การออกแบบและพัฒนาระบบควบคุมกล้องถ่ายภาพและระบบบันทึกข้อมูลของเครื่องบินไร้คนขับ
เพื่องานแผนที่ทางอากาศ

นายสันติสุข ก่องแก้ว

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมสำรวจ ภาควิชาวิศวกรรมสำรวจ
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2557

A DESIGN AND DEVELOPMENT OF CAMERA CONTROLLER AND DATA LOGGER IN UNMANNED AERIAL VEHICLE (UAV) FOR AERIAL MAPPING

Mr. Santisouk Kongkeo

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Survey Engineering
Department of Survey Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2014

Thesis Title	A DESIGN AND DEVELOPMENT OF CAMERA CONTROLLER AND DATA LOGGER IN UNMANNED AERIAL VEHICLE (UAV) FOR AERIAL MAPPING
By	Mr. Santisouk Kongkeo
Field of Study	Survey Engineering
Thesis Advisor	Assistant Professor Sanphet Chunithipaisan, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

_______________________Dean of the Faculty of Engineering

(Professor Bundhit Eua-arporn, Ph.D.)

THESIS COMMITTEE

_______________________Chairman

(Associate Professor Banjerd Phalakarn, Ph.D.)

_______________________Thesis Advisor

(Assistant Professor Sanphet Chunithipaisan, Ph.D.)

_______________________Examiner

(Colonel Kanok Weerawong, Ph.D.)

_______________________External Examiner

(Associate Professor Deeboon Methakullachat, Ph.D.)

สันติสุข ก่องแก้ว : การออกแบบและพัฒนาระบบควบคุมกล้องถ่ายภาพและระบบบันทึกข้อมูลของเครื่องบินไร้คนขับเพื่องานแผนที่ทางอากาศ (A DESIGN AND DEVELOPMENT OF CAMERA CONTROLLER AND DATA LOGGER IN UNMANNED AERIAL VEHICLE (UAV) FOR AERIAL MAPPING) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร.สรรเพชญ ชื้อนิธิไพศาล, 98 หน้า.

โดยทั่วไป แผนที่ภาพถ่ายทางอากาศต้องการภาพถ่ายที่มีคุณภาพสูง เช่น ภาพที่แสดงรายละเอียดชัดเจน ภาพถ่ายในแนวดิ่ง ภาพถ่ายต้องมีพื้นที่รายละเอียดที่ซ้อนทับ นอกจากนี้รวมถึงค่าพารามิเตอร์ของ Interior orientation, exterior orientation และ ค่าพิกัดของหมุดควบคุม ทั้งหมดนี้มีความจำเป็นสำหรับการประมวลภาพถ่ายเพื่อทำแผนที่ออโธ ดังนั้นการควบคุมการถ่ายภาพและระบบบันทึกข้อมูลจึงมีความสำคัญและเป็นประโยชน์ อย่างไรก็ตาม ระบบดังกล่าวส่วนใหญ่จะไม่มีในระบบของอากาศยานไร้คนขับ

งานวิจัยนี้นำเสนอการออกแบบและจัดทำตัวควบคุมกล้องถ่ายภาพแบบ 2 แกน และการพัฒนาระบบบันทึกข้อมูลสำหรับ UAV แบบปีกหมุน เครื่องมือที่พัฒนาขึ้นนี้ใช้เพื่อควบคุมกล้องถ่ายภาพให้รักษาการถ่ายภาพให้อยู่ในแนวดิ่ง พร้อมกับบันทึกค่าพารามิเตอร์ exterior orientation เพื่อใช้ในการประมวลผลภาพถ่ายและให้ได้ค่าพิกัดขณะถ่ายภาพของแต่ละภาพ ต้นแบบระบบความคุมกล้องถ่ายภาพแบบ 2 แกน สามารถหมุนรอบแกน x (roll) และแกน y (pitch) ในช่วงมุม 45 องศา โดยใช้มอเตอร์แบบ Brushless เพื่อช่วยให้การหมุนเป็นไปอย่างต่อเนื่องราบลื่น ผลการทดลองแสดงให้เห็นว่าภาพถ่ายที่ประมวลผลแล้วมีคุณภาพสูงขึ้นจากการควบคุมกล้อง 2 แกนและจากการบันทึกค่าพารามิเตอร์ขณะถ่ายภาพ ภาพถ่ายจะมีการเอียงของภาพอยู่ในช่วง -1.889 ถึง 2.099 องศา และ -1.247 ถึง 1.625 องศาของค่า roll กับ pitch ตามลำดับ นอกจากนี้ค่าพารามิเตอร์ exterior orientation ที่ได้นั้นช่วยให้การประมวลผลภาพถ่ายทางอากาศง่ายและให้ผลดีขึ้น

ภาควิชา วิศวกรรมสำรวจ ลายมือชื่อนิสิต ....

สาขาวิชา วิศวกรรมสำรวจ ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

ปีการศึกษา 2557

5670486821 : MAJOR SURVEY ENGINEERING
KEYWORDS: GIMBAL / EXTERIOR ORIENTATION PARAMETERS / DATA LOGGER / UAV / MAPPING

SANTISOUK KONGKEO: A DESIGN AND DEVELOPMENT OF CAMERA CONTROLLER AND DATA LOGGER IN UNMANNED AERIAL VEHICLE (UAV) FOR AERIAL MAPPING. ADVISOR: ASST. PROF. SANPHET CHUNITHIPAISAN, Ph.D., 98 pp.

Generally, an aerial mapping requires image with good quality such as clear image, nearly vertical image, including more overlapping image, as well as constant interior orientation, exterior orientation parameters and ground control point (GCP), all of which are necessary for image processing to generate the orthophoto map. Therefore the control of imaging and data logging system are important and very useful. However, these kind of systems are generally not supported in UAV platform.

This research will present the designing and creating of 2-axis brushless gimbal (camera controller), including the development of data logger for using in unmanned aerial vehicle (UAV) platform, particularly multi-rotor UAV (or Drone). Which these developed tools will be used in order to control and stabilize a camera for vertical imaging to get a nearly vertical image, as well as data logging of exterior orientation parameters for digital image processing and georeferencing in aerial mapping. The prototype of 2-axis brushless gimbal is capable of rotating around X-axis (roll) and Y-axis (pitch) in a range of maximum at -45° to 45° by using Brushless DC (BLDC) motor, which helps the camera stabilize smoothly. The results of the experiment shows better quality of images that are derived from developed 2-axis brushless gimbal and data logger. Images were tilted in range of -1.889° to 2.099° and -1.247° to 1.625° of roll and pitch angles respectively. In addition, all exterior orientation parameters of each image can also be recorded to facilitate aerial mapping processing.

Department:	Survey Engineering	Student's Signature	....
Field of Study:	Survey Engineering	Advisor's Signature	....
Academic Year:	2014		

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere thanks to my thesis advisor, Asst. Prof. Dr. Sanphet Chunithipaisan for his invaluable help and constant encouragement throughout the course of this research. I am most grateful for his teaching and advice. I would not have achieved this far and this thesis would not have been completed without all the support that I have always received from him.

Besides my advisor, I am also grateful for the teachers of survey engineering department: Assoc. Prof. Dr. Banjerd Phalakarn, Col. Dr. Kanok Weerawong, Assoc. Prof. Dr. Deeboon Methakullachat, and others person for comments, suggestions and all their help to make this thesis more completely.

Finally, I most gratefully acknowledge my parents and my friends for all their support throughout the period of this research.

CONTENTS

Page

Page

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

Page

CHAPTER 1
INTRODUCTION

1.1 Statement of the problem

In the recent years, in the field of photogrammetry and the remote sensing, an unmanned aerial vehicle (UAV) platform has been used extensively and popularly for collecting the data, which were mostly used in civil engineering applications such as an autonomous surveillance, emergency and disaster management, traffic surveillance and management, photogrammetry for 3D modeling, remote-sensing based inspection systems (Siebert and Teizer 2014). It is also used to measure of cell phone, radio, television signal coverage over any terrain and so on. It has many advantages such as low-cost, no needed a pilot, easy to transport from one place to another as well as it can fly in high risk situations, inaccessible area, adverse weather conditions, through poisonous gas clouds and over regions of high radioactivity at low altitude.

The UAV system has been developed and expanded increasingly in the fields of aerial surveying. However, the low-cost UAV system it has still limited of the sensor payload in weight and dimension (Henri 2009). Hence, in order to select the appropriate equipment for small UAV should be considered due to some equipment that can find in the market but cannot use directly. Especially, the camera mounts or it's called "Gimbal" due to limitation as a mentioned previously that includes the purpose of different usage. Because in photography applications, it was applied as mounting for the camera to control and stabilize gimbaled camera in order to use for taking photo, video, and filming on the terrestrial or aerial photography. Therefore, it is necessary to develop this equipment for a mission-oriented UAV (Lee, Kaminer et al. 2010) like an tracking of the object or target in the real time for surveillance applications.

In general, the fundamental of photogrammetric point determination requires image coordinate of tie points, constant interior orientation, exterior orientation and ground control points (GCPs) (Michael 2007). Which in the past, the exterior orientation parameter can be determined by GPS/IND (Legat 2006), with one or many images that known position of GCP (Elnima 2013). But nowadays, there are low-cost sensors such as global positioning system (GPS), Barometer, and initial measurement unit (IMU) providing the continuous of estimated position, altitude and orientation of the UAV, which it's possible to generate the georeferencing imagery directly from UAV without the use of GCP. The aerial mapping requires the good quality image that good quality such as clear image, nearly vertical image, including more overlapping image, as well as exterior orientation parameters, which all of them are important and indispensable for automatic processing to generate the orthophoto map. Therefore, the control of imaging and data logging system are important and very useful. However, this kind of systems is generally not supported in UAV platform. Data captured from UAV is typically the flight data log and images from a camera which they are recorded independently. Because of this limitation, it is time consuming for matching them together using time stamp between image and flight data log before processing map. Furthermore, during UAV is flying it is difficult to control itself in order to stabilize a vehicle along horizontal plane without tilting. Therefore, it may cause the acquired images to be blur and not sufficient overlap for a complete processing of the data set (Henri 2009).

This research will present the design and development of camera controller and data logger in Unmanned Aerial Vehicle (UAV) for aerial mapping. Which consist of the design, and create the prototype of 2-axis brushless gimbal (camera mounts frame) including data logger in order to control the imaging in horizontal plane to get the nearly vertical images and record the exterior orientation data for supporting the automatically digital images processing and georeferencing.

1.2 Objectives

The main objectives of this research are:

1.2.1 To design and create a prototype of gimbal (camera controller) for small UAV platform. Particularly, rotary wing type in order to use together with a compact digital camera in the aerial photography.

1.2.2 To make the data logger for recording the exterior orientation data from many sensors while imagery were taken.

1.3 Scope of research

1.3.1 Application program that used in research

1) Arduino IDE (Ver. 1.0.5-r2)

2) Brushless Gimbal - BruGi (Ver. 50_r217)

3) Mission Planner (Ver. 1.3.19)

4) Autodesk AutoCAD (Ver. 2015)

5) KISSlicer (Ver. 1.1.0.14)

6) Agisoft PhotoScan (Ver. 1.1.4)

1.3.2 Equipment that used for research and development

1) Small UAV, which is rotary wing type consist by six propellers (Hexacopter).

2) Compact digital camera (Pentax Optio WG-2).

3) Brushless DC motor for gimbal (EMAX GB2210).

4) Gimbal controller board and IMU (RCTIMER 2-axis brushless gimbal controller & IMU v1.0).

5) Arduino controller board (Arduino Mega 2560).

6) Breakout board IMU (MPU-9150).

7) Breakout board for SD-MMC and SD card.

8) Infrared (IR) transmitter.

9) 3D printer (ExtraBot).

1.3.3 Data that used in research

For the data used to this research comprise of images with attributes data (file name or number of image, date, time), interior orientation parameters (focal length, principal distance, lens distortion) that derived from digital camera, and exterior orientation parameters (latitude, longitude, altitude, roll, pitch, yaw angles) that derived from GPS receiver module, Barometer, and IMU. Which all of the data were recorded into SD card and they are the data that derived from actual experiment.

In addition, in this research the ground control point (GCP) will not be used in the orthophoto mosaicing.

1.4 Expected Benefits

The expected outcomes of this research are:

1.4.1 Obtain an appropriate tool for aerial photography used with a small UAV platform. Particularly, rotary wing type.

1.4.2 Obtain the nearly vertical image and log file (exterior orientation parameters) for processing of orthophoto with georeferencing.

1.4.3 Reduce the procedure and time for collecting, preparing of the images and important data for processing.

CHAPTER 2
RELATED TECHNOLOGY AND LITERATURE REVIEWS

In this chapter the hardware, software, and principles that related with this research will be described. To meet the objectives of this research, there are several components have to be concerned. The characteristics of UAV have to be reviewed. The camera controller requires two parts including a gimbal to keep the camera stabilized and data logger to record all necessary parameters for photogrammetry processing. All parts relating to building up the camera controller will be described. The theory of photogrammetry will be briefly introduced. Finally the related research will be drawn.

2.1 UAV system

In the last few years, the members of open source community have developed flight control, which supporting many modes of flying such as stabilized, altitude hold, loiter, return-to-launch (RTL), follow me, auto (take-off, landing, way point navigation) and so on. The follow me mode allows system to follow another one in an automated flying mode. The return-to-launch mode is essential mode for fail safe condition. Furthermore, in particular the way point navigation function is a useful tool for photogrammetric applications (Henri 2009).

In addition, UAVs can be categorized by its structure and by its type of taking-off/landing. The following explains the advantages and limitations of four main UAV types. Table 2.1 summarizes the UAV categories.

