 degrees, the range is between -25 and 25 degrees. The joystick value can be either the pitch, roll or the yaw and it is scaled to the given range. The angle limit is subtracted from the result so that center point is 0 and the range is set between the angle limits. The dampen input parameter can be configured during runtime and it is used to scale the value inside the given range.

8 COMMUNICATION

8.1 LORA PROTOCOL

LoRa is a wireless radio communication technology that is used in many IoT (Internet of Things) implementations. It provides long range communication and low power consumption. LoRa was chosen as the communication technology, because we had a course at VAMK, University of Applied Sciences where I was introduced to LoRa. The Adafruit 32u4 has an integrated LoRa module that is fast to setup and the module provides easy to use libraries to handle the communication. LoRa supports also two-way communication so that the drone can send battery status or other information to the joystick. The RMF95 LoRa module uses 868 MHz frequency, which is an allowed hobbyist frequency in Europe.

The LoRaWAN supports a network architecture that has class A, B and C devices, packet forwarding and other features. The raw LoRa protocol is used in the thesis because there is no need for the network architecture. It is only required to send packets between the joystick and the quadcopter. [27]

8.2 JOYSTICK AND QUADCOPTER COMMUNICATION

RadioHead library [28] is used to communicate with the LoRa module. It handles initialization of the registers, switching between LoRa modes, receiving interrupts when a packet is received and parsing the LoRa packet to provide the payload. The microcontroller's interrupt pin 6 is connected to the LoRa module interrupt service. It is used to handle the transmit and receive interrupts. On each flight controller loop a flag status is checked. When RadioHead has received and parsed a LoRa packet, it sets the flag to true to signal that the payload can be read. The received payload is copied to a buffer of 7 bytes. The packet is validated and then parsed into a joystick state than can be used to set the target orientation for the drone.

8.3 PACKET VALIDITY AND SECURITY

Since the quadcopter might receive any LoRa packet, not only the packets sent from the joystick, the packet needs to be validated to make sure that it comes from the joystick, contains proper data and is not corrupted during transmission. Packets are sent continuously as fast as possible and on average the quadcopter receives ten packets per second. LoRa packets are often encrypted with AES-128 [29] or other strong encryption algorithms but to maximize the flight controller loop performance only computationally cheap validation algorithms are used in the thesis. The packet contains six bytes of joystick data (pitch, roll, yaw, throttle and two press buttons) and the last byte is reserved for CRC (Cyclic Redundancy Check).

CRC is calculated of the 6 byte payload by summing all the bytes. CRC is set as the 7th byte of the payload. An exclusive or (XOR) operation is run on each byte of the payload using a three byte XOR key. CRC and XOR provide data integrity and validation so that invalid packets from unknown sources can be ignored. It also makes interpretation of the packet slighty harder for a device that the packet does not belong to.

8.4 PERFORMANCE

The commercial drones usually communicate using a multichannel analog signal and the protocol varies between the implementations [30]. At least four channels are required for the throttle, pitch, roll and the yaw. The multichannel communication protocols are faster than the packet based LoRa technology. The delay between LoRa packets is roughly 100 ms. This is acceptable when the drone implements a self-leveling flight mode.

The more interesting aspect of LoRa technology is that it can be used to control multiple self-sustaining drones. The drone's could navigate to a GPS location that is sent by the control system and the drone's could send different kind of measurement data to the control system. LoRa is a flexible technology but it is not suitable for use cases where high throughput or low latency is required.

9 DRONE FLYING AND THE FINNISH LAW

Drone is an unmanned flying vehicle that is meant for hobby and sport use. There are various details in Finnish law that concerns flying a drone. The aviation law is available at Finlex [31] and Trafi's Q&A page [32] answers the most important questions for a private person.