Table 2.1 Comparison of advantages and limitations of existing flight systems (Siebert and Teizer 2014)

Aircraft Type	Airships	Fixed wing aircraft	Helicopters	Multicopters
Efficiency and range*	Very good	Very good	Average	Poor
Flexibility and maneuverability	Average	Poor	Very good	Very good
Weather dependency	Poor	Good	Good	Good
Payload	Very good	Good	Very good	Average
Safety	Good	Average	Poor	Average
Complexity and simplicity	Good	Average	Poor	Good
Running costs	Poor	Average	Average	Very good
Setup time	Poor	Average	Good	Very good

(*can be limited by federal or state regulations)

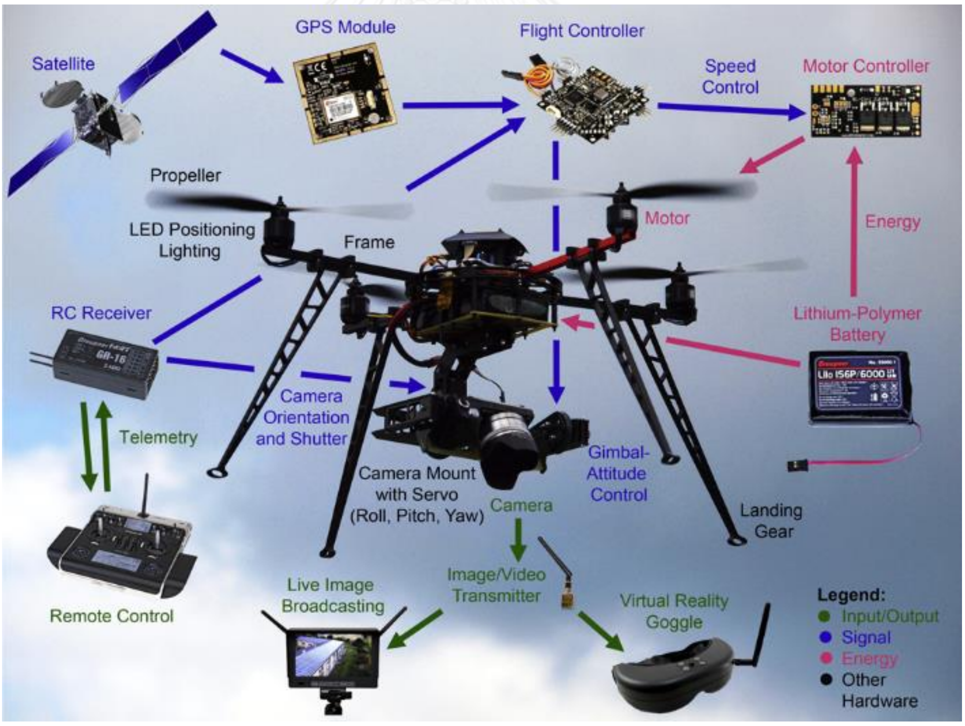


Figure 2.1 The main components of the developed quadcopter UAV system (Siebert and Teizer 2014)

Multicopter is a special type of a rotary wing aircraft. The torque does not create energy loss due to the rotors rotating in the opposite direction. The high efficiency of helicopters has yet to be met because of its reliance on electronic motors and batteries. Their main advantage is the ease of operation, high flexibility, and stability. Commonly used multicopters have at least four rotors (quadrocopter), but can have more (hexacopter or octocopter) for redundancy purposes. Purchasing cost and maintenance typically increases respectively, although an increase in safe operation and payload are sometimes important factors to consider. Multicopters can also be programmed to fly autonomously along pre-programmed waypoints. Several software supports for this purpose (Siebert and Teizer 2014).

2.2 Camera controller system

Camera mounts for small format aerial photography (SFAP) vary tremendously depending upon the type of platform, camera, and functional requirements for camera operation (James, Irene et al. 2010). The mount could be as simple as a hand-held camera shooting out the window of a small airplane, helicopter, multicopters, or as complex as a camera controller. Which the camera controller system may comprises of camera controller and many sensors such as, barometer (altimeter), GPS, IMU, and data logger for using in the specific work like a photogrammetry.

In addition, every sensors that input into this system will be added weight, power, and integration issues. Therefore, these complications should be taken into account when considering camera controller system for using on small UAV platform to apply in the aerial mapping applications.

2.2.1. Gimbal (camera controller)

Gimbal is a pivoted support that allows the rotation of an object around axis. It is extensively used in components that are required to stay stable in unsteady conditions. Basically a gimbal consists of two or three rings attached to each other axially at 90°, with the component to be protected against movement suspended between the rings.

Gimbal was used to prevent the movement of the early compasses on ships. Nowadays it has a multitude of applications such as inertial navigation, rocket engines, photography, filming and video.

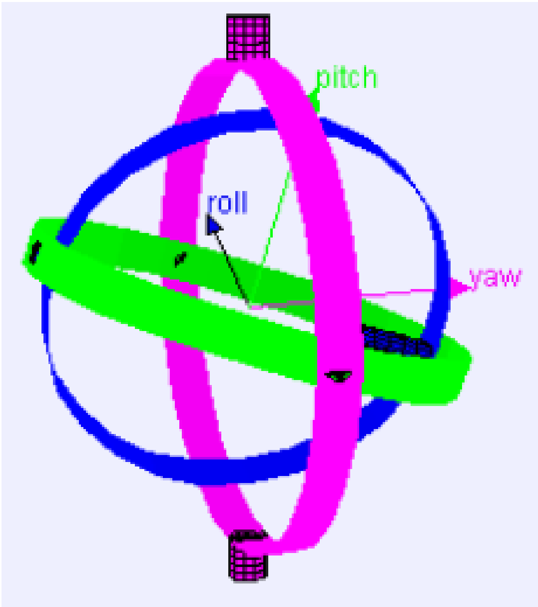


Figure 2.2 A set of three gimbals mounted together, each offers a degree of freedom: roll, pitch and yaw

(Source: http://upload.wikimedia.org/wikipedia/commons/5/5a/Gimbal_3_axes_rotation.gif)

1) Inertial navigation

In inertial navigation, as applied to ships and submarines, a minimum of three gimbals are needed to allow an inertial navigation system (stable table) to remain fixed in inertial space, compensating for changes in the ship's yaw, pitch, and roll. In this application, the Inertial Measurement Unit (IMU) is equipped with three orthogonally mounted gyros to sense rotation about all axes in three-dimensional space. The gyro outputs drive motors controlling the orientation of the three gimbals as required to maintain the orientation of the IMU.

2) Rocket engine

In spacecraft propulsion, rocket engines are generally mounted on a pair of gimbals to allow a single engine to vector thrust about both the pitch and yaw axes; or sometimes just one axis is provided per engine. To control roll, twin engines with differential pitch or yaw control signals are used to provide torque about the vehicle's roll axis.

3) Photography

Gimbals are also used to mount everything from small camera lenses to large photographic telescopes. In portable photography equipment, single-axis gimbal heads are used in order to allow a balanced movement for camera and lenses. This proves useful in wildlife photography as well as in any other case where very long and heavy telephoto lenses are adopted: a gimbal head rotates a lens around its center of gravity, thus allowing for easy and smooth manipulation while tracking moving subjects.

Very large gimbal mounts are used in satellite photography for tracking purpose. In addition, gyro stabilized gimbals which house multiple sensors are also used for airborne surveillance applications.

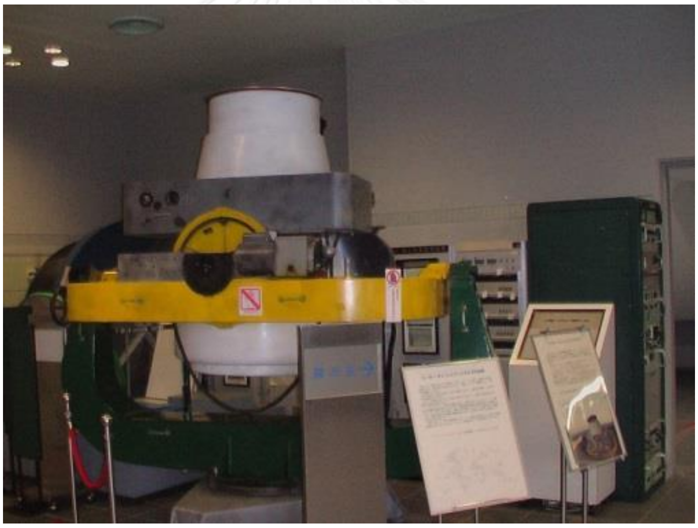


Figure 2.3 A Baker-Nunn satellite-tracking camera on an altitude-altitude-azimuth mount

(Source: http://upload.wikimedia.org/wikipedia/commons/thumb/f/f8/Baker-Nunn_camera_001.JPG/200px-Baker-Nunn_camera_001.JPG)

4) Filming and video

Handheld gimbals are used in stabilization systems designed to give the camera operator the independence of handheld shooting without camera vibration or shake. By using the brushless motors, the gimbals can keep the camera level on all axes as the camera operator moves the camera. An inertial measurement unit (IMU) responds to movement and use separate motors to stabilize the camera.

In addition, the gimbals can be mounted on cars and drones, where vibrations or other inconceivable movements would make tripods or other camera mounts unacceptable.

Figure 2.4 2-axis handheld gimbal

(Source: http://www.aliexpress.com/store/product/G-Stabilizer-2-Axis-Brushless-Camera-Gimbal-Video-Stabilizer/811281_1783496846.html)

Figure 2.5 2-axis aerial gimbal

(Source: http://www.rcmart.hk/images/upload/RCX08-011-2-Axis-Brushless-Gimbal-for-DSLR-Camera-Gimbal-Brushless-Motor-01.jpg)

2.2.1.1. Open source brushless gimbal controller

The brushless gimbal controller board is used with brushless DC (BLDC) motors. It comprises of sensor unit with gyro and accelerometer. The controller unit receives data from IMU to define camera inclination angles. PID-controller calculates the amount of compensation and sends command to the power unit, which it controls the current in the windings and thus the direction of the vector of magnetic field in the stator. Magnetic field then moves the rotor to the right position.

An open source 2- axis brushless gimbal controller boards based on controlling of Arduino programming are shown below. Each one has some subtle differences but they all perform about the same (Martinez brushless gimbal controller, 2014).

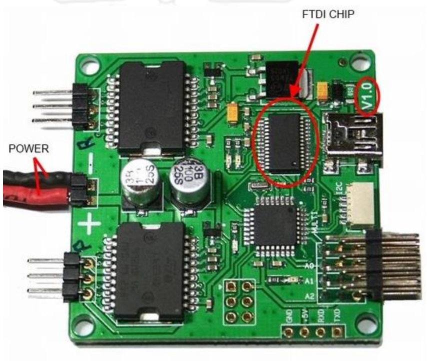


Figure 2.6 2-axis brushless gimbal controller (BGC) RC Timer V1.0 board

(Source: http://www.itsqv.com/QVW/index.php?title=File:Rct02.jpg)

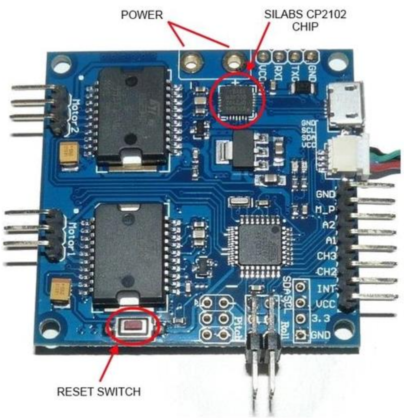


Figure 2.7 2-axis brushless gimbal controller (BGC) 2.0/2.1 style board

(Source: http://www.itsqv.com/QVW/index.php?title=File:BGC20.jpg)

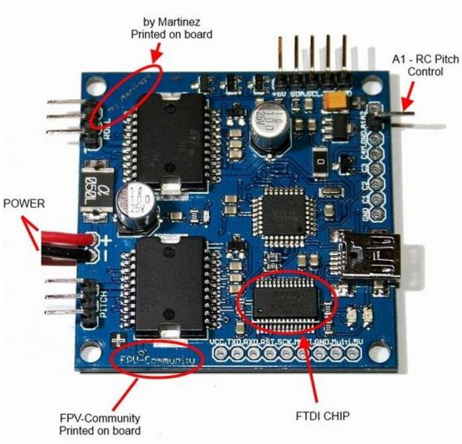


Figure 2.8 2-axis brushless gimbal controller (BGC) Martinez V3 board

(Source: http://www.itsqv.com/QVW/index.php?title=File:Martinez_V3a.jpg)

2.2.1.2. Brushless DC motor (BLDC)

The brushless motor is designed with an armature that is surrounded by one or more electromagnets. The current switch is controlled by a micro controller or computer. The flexibility of the brushless motor is in the analogue hardware or digital firmware; it allows for “micro stepped” operation for slow and/or fine motion control, and a holding torque when stationary.

Almost all brushless motors will have the KV ratings stamped somewhere on them. Some motors will have KV ratings on the motor, but some time will only see on the motor’s spec sheet. The KV is means round per minute (RPM) of motor per volt with no load. A brushless motor with a lower KV will not run fast, but give high torque. On the other hand, a brushless motor with a higher KV will have more top end speed, but give less torque.

Figure 2.9 Brushless DC motor for gimbal

(Source: http://www.axisgimbal.com/direct-drive-brushless-motor/)

The advantages brushless DC motors also known as a Brushless DC (BLDC) when comparing to brush DC motors, are more torque per weight, more torque per watt, more efficient, longer life, lightweight, quiet and overall reduction of electromagnetic interference (EMI). For the application, BLDC are the engines that enable the camera move up, down, left, and right. Typically can see axis gimbals using "Outrunner" brushless motors due to increased torque, power, and speed control.

To control the motors, a special controller was designed. It receives information from gyroscopes and accelerometers mounted on the camera platform. A standard IMU algorithm is used to define camera inclination angles. With the remote control, operator sets desired tilt angle. PID-controller calculates the amount of compensation and sent command to the power unit, which controls the current in the windings and thus the direction of the vector of magnetic field in the stator. Magnetic field moves the rotor to the right position (Aleksey 2012).

2.2.1.3. Mechanical design of gimbal frame

To design gimbal frame should be considered the weight, balance, friction of joints between ring, limitation of rotation angles around axes X and Y of rings, center of rotation of the optical axis, and vibration of UAV. The major principles of mechanical designs of the gimbal frame for the imaging applications are as follow:

- The possibility of precise balancing on three axes: the better the balance, the lower the current need for stabilization.
- each axis of rotation passes through the center of gravity (CG) of the "Camera + frame" system and rigidly connected to the BLDC motor
- Mechanical rigidity of bearing elements: to prevent resonances from working propellers in flight.
- Minimizing friction in the joints: the better the gimbal axis are unleashed from UAV camera platform, the less effort is required from the engine to stabilize.
- Elimination of vibrations from the main UAV frame: the vibrations have a negative impact on the quality of the video. But another negative effect in this system is that vibration will lead to self-excitation of the closed-loop controller (Aleksey 2012).

2.2.1.4. Tuning of BLDC motor

The important parameters for tuning of BLDC motor in order to control its rotation consists of pulse with modulation (PWM) and proportional integral derivative (PID).

1) Pulse width modulation (PWM) parameter

PWM is a direct multiplier of the power provided to the motors by the controller from the battery. Higher 'PWM' usually means better performance especially for erratic flying and or harsh conditions but also means more heat on the motors, which 'PWM' value should be no higher than what is required for stability. In addition, should be check the motors and control board for heat when tuning and lower the 'PWM' value if necessary.

- Low PWM - will result in the gimbal being pushed into position by wind or other weak forces with not enough power to recover.
- High PWM - can result in motor heating, aggressive oscillation and controller failure.

2) Proportional integral derivative (PID) parameter

PID (proportional-integral-derivative) is a control loop feedback system often utilized in automatic control devices. The gimbal PID controller calculates an "error" value as the difference between a measured position (by MPU-6050 sensors) and a desired set point. The controller tries to minimize the error in outputs by adjusting the process control inputs.

- P - is a basic factor of stabilization, sets gain of the signal from the gyro in the feedback loop, describes the power of disturbance response. Higher values imply a stronger response reaction to external disturbance. You should raise this value until the stabilization quality of fast disturbances will be adequate. If the “P” value is too high, the oscillations of the axis will start to be present.
- I - is a gain factor of the absolute error of stabilization (the difference between the target and current angles) in the feedback system. The value I is tuned in the least to achieve minimum angle error.

- D - is a stabilization factor of P. When the value of D is not sufficient, the system may go into a state of self-excitation. The “D” value reduces the reaction speed. This value helps to get rid of low-frequency oscillations. A “D” value that is too high can cause high-frequency oscillations and noise (Martinez brushless gimbal controller, 2014).

2.2.2. Data logger

Data logger is generally made up of many hardware. If consists of some kind of acquisition system such as, analog sensors, digital sensors to obtain data. If also contains some kind of memory for storing sizeable quantities of data over a long period of time (Jeremy 2013).

Furthermore, any sensor whether it’s breakout board SD card, GPS receiver module, MPU and infrared transmitter can be connected to the analog and digital input on the Arduino controller board. Moreover, Arduino controller board can be connected with two or more by using serial port to send and receive the data each other. All data that store on the SD card can be open the file in text editor or other programming, which can be made graphing and visualizing the data to get a better understanding of its trends.

The Arduino doesn’t have a convenient way of storing data, apart from the onboard Electrically Erasable Programmable Read-Only Memory (EEPROM), which is only suitable for storing configuration settings and tiny amounts of data. To store the data logged from many sensor the other external memory such as Secure Digital (SD) cards was used for storing more data, because SD memory card had high capacity than onboard EEPROM on the Arduino (Martin, Joshua et al. 2013).

2.2.2.1. SD cards and SD library

SD cards are one of the best ways to store data externally. It had high capacity that should have plenty of space not only for data logging but also hosting other resource files (such as sound files to play or pictures to display), and Arduino comes with an SD library, making reading from and writing to SD cards easy.

Table 2.2 provides an overview of the main SD class functions available in the SD library. There are many File class functions enabling to read from and write to files. Table 2.3 provides a list of the main File class functions.

Table 2.2 SD library SD class functions (Martin, Joshua et al. 2013)

Function	Description
begin(ChipSelect)	Initializes the SD library and card, optionally passing in the chip select pin.
exists()	Tests if a file or directory is present on the SD card.
mkdir("/directory/to/create") rmdir("/directory/to/remove")	Creates or removes a directory on the card.
open("file/to/open", mode) remove("file/to/remove")	Opens or removes a file at the specified path. When opening, can specify a mode (FILE_READ or FILE_WRITE) if would like to limit access to only reading, or to read and write to the file.

Table 2.3 SD library File class functions (Martin, Joshua et al. 2013)

Function	Description
available()	Checks for bytes available to be read from the file
close()	Closes a file, ensuring all data is written to the SD card
flush()	Ensures data is physically saved to the SD card; used by close()
print() println() write()	Prints or writes data to the file
read()	Reads a byte from the file

- **Sketch for an SD card sensor logger**

This section shows a simple how to log sensor data which get from one of the analog inputs (analog input 0) to SD cards.

SD card data logger:

```
#include <SD.h>
const int chipSelect = 4;
void setup()
{
  Serial.begin(9600);
  Serial.print("Initializing SD card...");
  pinMode(10, OUTPUT);
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    return;
  }
  Serial.println("card initialized.");
}
void loop()
{
  String dataString = " ";
  int sensor = analogRead(0);
  dataString += String(sensor);
  File dataFile = SD.open("datalog.txt", FILE_WRITE);
  if (dataFile) {
    dataFile.println(dataString);
    dataFile.close();
    Serial.println(dataString);
  }
  else {
    Serial.println("error opening datalog.txt");
  }
}
```

Include SD library

Initialize serial for debugging

Set default chip select to OUTPUT

1 Check for SD card

2 Create a string to store data reading

3 Read sensor data and store in data string

4 Open file, print data, and close

5 Print to serial for debugging

❶ First must check to make sure a SD card is present. If it is, everything should open up properly; if not, must return.

❷ Once it’s all set up, the main loop will be start and create a string to store the data.

❸ Read and store the current sensor reading into a String.

❹ Open the data and print the data to the file before calling close.

❺ Print the data to the console for debugging.

2.2.2.2. ArduPilot Mega controller board

ArduPilot (also ArduPilotMega - APM) is an open source unmanned aerial vehicle (UAV) platform. It is able to control autonomous multicopters, fixed-wing aircraft, traditional helicopters and ground rovers (3DRobotics 2007). It was created in 2007 by the DIY Drones community based on the Arduino open-source electronics prototyping platform.

An APM consists of a barometric pressure sensor for altitude, compass, Inertial Measurement Unit (IMU) including the accelerometers, gyroscopes and magnetometers. This can turn any RC vehicle into a fully autonomous unmanned aerial (or ground) vehicle. Depending on which software that choosing, it can fly fixed-wing aircraft, multicopter, helicopters, and control ground rovers and boats.

APM has main feature as follows:

- Free open source firmware supports planes, multicopters (quads, hex, oct, etc), helicopters and ground rovers.
- Simple setup process and firmware loading via a point-and-click utility. No programming required.
- Full mission scripting with point-and-click desktop utilities.
- Can support hundreds of 3D waypoints.
- Two-way telemetry and in-flight command using the powerful MAVLink protocol.

- Choice of free Ground Stations, including the state-of-the-art Mission Planner which includes mission planning, in-air parameter setting, on-board video display, voice synthesis, and full data logging with replay.

- Autonomous takeoff, landing and special action commands such as video and camera controls

- Supports full "hardware-in-the-loop" simulation with Xplane and Flight Gear

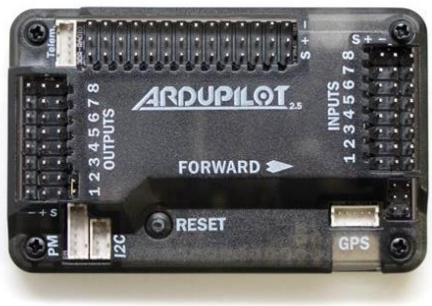


Figure 2.10 APM 2.5 controller board
(Source: http://copter.ardupilot.com/wiki/common-installing-3dr-ublox-gps-compass-module/)

2.2.2.3. GPS receiver module

GPS receiver module is a self-contained high-performance Global Positioning System receiver designed for navigation, tracking and positioning applications. New improved GPS Module has built-in antenna and memory back-up for OEM and hobbyists projects. GPS module is low power consumption and high sensitivity. It is ideal for navigation systems, distance measurements, vehicle monitoring and recording, boating direction, flying direction and location, together with hiking and cross country exploring.

GPS receiver module includes high precision surface mount technology to provide both high accuracy and very compact size. The module can be easily installed on a main board, with all inputs using standard Transistor–Transistor Logic (TTL) signal levels. A National Marine Electronics Association (NMEA) is a standard protocol for communication with GPS receiver modules of a serial link. NMEA format messages are provided to give position, satellite information, time, etc. (Dale 2002). The module can be easily connected directly to a microcontroller to display and record this information. GPS navigation system can connect to a pocket pc for a low cost navigation system.

Figure 2.11 3DR UBlox GPS + Compass Module
(Source: http://copter.ardupilot.com/wiki/common-installing-3dr-ublox-gps-compass-module/)

2.2.2.4. Motion Processing Unit (MPU)

In present, the latest Motion Processing Unit (MPU) integrate 9-axis motion tracking device that combines a 3-axis Micro Electro Mechanical systems (MEMS) gyroscope, a 3-axis MEMS accelerometer, a 3-axis MEMS magnetometer and a Digital Motion Processor (DMP) hardware accelerator engine. The MPU was used in various applications (InvenSense 2011) such as the following:

- BlurFree technology (for Video/Still Image Stabilization)
- AirSign technology (for Security/Authentication)
- TouchAnywhere technology (for “no touch” UI Application Control/Navigation)
- MotionCommand technology (for Gesture Short-cuts)
- Motion-enabled game and application framework
- InstantGesture iG gesture recognition
- Location based services, points of interest, and dead reckoning
- Handset and portable gaming
- Motion-based game controllers
- 3D remote controls for Internet connected DTVs and set top boxes, 3D mice
- Wearable sensors for health, fitness and sports
- Toys

- Pedestrian based navigation
- Navigation
- Electronic Compass

2.3 Photogrammetry

2.3.1. Coordinate systems

Coordinate systems are necessary for the photogrammetric processing of UAV data. The coordinate systems that mention in this section comprises of global coordinate system, local coordinate system, platform and device coordinate systems, and image coordinate systems.

2.3.1.1. Global coordinate system

Global coordinate systems are described as solar or earth fixed coordinate system, where the coordinate axis are for example defined by means of VLBI (Very long baseline interferometry), SLR (Satellite Laser Ranging), GPS measurements or observation to quasars. These systems have a defined origin; for the geocentric coordinate systems this is the center of gravity of the earth. GPS measurements use the WGS84 (World Geodetic System 1984), which is a geocentric coordinate system. The center of gravity results from the global equipotential model of the earth.

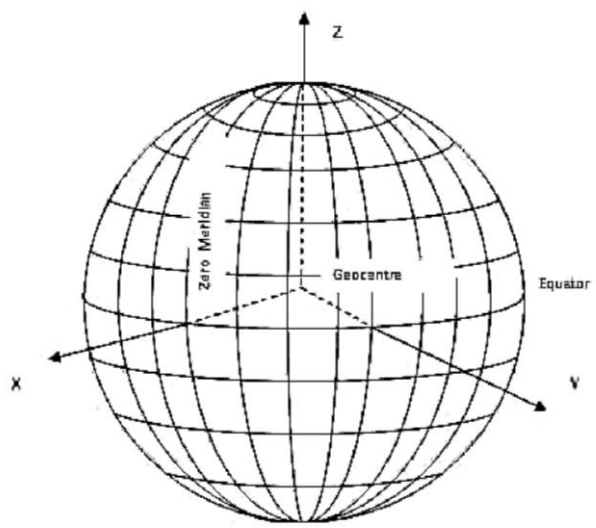


Figure 2.12 Model of the geodetic coordinate system WGS84 (Henri 2009)

2.3.1.2. Local coordinate system

In contrast to the global coordinate system, the local coordinate systems are only applied for a small area. Generally, these systems are right-handed systems centered at a chosen point, with the X-axis being a tangent to the local parallel and looking toward East. The Y-axis is tangent to the local meridian and looks toward North, while the Z-axis is directed upwards using orthometric heights (geoid).

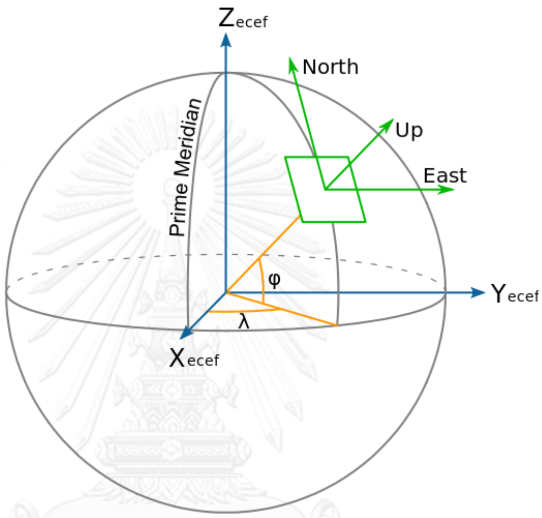


Figure 2.13 Earth centred earth fixed (ECEF) and local east, north, up (ENU) coordinates

(Source: http://en.wikipedia.org/wiki/Geodetic_datum)

2.3.1.3. Platform and device coordinate systems

UAV, Initial navigation system (INS) and camera systems are local 3D Cartesian coordinate systems. The INS system of the UAV determines also the coordinate system of the navigation and stabilization unit of UAV system. With respect to the inertial reference frame, the rotation angles are defined as: Yaw angle (Ψ) around Z-axis, pitch angle (Θ) around Y-axis and roll angle (Φ) around X-axis.

The camera coordinate system depends on the orientation of the camera mounted on the UAV. In general, the X-axis of the camera is mounted parallel to the bow direction of the UAV (flight direction), the Z-axis is parallel to the acquisition axis with the normal to the baseline and the Y-axis completes the right-hand-coordinate system. The center of origin is defined by the projection center of the camera.