- Flying a drone is not age restricted.
- The person controlling the drone needs to have constant line of sight to the drone.
- A drone can be flied in FPV (first person view) if there is another person that can observe the airspace and has constant line of sight to the drone.
- A drone cannot be flied at altitudes higher than 150 meters.
- Weather and lightness needs to be good enought to spot other flying vehicles and evade them with using bare eye contact without using any equipment
- A drone can be flied indoors if the owner of the building accepts it.
- A drone needs to evade all manned flying vehicles.
- It is against law to fly a drone in locations, such as an airport, where it will disturb manned. flying vehicles or otherwise cause any dangerous situation.
- An autopilot can be used, if you can at any time take the control of the drone and evade people and other flying vehicles.
- Flying a drone on your own property can still violate someone's privacy. In this case they can contanct Trafi to reclamate.
- A drone does not need to be registered.

10 CURRENT STATE OF THE PROJECT

10.1 CURRENT IMPLEMENTATION

The drone's hardware is fully functional. All of the components are compatible with each other, a custom made circuit board is used to connect the microcontroller and sensors to the power supply and the communication between the joystick and the drone is stable and reasonably fast. The sensors are installed on rubber feet that reduce the oscillation of the sensors.

The joystick is able to read 3-axis potentiometer values and communicate to the quadcopter. The quadcopter receives 10 packets per second. The packets are parsed into a joystick state and it is used to set the target orientation of the drone. The flight controller uses the accelerometer and the gyroscope measurements to calculate the current orientation of the drone. The PID controller's output is used to calculate the speed of the motors.

10.2 REQUIRED IMPROVEMENTS

Most of the flight controller's features are working, but the flight controller is not able to stabilize the quadcopter properly.

10.2.1 Flight controller loop frequency

Many commercial flight controllers have a flight controller loop running at 1 kHz frequency. It can be even as fast as 8, 16 or 32 kHz. The frequency of flight controller loop implemented in thesis has been around 180-400 Hz during different development stages. A small racing size drone requires a fast executing flight controller loop, because they have less mass, higher motor rpm and fast maneuvers. This seems to be one of the limitations in the current implementation, because during the flight it can be seen that the flight controller is trying to fix the orientation but does not do it fast enough or does fix the orientation too much.

Adafruit 32u4 has programmable clock dividers that can be changed to achieve faster clock signal. By default the clock frequency is 8 MHz, but it should be possible to increase the frequency to 16 MHz.

There should be various optimisations that improve the execution time of the flight controller loop. For example, an external hardware PWM module should not be used, because SPI communication to it takes few milliseconds each loop and that has a huge effect on performance. This requires changes in the circuit board and PWM signal generation.

A minor improvement would be to use quaternions instead of euler angles to calucate the drone's orientation, because quaternions are slightly faster to compute. However, this alone would not increase the performance a lot. There are

likely other algorithms which should also be optimised, such as motor speed calculation that uses unnecessary conversion of values.

10.2.2 Yaw calculation

Currentyle the flight controller does not use a magnetometer to compensate gyroscope's drift when calculating the yaw. This is due to configuration and calibration problems with the magnetometer and therefore it does not provide reliable measurements. The drone tends to yaw to the right and this affects the flight when gyroscope drift grows.

10.2.3 PID controller tuning

After the above major errors have been fixed, the drone requires PID controller tuning to stabilize the flight correctly. Currently PID controller tuning affects the flight performance of the drone, but other problems in the flight controller prevent tuning and testing the PID controller properly.

11 CONCLUSIONS

The goal of the thesis was to design, build and program a quadcopter and a joystick and to find the possibilities and limitations that LoRa technology introduces to controlling a drone. LoRa is commonly used in Internet of Things applications, such as providing sensor measurement data of industrial facility. These use cases require long range and low power consumption but are not as demanding as a drone when it comes to the speed of the data transmission.