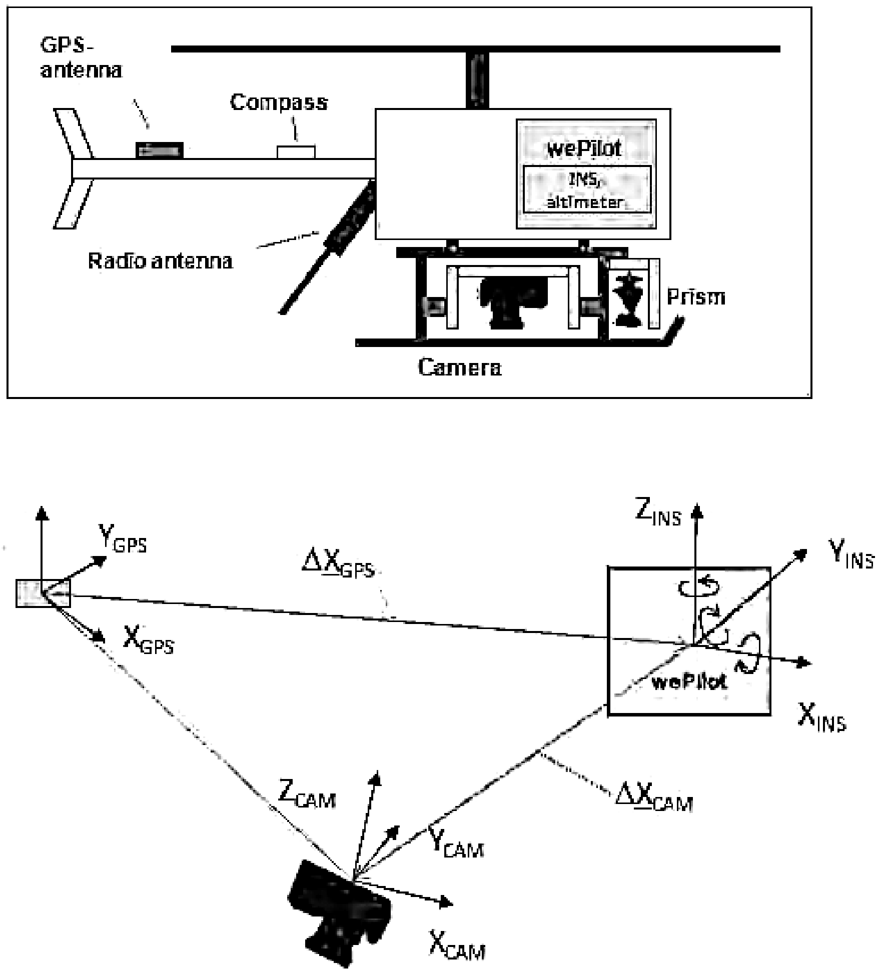


Figure 2.14 Scheme of the main navigation sensors and camera integrated in the UAV (Henri 2009)

2.3.1.4. Image coordinate systems

The sensor coordinate system is a 2-dimensional system, where the y_{Pho}'-axis (rows) is parallel to the acquisition axis and the x_{Pho}'-axis (columns) is defined lateral (see Figure 2.15). The center of the image M' (x_M and y_M) is defined by means of reference or fiducial points (film based system) or the pixel array (electronic systems). The principal point H' (x_p and y_p) is defined as the intersecting point of the optical axis and the image plane. The sensor coordinates (Pixel for digital cameras) can be transformed into metric coordinates using the pixel size p_x and p_y (equation 1).

$$x' = (x_{Pho}' - x_M)\, p_x \\ y' = (y_{Pho}' - y_M)\, p_y \qquad (1)$$

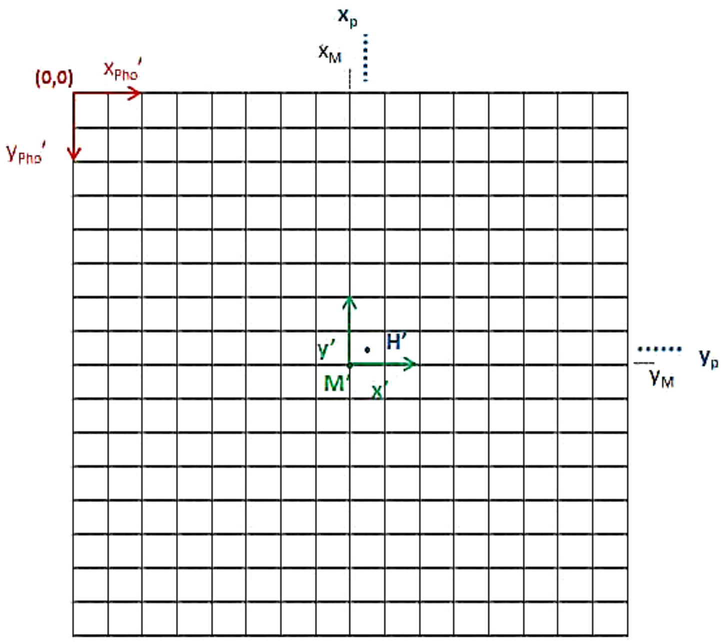


Figure 2.15 Definition of sensor and image coordinate system (Henri 2009)

2.3.2. Geometry of photograph

The vertical and tilted photograph are illustrated by the nadir angle (*v*), which is the angle between the optical axis and the vertical line through the perspective center (nadir- line) and is the composition of depression angle. Therefore, images are classified according to their degree of tilt from vertical (James S. Aber et al., 2010).

- True vertical (v = 0°, often difficult in practice particular with typical small UAV platform)
- Near-vertical or slightly tilted (v < 3°)
- Low oblique (v ≥ 3°, horizontal not visible)
- High oblique (usually v > 70°, horizontal visible)

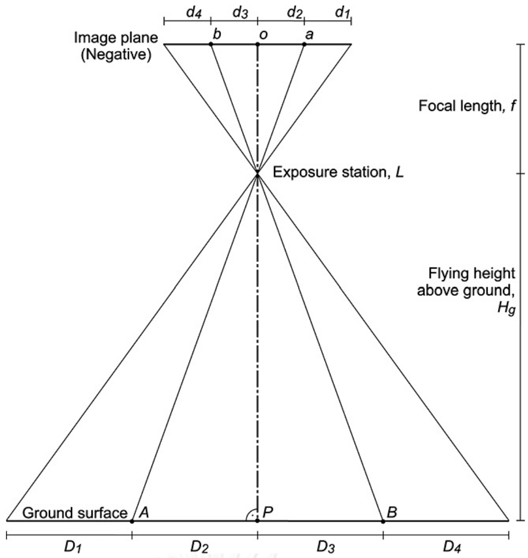


Figure 2.16 Vertical photograph taken over completely flat terrain

(James, Irene et al. 2010)

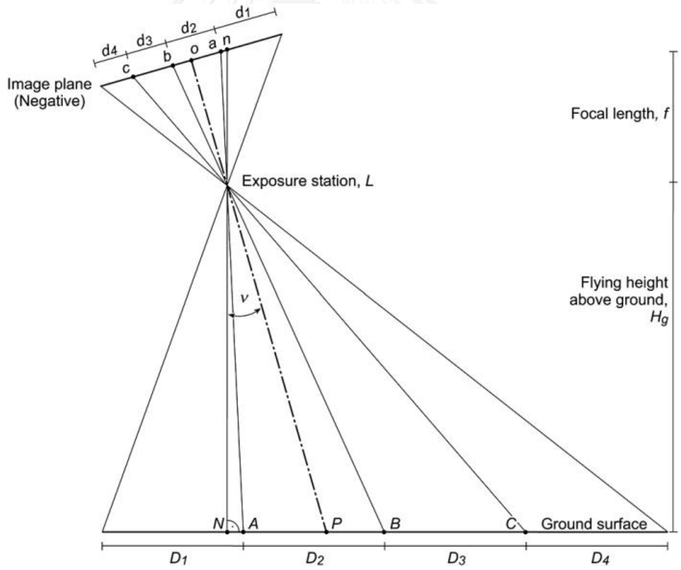


Figure 2.17 Tilted photograph taken over completely flat terrain

(James, Irene et al. 2010)

2.3.3. Interior orientation

The interior orientation, or sensor model, defined the sensor or the camera characteristics required for the reconstruction of the object space bundle of rays from the corresponding image points. These characteristics comprises of focal length or principle distance, the location of the principal point in the image plane, and description of the lens distortion. The principal point (o) is defined as the intersection of the optical axis with the image plane and falls quite close to the origin of the image coordinate system at the image center (James, Irene et al. 2010).

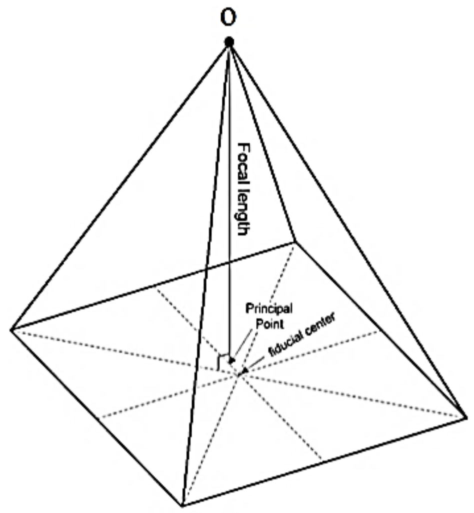


Figure 2.18 The focal length and position of principal point of photograph (Nielsen 2004)

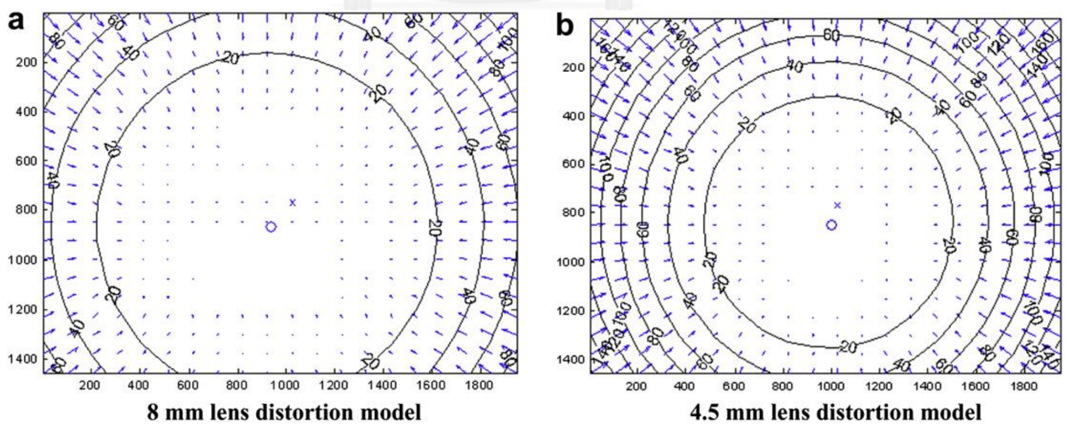


Figure 2.19 Lens distortion models (Xiang and Tian 2011)

2.3.4. Exterior orientation

The exterior orientation or platform model establishes the position and orientation of the bundle of rays with respect to the object coordinate system. The element of exterior orientation includes the position (X_0, Y_0, Z_0) and three orientation (ω , φ , κ), which can be determined by using the GPS and IMU information. (James, Irene et al. 2010).

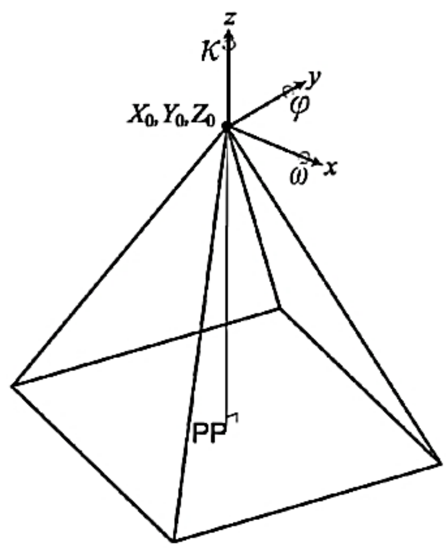


Figure 2.20 The exterior orientation parameters of photograph (Nielsen 2004)

2.3.5. Direct georeferencing

The direct georeferencing is defined as the direct measurement of the exterior orientation parameters for each image or scanning line of line sensors and aerial laser scanners. Thus, the exterior orientation parameters are observed through the GPS and INS sensors integrated into the platform.

Using the coordinate systems defined previously, for each point observed in the image (P), the relation between camera and ground coordinate system (Ob) can be described with the following equation (2).

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix}_{Ob} + \lambda R_P^{Ob}(\omega,\varphi,\kappa) \begin{pmatrix} (x'-x_p) \\ (y'-y_p) \\ -f \end{pmatrix}_P \quad (2)$$

The parameters of the interior orientation are the principal point (x_p and y_p) and the focal length f, while the parameters of the exterior orientation consist of coordinates of the projection center in the object coordinate system and the rotations matrix $R_P^{Ob}(\omega,\varphi,\kappa).R_P^{Ob}$ is the rotation matrix from image coordinate into the object coordinate system. λ is a scale factor.

The position of the camera from GPS and INS in the UAV are similar to the components that attached in manned aircrafts (Henri 2009). Therefore, the equation (2) can be expanded to equation (3).

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix}_{Ob} + \lambda R_{INS}^{Ob} \left(\lambda R_P^{Ob} \begin{pmatrix} (x'-x_p) \\ (y'-y_p) \\ -f \end{pmatrix} + \begin{pmatrix} \Delta X_{cam} \\ \Delta Y_{cam} \\ \Delta Z_{cam} \end{pmatrix} + \begin{pmatrix} \Delta X_{GPS} \\ \Delta Y_{GPS} \\ \Delta Z_{GPS} \end{pmatrix} \right)_P \quad (3)$$

The single terms are defined as follows:

$P = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{Ob}$ Object point in object coordinate system (Global or local coordinate system) Object

$\begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix}_{Ob}$ GPS/INS position in the object coordinate system Object

$\begin{pmatrix} \Delta X_{cam} \\ \Delta Y_{cam} \\ \Delta Z_{cam} \end{pmatrix}$ Vector between INS and projection center of the camera, defined in the INS coordinate system.

$\begin{pmatrix} \Delta X_{GPS} \\ \Delta Y_{GPS} \\ \Delta Z_{GPS} \end{pmatrix}$ Vector between GPS (antenna phase center) and INS, defined in the INS coordinate system.

R_{INS}^{Ob} Rotation of the INS coordinate system into the object coordinate system Object the rotation angles are defined through the GPS/INS module.

R_P^{Ob} Rotation of the image coordinate system P into the INS coordinate system

2.4 Related researches

Dae woo Lee et al., (2010) developed electro-optical system (EOS) for small unmanned aerial vehicle (UAV) especially fixed wing type, to track images and measure 3-D position of target in real time using process of Kalman filter. This developed system consists of transmission system, gimbal controller, and image acquisition system, as well as algorithm for the EOS such as arithmetic and linear parametric varying (LVP). The results of study had found the difference between a known target position and target position obtained by image becomes minimum value. Using developed system with the modified LPV based nonlinear filter, can be verify that the average error is within 10 m on 3-D measurement.

Deok-Jin Lee et al., (2010) developed feature following control, distributed navigation algorithm and low-cost imaging sensor unit such as gimbaled camera system to the target tracking for real-time visual surveillance using a small unmanned aerial vehicle, which developed the algorithms and 2-axis servo (pan, and tilt) gimbaled system in order to usage for track the feature path of a general road. The results of study had found the feature following errors between the commands and true UAV track and the mean value of the deviation errors reside within 40m and the maximum deviation from the desired trajectory is about 50m during the feature following application.

Ashraf Qadir et al., (2013) developed a vision-based neuro-fuzzy controller for a two axes (pan, and tilt) gimbal system with small UAV, which used the vision-based object detection of position and difference in positions as inputs generate pan and tilt motion and velocity commands for the gimbal motion controller that in turns moves the camera in order to keep the interest object at the center of the image frame. The results of study had found the controller is able to converge effectively and generate accurately position and velocity commands to keep the object at the center of the image frame.

Aleksander Nawrat et al., (2013) designed and created prototype of stabilized UAV gimbal for day-night surveillance applications and developed algorithm for stabilization. The prototype gimbal is capable of pan and tilt rotation of a camera in range of full 360 degrees by using two servos with 10 bit resolution which full turn takes approximately 1.2 second, the feature of gimbal is used for tracking detected target pan and tilt gimbal's capabilities without the need to change the trajectory of the whole UAV. The results of study have shown a great potential of gimbal for target detection and tracking object while it was rotate by pan and tilt.

A gimbal has been developed for mounting on UAV and also used in aerial photography applications for tracking the object or target in the real time for surveillance applications.

CHAPTER 3
RESEARCH METHODOLOGY

This chapter describes the detail in designing and creating a gimbal (camera controller) and data logger for using in aerial mapping on UAV. The workflow of this research work is shown in following Figure 3.1.

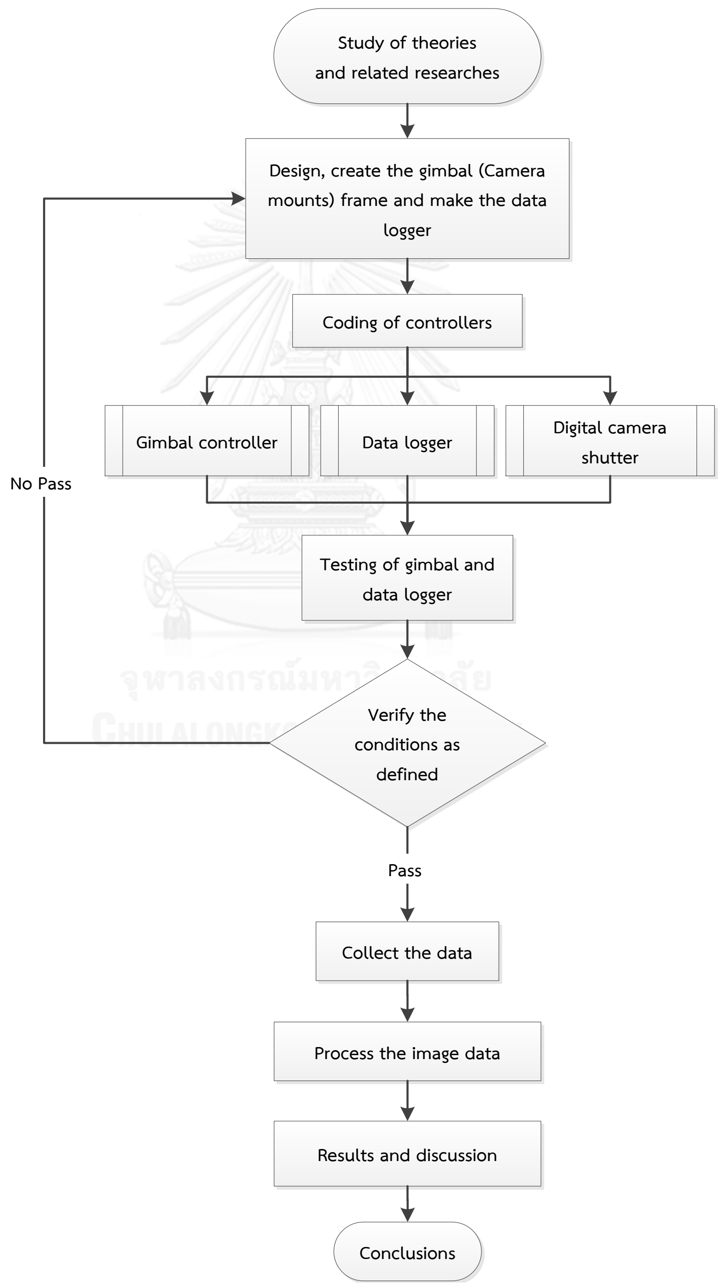


Figure 3.1 Diagram of the research workflow

3.1 Design and creating of gimbal

3.1.1 Design of gimbal frame

The commercial software as Autodesk AutoCAD 3D has been used for designing and modeling 2-axis brushless gimbal (Figure 3.2). The purpose of this work was to create a camera controller that is capable of attaching on small UAV such a multi-rotor. The issue have to be considered for designing the gimbal are as follows:

- Weight of gimbal frame
- Dimensions of gimbal frame for attaching a camera
- Balance of a camera in 2-axis
- Friction of joints between ring
- Limitation of rotation angles around axes X and Y of rings
- Center of rotation of the optical axis
- Vibration of UAV
- Space for placing of controller boards and motors.

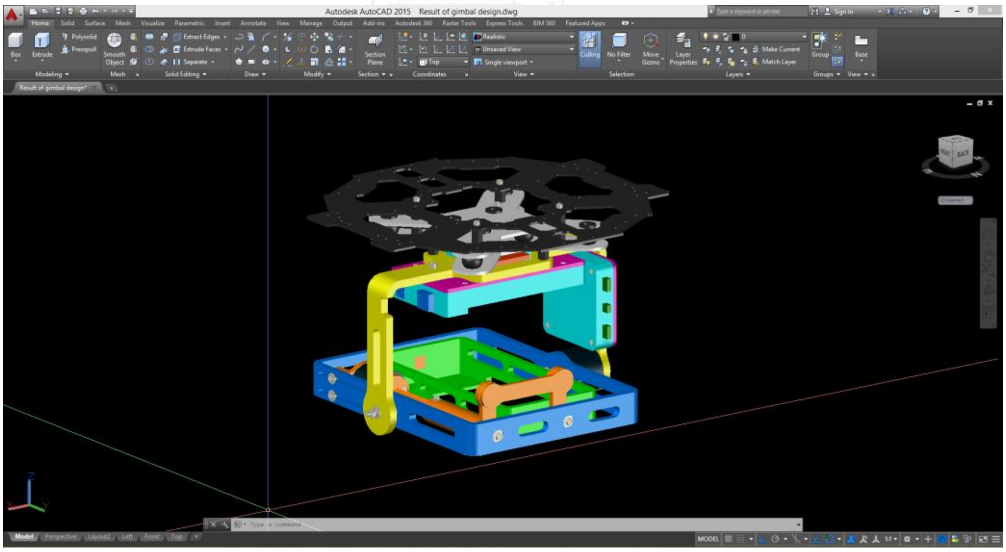


Figure 3.2 Modeling of 2-axis brushless gimbal frame

A design of this gimbal is divided into three main components. For the first component is a part of gimbal ring, which consist by three rings and steering arm. The second is an anti-vibration which include two plates with holes in order to put a rubber ball. The last one is a case for covering of controller boards. In addition it designed to able to rotate a camera around X-axis (roll) and Y-axis (pitch) in range of maximum at -45° to 45° by using Brushless DC (BLDC) motor, which has shown in Figure 3.3 the developed model of 2-axis brushless gimbal frame for small UAV.

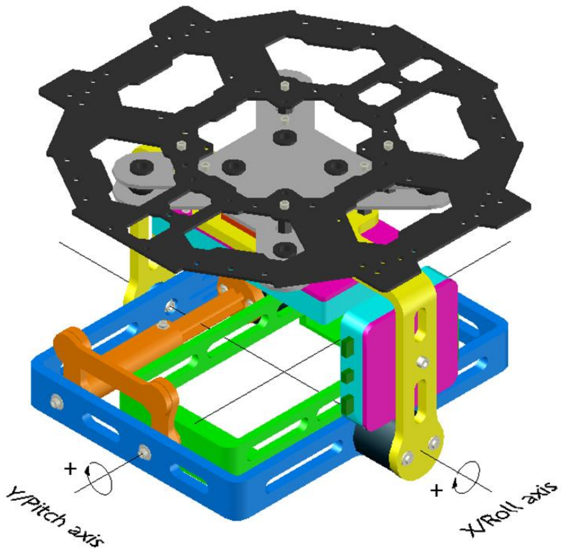


Figure 3.3 The developed model of 2-axis brushless gimbal frame for small UAV (Drone)

3.1.2 Creating of gimbal frame

A 3D printer (Figure 3.4) was used in order to create the objects from designed 3D models. The 3D printer is an appropriate tool for build up the prototype of gimbal frame.

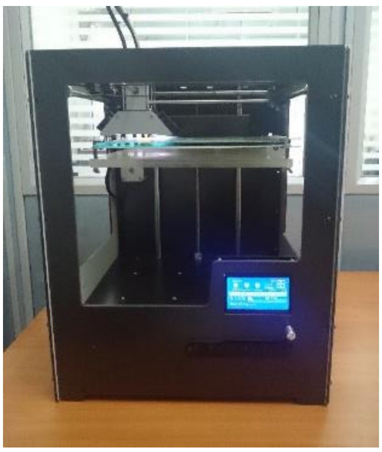


Figure 3.4 An ExtraBot 3D printer

A KISSlicer software was used for slicing of 3D models and creating of G-code file (see Figure 3.5). G-code is a file that used for printing the gimbal frame. A polylactic acid (PLA) was used as material for printing gimbal.

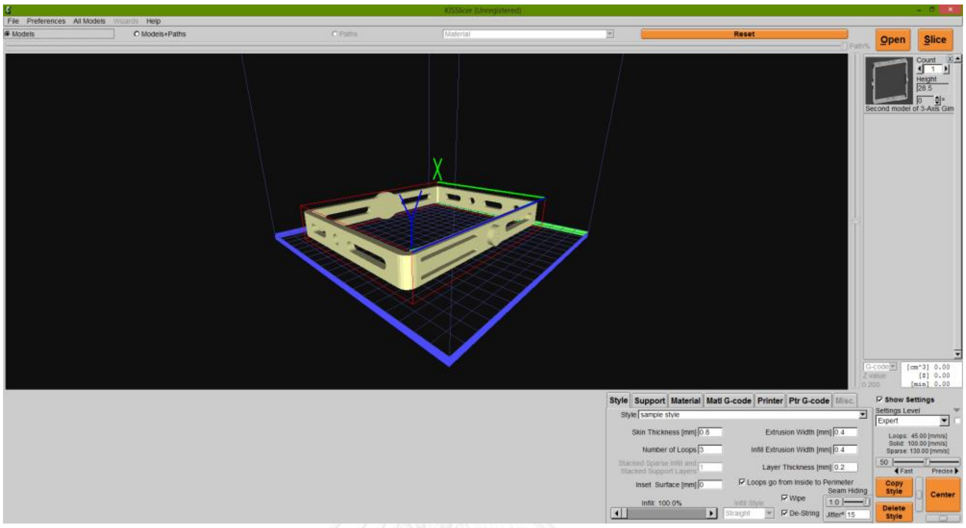


Figure 3.5 The slicing of a model by using KISSlicer

The object of the gimbal frame that get from the printing by using a 3D printer has shown in the Figure 3.6.

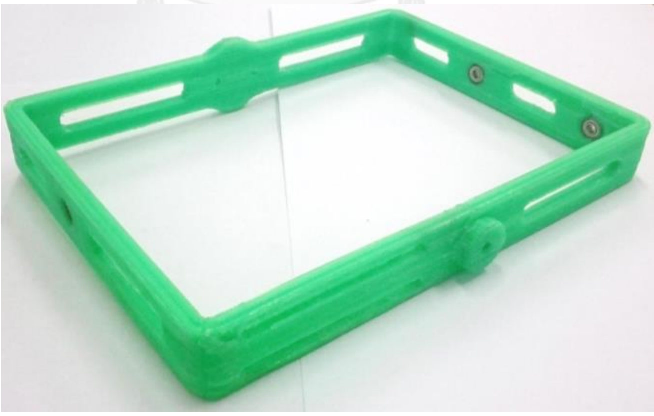


Figure 3.6 The object that get from printing by using 3D printer

3.1.3 Selecting of brushless gimbal controller and BLDC motors

A brushless gimbal controller board is necessary for stabilizing a camera. The selecting of brushless gimbal controller board it is necessary because the suitable controller board should be able to customize the firmware of controller. Therefore, the open source RCTIMER 2-axis brushless gimbal controller board and IMU (see Figure 3.7) was selected and used for customizing of controller firmware and tuning of brushless direct drive (BLDC) parameters such as, PWM and PID in order to control the rotation of BLDC motors to defined direction.

BLDC motors has more advantage such as, controllability over wide range of speeds, capable of rapid acceleration and deceleration, more torque per weight, no mechanical wear problem due to commutation, better heat dissipation arrangement.

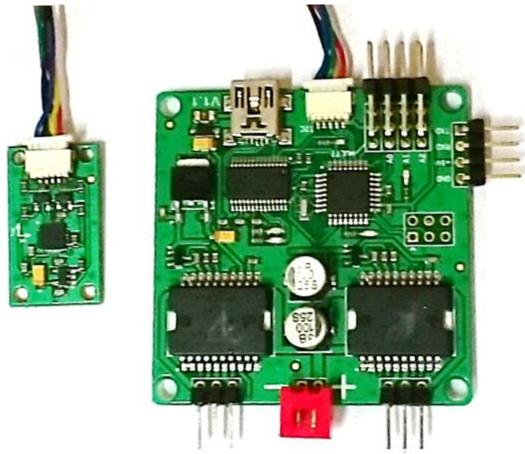


Figure 3.7 RCTIMER 2-axis brushless gimbal controller & IMU v1.0

A BLDC motor with a lower KV rating was selected because it give high torque and low speed. An EMAX GB2210-110 KV (see Figure 3.8) was used because it is light weight (approximate 48g) and give sufficient torque (approximate 150-200g) to rotate a compact digital camera for stabilization.

Figure 3.8 EMAX GB2210-110 KV brushless DC motor for gimbal

3.2 Development of data logger

The main hardware was used in this data logger is consist of ArduPilot Mega 2.5 (APM 2.5) controller board, GPS module, Aruduino MEGA 2560, IMU (MPU-9150), SD-MMC card breakout board, infrared (IR) transmitter, and Pentax Optio WG-2 compact digital camera. All hardware required for data logger are shown in Figure 3.9.

Figure 3.9 The main hardware of data logger

The data logger is made up of two main sections are the UAV system and data logging system (shown in Figure 3.10). The first section, the position parameters (X/longitude and Y/latitude) of the images get from GPS module, an altitude above the ground (Z/altitude) get from barometer (altimeter). The orientation parameter (yaw or heading) of the image get from IMU. Barometer and IMU are embedded in ArduPilot Mega 2.5 (APM 2.5) controller board. These all parameters can be obtained from APM through a serial port.

The second section, two orientation parameters (roll and pitch) of the images will be tracked by second IMU (MPU-9150) which attached on a camera. Shuttering camera is controlled by using infrared (IR) transmitter. All parameters from two sections are recorded into SD card in text file (CSV file format). An Arduino Mega 2560 controller board is applied as the main controller of data logging system.