The LoRa communication protocol is packet based and has lower data transmission speed compared to the analog multichannel protocols used by the regular drone joysticks. However, it should be fast enough to control a self-leveling drone and it also introduces possibilities to control a self-sustaining drone. The drone could navigate independently to the GPS locations sent by the control system and also communicate back to the control system, providing for example measurement and status data. The LoRa technology did not set any limitations to the project and the problems reside in the flight controller that has limited performance due to frequent communication to the other modules and errors in the yaw calculation.

During the project I learned a lot of detailed information about drones, embedded systems programming and hardware design. Implementing a flight controller is a challening task because there are various things on the hardware and software level that need to work together in a fast changing system and a demanding environment. I will continue my drone project as a free time hobby - improving the flight controller and implementing fancy features like FPV camera and battery monitoring.

12 REFERENCES

[1] Aircraft principal axes
https://en.wikipedia.org/wiki/Aircraft_principal_axes

[2] Arco and self-level modes
https://oscarliang.com/rate-acro-horizon-flight-mode-level/

[3] Quadcopter flight configurations
http://ardupilot.org/copter/docs/connect-escs-and-motors.html

[4] Brushless DC motor
http://www.thinkrc.com/faq/brushless-motors.php

[5] Brushless DC motor fundamentals
https://www.monolithicpower.com/pub/media/document/Brushless_DC_Motor_Fundamentals.pdf

[6] 3-phase brushless DC motor steps
https://www.digikey.com/en/articles/techzone/2013/mar/an-introduction-to-brushless-dc-motor-control

[7] LiPo battery
https://www.dronetrest.com/t/lipo-batteries-how-to-choose-the-best-battery-for-your-drone/1277

[8] Two bladed propeller
https://www.slideshare.net/tomcpaulus/intro-to-multicopters

[9] Emax MT2204 description
https://www.emaxmodel.com/emax-multicopter-motor-mt2204-kv2300.html

[10] PmodNav references
https://reference.digilentinc.com/reference/pmod/pmodnav/start

[11] Accelerometer information
https://www.phidgets.com/docs/Accelerometer_Primer

[12] Accelerometer image
https://www.iotone.com/term/accelerometer/t20

[13] LSM9DS1 datasheet
http://www.st.com/resource/en/datasheet/lsm9ds1.pdf

[14] Complementary filter, gyroscope and accelerometer
http://www.pieter-jan.com/node/11

[15] Barometer datasheet
http://www.st.com/web/en/resource/technical/document/datasheet/DM00141379.pdf

[16] Adafruit 32u4 product description
https://www.adafruit.com/product/3078

[17] LiPo battery discharge
http://batteryuniversity.com/learn/article/discharging_at_high_and_low_temperatures

[18] SPI protocol
https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi

[19] SPI protocol image
https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

[20] Complementary filter
http://www.pieter-jan.com/node/11

[21] PID controller
https://www.csimn.com/CSI_pages/PIDforDummies.html

[22] PID controller (image)
https://en.wikipedia.org/wiki/PID_controller

[23] PID tuning
https://oscarliang.com/quadcopter-pid-explained-tuning/

[24] PWM BLHeli frequencies
https://github.com/bitdump/BLHeli/issues/55

[25] PWM signal
https://www.arduino.cc/en/Tutorial/PWM

[26] Motor direction calculation
https://github.com/simondlevy/Hackflight/blob/master/src/mixer.hpp

[27] LoRaWAN
https://www.digita.fi/yrityksille/iot/mika_on_lorawan

[28] RadioHead library
http://www.airspayce.com/mikem/arduino/RadioHead/

[29] LoRa packet encryption
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5038744/

[30] Drone radio communication
http://www.thedronefiles.net/2016/10/03/understanding-radio-communication-protocols/

[31] The aviation law
https://www.finlex.fi/fi/laki/alkup/2014/20140864

[32] Trafi, drone Q&A
https://www.trafi.fi/tietopalvelut/usein_kysyttya/ilmailu_-_miehittamattomat_ilma-alukset_ja_lennokit