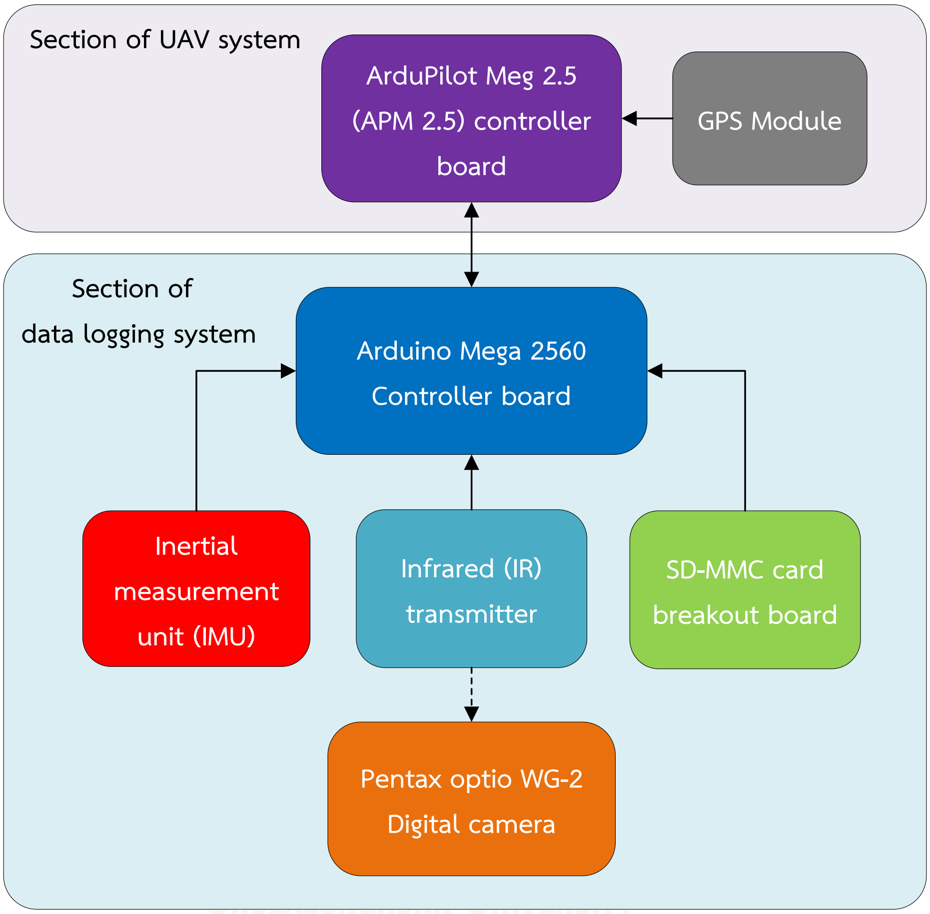


Figure 3.10 The diagram of developed data logger for UAV platform

3.3 Programing of controllers

3.3.1 Brushless gimbal controller

Brushless gimbal controller is very important thing and it is a part of the gimbal (camera controller). To stabilize a camera, it will be installed into the gimbal frame in order to control brushless DC motors for stabilizing camera according to the program. The firmware of brushless gimbal controller is based on C/C++ programing language and it is an open source, it was used to control the rotation of camera in one, two or three axes. To achieve accordingly to the purpose of this research is stabilizing camera. The brushless gimbal controller firmware will be modified to control a camera for stabilizing in the vertical plane by using brushless gimbal controller and BLDC motors.

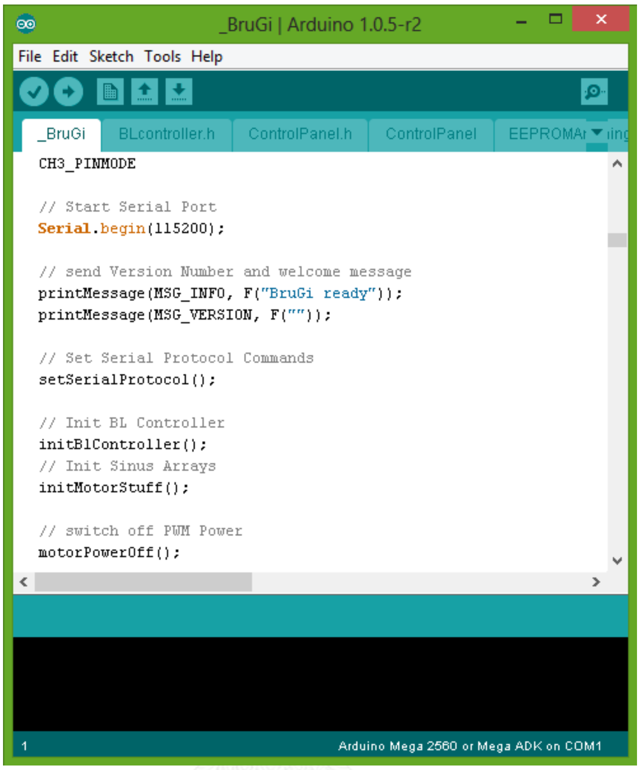


Figure 3.11 Open source Arduino software (IDE)

Furthermore, the open source Arduino software (IDE) has been used for customizing and uploading firmware which shows in Figure 3.11. A Brushless-Gimbal-Tool for BruGi firmware (see Figure 3.12) is used for tuning gimbal parameter.

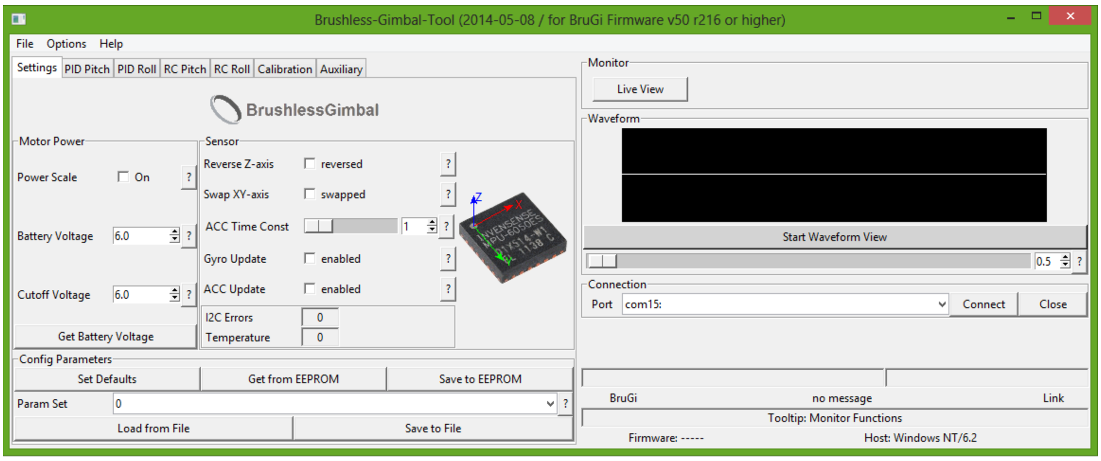


Figure 3.12 Brushless-Gimbal-Tool for BruGi firmware

3.3.2 Data logger

UAV system and data logging system are the main system of developed data logger. Both systems are running on microcontroller by using open-source Arduino software (IDE), which programming language is based on C/C++. In addition, a data logging system consists of two operations including sending and receiving of data on microcontroller.

3.3.2.1 Sending of data

The UAV system was used for sending data, it comprises of ArduPilot Mega 2.5 with completely soldered serial pins (see Figure 3.13. Generally, serial pin of APM 2.5 board is not solder) and 3DR UBLOX GPS + compass module. The data will be sent from APM 2.5 board to another controller board like an Arduino Mega 2560 that connected SD card for recording data, due to APM 2.5 is not supported SD card. The Arduino serial communication was used in order to communicate between two Arduino controller boards for sending and receiving data with each other on the microcontroller.

APM 2.5 firmware is modified to define and send the required parameters such as, yaw (heading), X/longitude, Y/latitude, and Z/altitude to the main controller board of data logger via serial port.

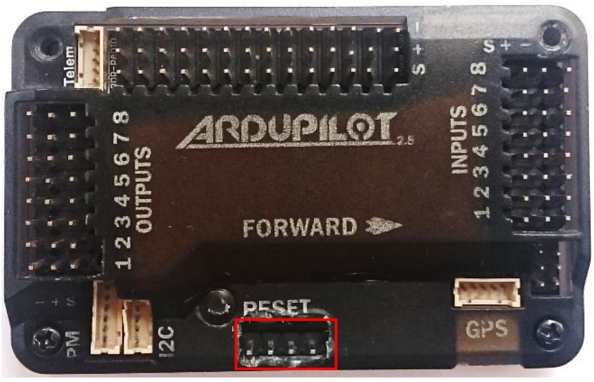
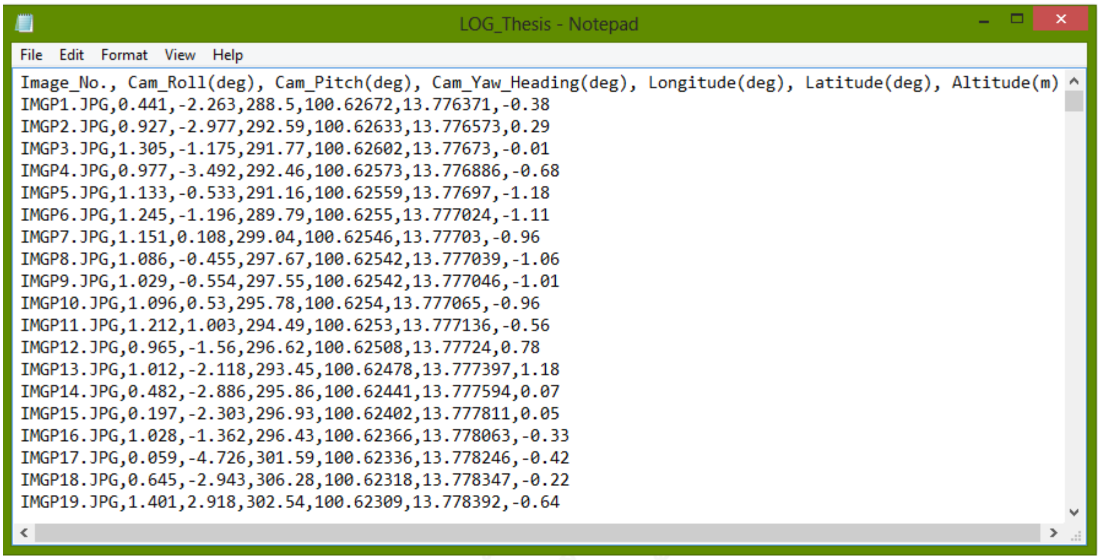


Figure 3.13 ArduPilot Mega 2.5 with completely soldered serial pins

3.3.2.2 Receiving of data

To receive the data from one Arduino to another Arduino controller board such as ArduPilot Mega 2.5 to Arduino Mega 2560 controller board, the serial pins of the both controller boards are used for transferring of data. Received data consists of data that get from serial communication and data are tracked from IMU (MPU-9150). All data will be recorded together into the SD card via SD card during the imaging.

Data will be recorded in the text file with comma delimited (CSV) format including of roll, pitch, yaw (heading), X/longitude, Y/latitude, and Z/altitude. These data are exterior orientation parameters that required for the aerial mapping. The sample of logging data is shown in Figure 3.14.


LOG_Thesis - Notepad

File Edit Format View Help

```
Image_No., Cam_Roll(deg), Cam_Pitch(deg), Cam_Yaw_Heading(deg), Longitude(deg), Latitude(deg), Altitude(m)
IMGP1.JPG,0.441,-2.263,288.5,100.62672,13.776371,-0.38
IMGP2.JPG,0.927,-2.977,292.59,100.62633,13.776573,0.29
IMGP3.JPG,1.305,-1.175,291.77,100.62602,13.77673,-0.01
IMGP4.JPG,0.977,-3.492,292.46,100.62573,13.776886,-0.68
IMGP5.JPG,1.133,-0.533,291.16,100.62559,13.77697,-1.18
IMGP6.JPG,1.245,-1.196,289.79,100.6255,13.777024,-1.11
IMGP7.JPG,1.151,0.108,299.04,100.62546,13.77703,-0.96
IMGP8.JPG,1.086,-0.455,297.67,100.62542,13.777039,-1.06
IMGP9.JPG,1.029,-0.554,297.55,100.62542,13.777046,-1.01
IMGP10.JPG,1.096,0.53,295.78,100.6254,13.777065,-0.96
IMGP11.JPG,1.212,1.003,294.49,100.6253,13.777136,-0.56
IMGP12.JPG,0.965,-1.56,296.62,100.62508,13.77724,0.78
IMGP13.JPG,1.012,-2.118,293.45,100.62478,13.777397,1.18
IMGP14.JPG,0.482,-2.886,295.86,100.62441,13.777594,0.07
IMGP15.JPG,0.197,-2.303,296.93,100.62402,13.777811,0.05
IMGP16.JPG,1.028,-1.362,296.43,100.62366,13.778063,-0.33
IMGP17.JPG,0.059,-4.726,301.59,100.62336,13.778246,-0.42
IMGP18.JPG,0.645,-2.943,306.28,100.62318,13.778347,-0.22
IMGP19.JPG,1.401,2.918,302.54,100.62309,13.778392,-0.64
```


Figure 3.14 The sample of recorded data of exterior orientation parameters in text file that get from data logger

3.3.3 Digital camera shutter

Controlling of camera shutter is very important for the imaging, because the imaging on UAV needs a controller to communicate with a digital camera for sending commands to control a camera shutter. The commands can be sent via USB cable, Infrared (IR), and Wi-Fi connection which depends on the supporting of a camera. In this research uses the IR transmitter, Arduino Mega 2560 and Arduino IR control library (Sebastian 2011) in order to control the Pentax Optio WG-2 camera shutter.

3.4 Integration of gimbal and data logger

To get the nearly vertical image and exterior orientation parameters on UAV platform is very importance for photogrammetry. Gimbal and data logger have to be integrated together to achieve the requirment. A prototype of brushless gimbal and data logger (Figure 3.15) consists of 2-axis X and Y, allowing rotations in the range between -45° to 45° for stabilizing a camera. The hardware was used in this prototype is made of 2 BLDC motors, 2-axis brushless gimbal controller and IMU.

The exterior orientation parameters can be taken from GPS and IMU. GPS module was used for tracking the position of image. IMU was also applied for tracking of the orientations by attaching on a camera. In addition, infrared (IR) transmitter was utilized for controlling of camera shutter, Moreover, SD-MMC card breakout board was used for data logging. All these things are connected with the Arduino Mega controller board.

The frame of gimbal, anti-vibration and case for covering the controller board were created from 3D printer. A prototype of 2-axis brushless gimbal and data logger have dimensions about 128.46mm, 168mm, and 131.50mm of width, length and height respectively. In addition, they have approximate weight 400g and 650 g by excluding and including the battery and Pentax Optio WG-2 compact digital camera respectively.

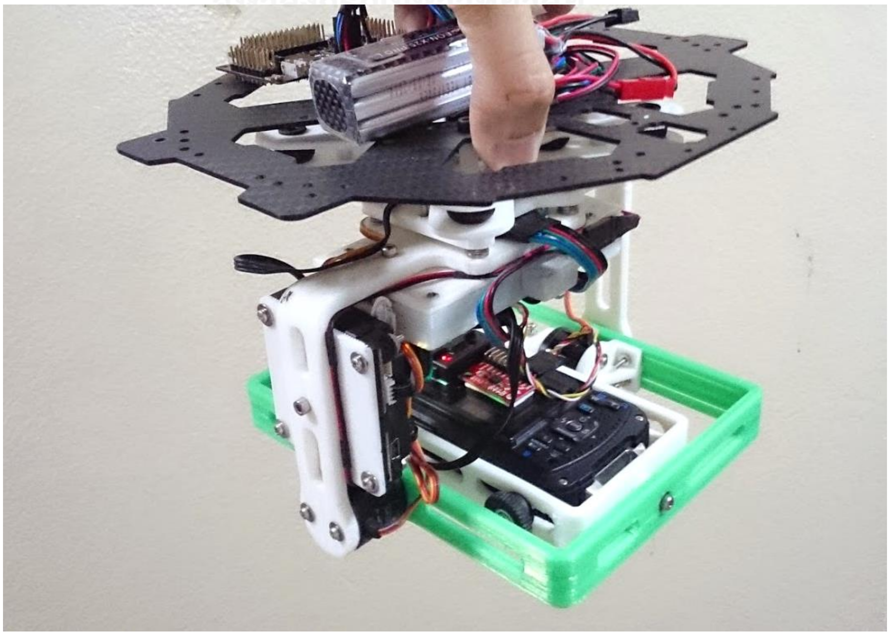


Figure 3.15 A prototype of 2-axis brushless gimbal and data logger

3.5 Testing of gimbal and data logger

The purpose of testing was to test gimbal's stabilization and data logging's capabilities. The first goal was to control the imaging to get the nearly vertical images, which means that the images orientation for roll and pitch will be controlled in range of -3 to 3 degrees (less than 3 degree is acceptable) of roll and pitch and not over than this against vertical plane. The second goal was to record the exterior orientation parameters of each image, and to check the parameters whether their parameters are correct or not, by comparing the exterior orientation parameters between derived from data logger directly and derived from photos alignment and orthophoto mosaic in Agisoft PhotoScan software.

3.5.1 Testing in laboratory

3.5.1.1 Camera calibration

Cameras for mapping should be carefully calibrated in order to determine precise and accurate values for a number of constants. The elements of interior orientation parameters of a camera can be determined from various software such as Agisoft Lens, CalCam, PhotoModeler, iWitness software, and so on. This research uses an Agisoft Lens software in order to calibrate a Pentax Optio WG-2 digital camera for determining of constant of interior orientation. The parameters include:

- Focal length (f_x, f_y), which f_x is horizontal focal length, in pixels and f_y is vertical focal length, in pixels
- Principal point coordinates (cx, cy), which cx is X coordinate of the principal point and cy is Y coordinate of the principal point
- Radial distortion coefficients, using Brown's distortion model (K1, K2, K3, P1, and P2).

Furthermore, the constants of interior orientation parameters that get from camera calibration will be used in the digital image processing such as orthophoto mosaicing in the section 3.7. Figure 3.16 shows the results of camera calibration of Pentax Optio WG-2 digital camera.

Parameter	Value	Std Error
Image width	4608	
Image height	3456	
Focal length (x)	3738.58	3.25765
Focal length (y)	3742.51	3.29664
Principal point (x)	2320.22	0.885692
Principal point (y)	1773.21	0.781233
Skew	0.998389	0.125122
Radial K1	0.0226349	0.00146816
Radial K2	-0.0690811	0.00596591
Radial K3	0.0635136	0.00731344
Radial K4	0	0
Tangential P1	0.00498367	5.17542e-05
Tangential P2	0.00152942	6.74066e-05

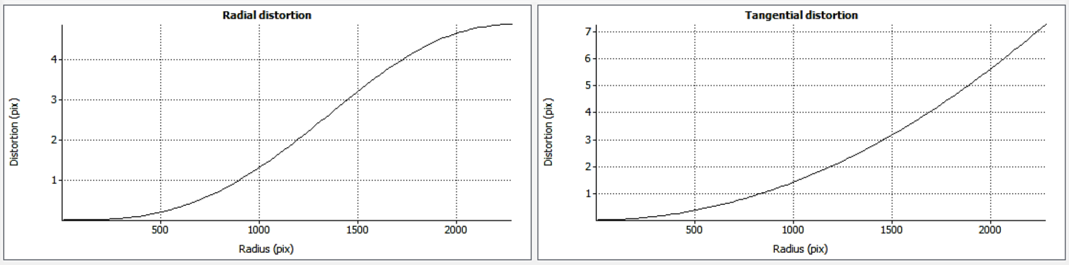


Figure 3.16 The results of camera calibration of Pentax Optio WG-2 digital camera

3.5.1.2 Gimbal calibration

Gimbal is used to stabilize a camera for vertical imaging in the aerial mapping applications. It should be calibrated to vertically as much as possible before using in real environment for data collection.

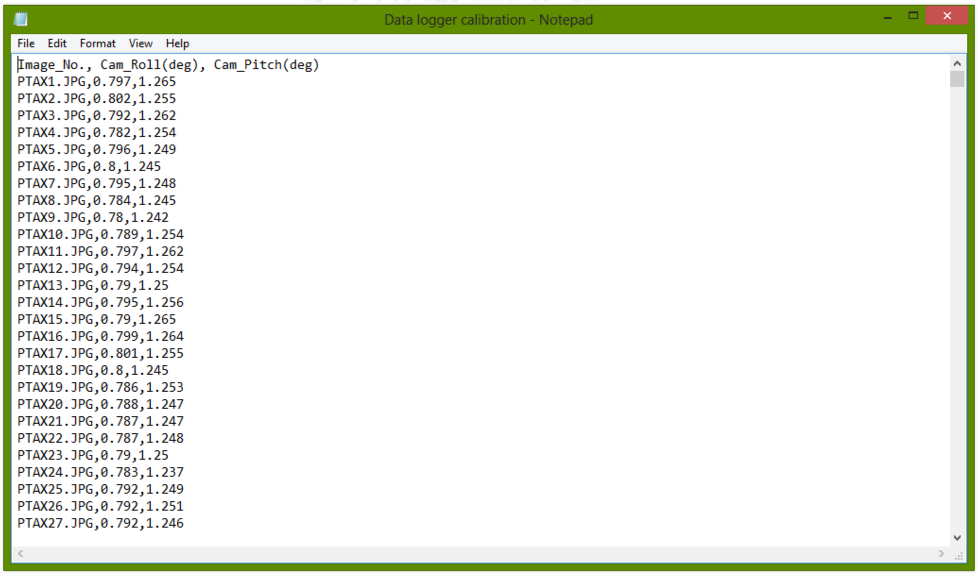
The orientation of optical axis of camera will be set to vertical by using the tuning of BLDC motor and use the pound bubble tube to check its orientation. In addition, IMU that attached on a camera for tracking of inclination of camera will be set the angular offsets to nearly zero as much as possible after the tuning is finished. Due to it is very sensitive to the motion of orientation changes.

3.5.1.3 Data logger calibration

To record the images and exterior orientation parameters together, data logger calibration is very important thing have to do. Due to the limitation of camera speeds for saving an image, which is different for each camera models. The main issue is the time interval for recording data. If set the interval of imaging and data logging are faster than the maximum speed of camera, it may cause the image and exterior orientation parameters have not equal number. Moreover, an IMU attached on a camera for tracking of its orientation may not parallel with the optical axis of camera and it may cause the angular offset of X (roll) and Y (pitch) different from the actual inclination of camera. Therefore, the calibration is necessary and indispensable for data logging system.

As the calibration of time interval for the imaging of selected camera (Pentax Optio WG-2), the result has shown that time interval cannot specify absolutely, because a part of the speed factor for saving an image is also depend on the speed of memory card. However, to specify the time interval for recording of the images and exterior orientation parameters together should be used the speed that get from the calibrated camera essentially. Which for the result of the use of Pentax Optio WG-2 camera, the optimal time interval for recording are 4.6 seconds or more.

In addition, the observation results of angular changing value of IMU (Figure 3.17) when it is stable on the gimbal about an hour has been shown that the IMU can be provided the precision of roll and pitch angles are approximate ± 0.040°.



Image_No., Cam_Roll(deg), Cam_Pitch(deg)
PTAX1.JPG,0.797,1.265
PTAX2.JPG,0.802,1.255
PTAX3.JPG,0.792,1.262
PTAX4.JPG,0.782,1.254
PTAX5.JPG,0.796,1.249
PTAX6.JPG,0.8,1.245
PTAX7.JPG,0.795,1.248
PTAX8.JPG,0.784,1.245
PTAX9.JPG,0.78,1.242
PTAX10.JPG,0.789,1.254
PTAX11.JPG,0.797,1.262
PTAX12.JPG,0.794,1.254
PTAX13.JPG,0.79,1.25
PTAX14.JPG,0.795,1.256
PTAX15.JPG,0.79,1.265
PTAX16.JPG,0.799,1.264
PTAX17.JPG,0.801,1.255
PTAX18.JPG,0.8,1.245
PTAX19.JPG,0.786,1.253
PTAX20.JPG,0.788,1.247
PTAX21.JPG,0.787,1.247
PTAX22.JPG,0.787,1.248
PTAX23.JPG,0.79,1.25
PTAX24.JPG,0.783,1.237
PTAX25.JPG,0.792,1.249
PTAX26.JPG,0.792,1.251
PTAX27.JPG,0.792,1.246

Figure 3.17 The observation results of angular changing value of IMU

3.5.2 Testing in real environment

For testing in real environment, a multi-rotor (Hexacopter) and ground station (computer laptop) has been used. The developed 2-axis brushless gimbal and data logger were also tested. Before flying to test it, an UAV system and their important equipment for the flying should be checked such as, battery volts, remote control, telemetry, including a preparing of 2-axis brushless gimbal and data logger for collecting data. An UAV and ground control station (GCS) are connected wirelessly to monitor the UAV status during flying (shown in Figure 3.18).

Figure 3.18 Preparing of 2-axis brushless gimbal and data logger for testing in real environment by using UAV

Mission planner is a software used for uploading way points and monitoring. It can be used to design the flight path, altitude, speed, and so on (see Figure 3.19). The modes of auto takeoff, way points and auto landing are supported.

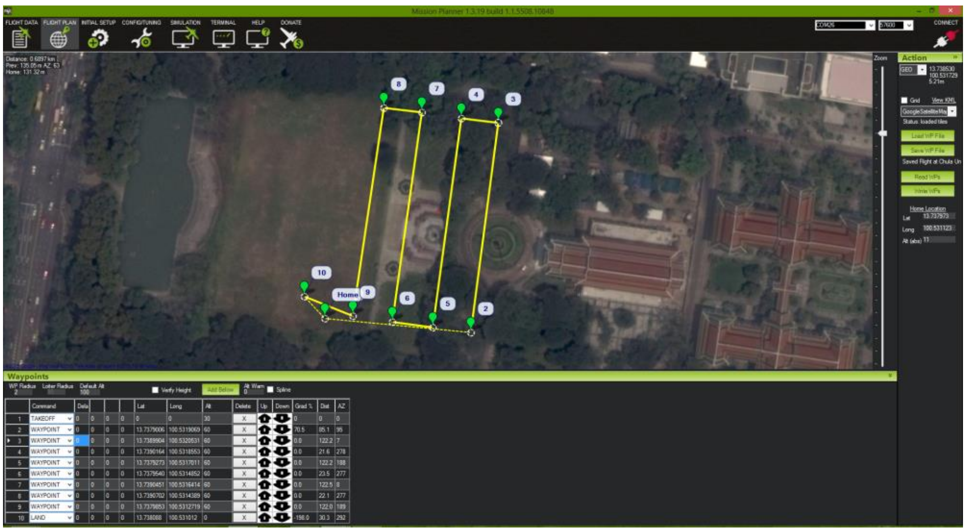


Figure 3.19 The flight planning in Mission Planner software

The test of 2-axis gimbal and data logger will be evaluated from data log file of their operations to check the images and orientation parameters whether there are relative with each other or not.

3.6 Data collection

To evaluate the capabilities of developed 2-axis gimbal and data logger, the data collection will be divided by two cases to illustrate clearly different of their results. For the first case (case A), gimbal and data logger were tested together in order to keep images orientation of roll and pitch. The second case (case B), only data logger were tested to get orientation parameters of image without controlling by gimbal. Both cases were tested in the same condition e.g. flight path, altitude, time period, and weather. The experimental area for collecting data is located in Chulalongkorn University (Equestrian Statue of King Chulalongkorn).

3.7 Data processing

The 3D reconstruction and photogrammetry software like an Agisoft PhotoScan is used for processing of the data that get from the developed gimbal and data logger in the real experiment. Images and their exterior orientation parameters of both cases including the interior orientation parameters of a camera has been directly used in automated orthophoto mosaicking processing, without the use of GCP.

Data processing of both cases used the same procedures and setting. The procedures of image processing and 3D model construction using Agisoft PhotoScan which comprises four main steps as follows:

- Camera alignment.
- Building dense point cloud.
- Building mesh.
- Building texture and export of orthophoto

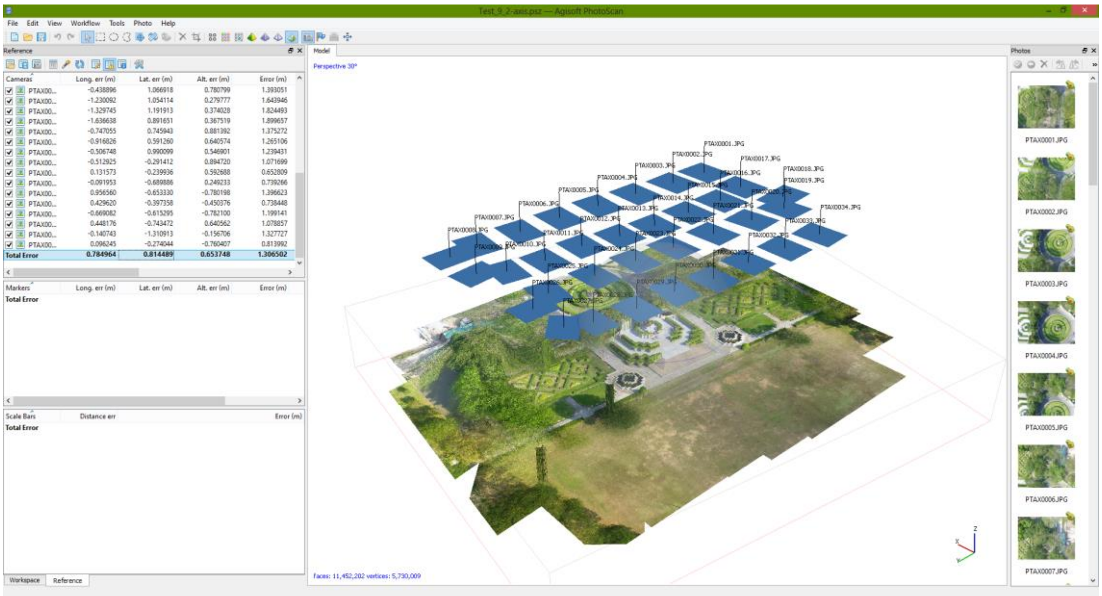


Figure 3.20 Orthophoto mosaicing in Agisoft PhotoScan software

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Results of the design and creating of gimbal and data logger

Designing and creating a prototype of 2-axis gimbal or camera controller were tested in several times in order to find the best model that can be stabilize a camera perfectly. As the result of the design in many models of gimbal frame, a better balancing of all axis will giving a better stabilization and also used low power consumption for BLDC motor to stabilize a camera. Minimizing the friction in the joints between rings using a bearing give the motion of gimbal smoothly. The developed 2-axis gimbal and data logger for UAV have shown in Figure 4.1.

The data logger to record the exterior orientation parameters of each image uses many sensors which they are separated independently. As the result of testing, time delay of camera and interval data recording are very important. If they are concordant with each other, it will give the correct recording of image and their exterior orientation parameters.

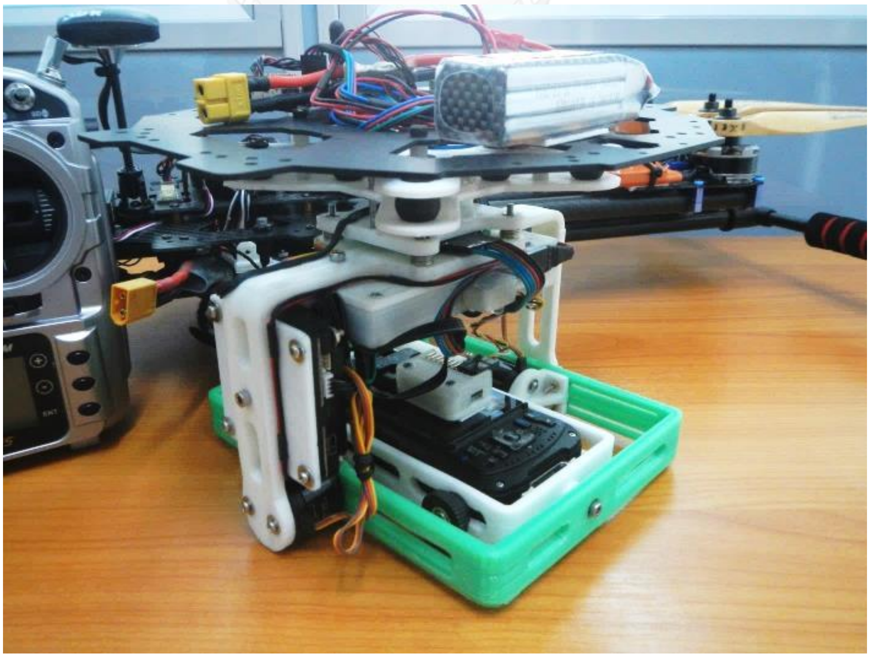


Figure 4.1 Developed 2-axis gimbal and data logger for Multicopter UAV (Drone)

4.2 Experimental results

4.2.1 Experimental result of the use of gimbal and data logger

According to the first flight for collecting data in real environment, the results from this experiments have shown that the images and exterior orientation parameters were recorded during the imaging was working perfectly and they have equal number of images and logging data show in Figure 4.2.

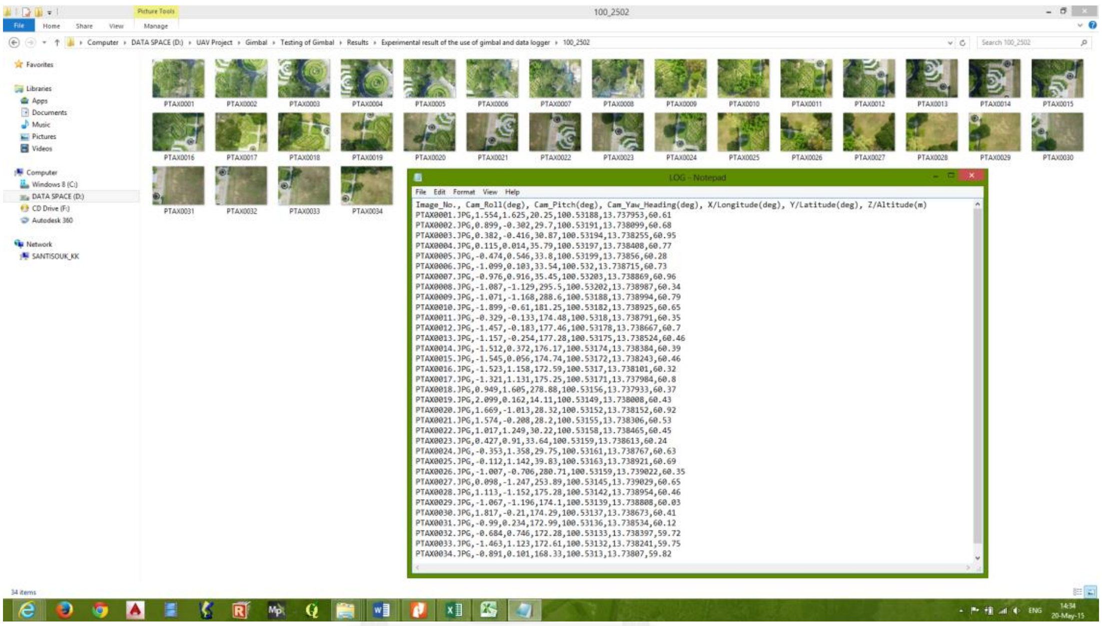


Figure 4.2 The Images and exterior orientation parameters of the first flight in the case A that get from SD card of a camera and data logger

The data that get from developed gimbal and data logger for the first case (case A) have shown a good result of the images in terms of blur and tilt. The exterior orientation parameters recorded during UAV flying through the flight path (Figure 4.3) have the orientation of images in range -1.889° to 2.099° and -1.247° to 1.625° of roll and pitch angles, respectively (see Table 4.1).

Table 4.1 The exterior orientation parameters of each image were recorded in case A

Image_No.	Cam_Roll (deg)	Cam_Pitch (deg)	Cam_Yaw_Heading (deg)	X/Longitude (deg)	Y/Latitude (deg)	Z/Altitude (m)
PTAX0001.JPG	1.554	1.625	20.250	100.53188	13.737953	60.610
PTAX0002.JPG	0.899	-0.302	29.700	100.53191	13.738099	60.680
PTAX0003.JPG	0.382	-0.416	30.870	100.53194	13.738255	60.950
PTAX0004.JPG	0.115	0.014	35.790	100.53197	13.738408	60.770
PTAX0005.JPG	-0.474	0.546	33.800	100.53199	13.738560	60.280
PTAX0006.JPG	-1.099	0.103	33.540	100.53200	13.738715	60.730
PTAX0007.JPG	-0.976	0.916	35.450	100.53203	13.738869	60.960
PTAX0008.JPG	-1.087	-1.129	295.500	100.53202	13.738987	60.340
PTAX0009.JPG	-1.071	-1.168	288.600	100.53188	13.738994	60.790
PTAX0010.JPG	-1.899	-0.610	181.250	100.53182	13.738925	60.650
PTAX0011.JPG	-0.329	-0.133	174.480	100.53180	13.738791	60.350
PTAX0012.JPG	-1.457	-0.183	177.460	100.53178	13.738667	60.700
PTAX0013.JPG	-1.157	-0.254	177.280	100.53175	13.738524	60.461
PTAX0014.JPG	-1.512	0.372	176.170	100.53174	13.738384	60.390
PTAX0015.JPG	-1.545	0.056	174.740	100.53172	13.738243	60.460
PTAX0016.JPG	-1.523	1.158	172.590	100.53170	13.738101	60.320
PTAX0017.JPG	-1.321	1.131	175.250	100.53171	13.737984	60.800
PTAX0018.JPG	0.949	1.605	278.880	100.53156	13.737933	60.370
PTAX0019.JPG	2.099	0.162	14.110	100.53149	13.738008	60.430
PTAX0020.JPG	1.669	-1.013	28.320	100.53152	13.738152	60.920
PTAX0021.JPG	1.574	-0.208	28.200	100.53155	13.738306	60.530
PTAX0022.JPG	1.017	1.249	30.220	100.53158	13.738465	60.450
PTAX0023.JPG	0.427	0.910	33.640	100.53159	13.738613	60.240
PTAX0024.JPG	-0.353	1.358	29.750	100.53161	13.738767	60.630
PTAX0025.JPG	-0.112	1.142	39.830	100.53163	13.738921	60.690
PTAX0026.JPG	-1.007	-0.706	280.710	100.53159	13.739022	60.350
PTAX0027.JPG	0.098	-1.247	253.890	100.53145	13.739029	60.650
PTAX0028.JPG	1.113	-1.152	175.280	100.53142	13.738954	60.460
PTAX0029.JPG	-1.067	-1.196	174.100	100.53139	13.738808	60.030
PTAX0030.JPG	1.817	-0.210	174.290	100.53137	13.738673	60.410
PTAX0031.JPG	-0.990	0.234	172.990	100.53136	13.738534	60.120
PTAX0032.JPG	-0.684	0.746	172.280	100.53133	13.738397	59.720
PTAX0033.JPG	-1.463	1.123	172.610	100.53132	13.738241	59.750
PTAX0034.JPG	-0.891	0.101	168.330	100.53130	13.738070	59.820

4.2.2 Experimental result of the use of data logger only

The second flight (case B) in the similar condition, the getting result was not as good as case A which images are tilted according to the inclination of UAV. Therefore, the images were tilted more than case A but the number of image and exterior orientation parameters are also equal. The exterior orientation parameters were recorded during flying through the flight path (Figure 4.4) show the orientation of images in range of -5.707° to 4.720° and -12.364° to 2.916° of roll and pitch angles, respectively (see Table 4.2).

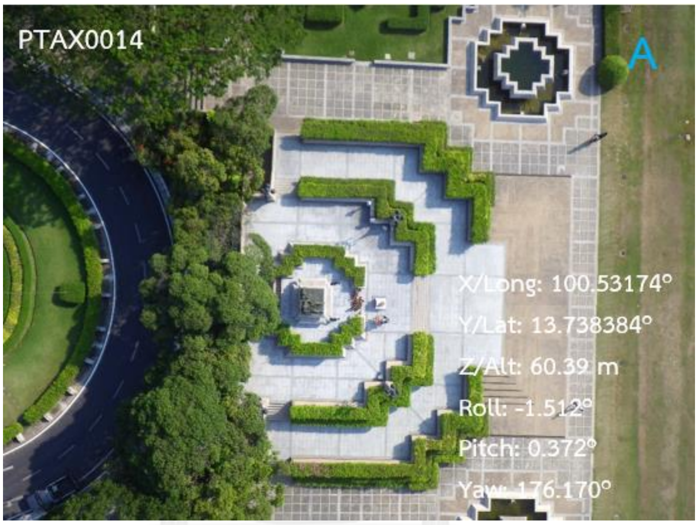


Figure 4.3 Image was taken during flying through the flight path of the case A

Figure 4.4 Image was taken during flying through the flight path of the case B

Table 4.2 The exterior orientation parameters of each image were recorded in case B

Image_No.	Cam_Roll (deg.)	Cam_Pitch (deg.)	Cam_Yaw_Heading (deg.)	X/Longitude (deg.)	Y/Latitude (deg.)	Z/Altitude (m)
PTAX0001.JPG	2.145	-1.975	32.730	100.53192	13.737997	60.380
PTAX0002.JPG	0.219	-4.748	32.130	100.53194	13.738152	60.560
PTAX0003.JPG	0.877	-3.287	31.030	100.53196	13.738316	60.770
PTAX0004.JPG	-0.026	-3.346	32.640	100.53199	13.738470	60.890
PTAX0005.JPG	-0.384	-3.528	35.600	100.53201	13.738626	60.430
PTAX0006.JPG	-0.403	-3.106	33.590	100.53202	13.738784	60.820
PTAX0007.JPG	-3.078	2.916	18.930	100.53204	13.738941	60.750
PTAX0008.JPG	-0.853	-2.15	293.20	100.53196	13.738997	60.410
PTAX0009.JPG	-4.354	-7.508	205.010	100.53185	13.738977	60.760
PTAX0010.JPG	-2.496	-7.238	179.520	100.53182	13.738843	60.380
PTAX0011.JPG	-3.719	-11.155	178.470	100.53181	13.738725	60.480
PTAX0012.JPG	-2.822	-7.200	177.980	100.53178	13.738581	59.960
PTAX0013.JPG	-1.474	-7.658	177.920	100.53176	13.738427	59.960
PTAX0014.JPG	-2.119	-6.060	176.170	100.53174	13.738266	60.100
PTAX0015.JPG	-0.772	-5.528	178.890	100.53172	13.738111	60.230
PTAX0016.JPG	2.527	-2.699	177.800	100.53169	13.737955	60.060
PTAX0017.JPG	-3.740	-5.653	280.320	100.53161	13.737933	60.640
PTAX0018.JPG	-2.042	-5.919	313.800	100.53151	13.737976	60.490
PTAX0019.JPG	-3.101	-3.703	24.810	100.53152	13.738113	60.090
PTAX0020.JPG	-3.624	-5.628	27.130	100.53154	13.738276	59.870
PTAX0021.JPG	-3.052	-3.405	23.750	100.53156	13.738440	60.220
PTAX0022.JPG	-2.010	-4.752	26.560	100.53158	13.738592	60.870
PTAX0023.JPG	-1.572	-2.948	26.900	100.5316	13.738749	60.680
PTAX0024.JPG	-2.456	-3.341	27.340	100.53162	13.738909	60.070
PTAX0025.JPG	0.396	-7.227	271.330	100.53161	13.739017	60.690
PTAX0026.JPG	-5.707	-1.858	273.330	100.53148	13.739022	60.470
PTAX0027.JPG	3.720	-9.150	180.760	100.53142	13.738952	60.060
PTAX0028.JPG	2.127	-10.273	178.040	100.53139	13.738807	60.310
PTAX0029.JPG	0.068	-10.359	174.930	100.53136	13.738658	60.370
PTAX0030.JPG	1.382	-12.364	172.150	100.53136	13.738505	60.080
PTAX0031.JPG	-1.687	-6.649	176.300	100.53134	13.738361	59.620
PTAX0032.JPG	-0.095	-10.631	169.140	100.53132	13.738219	59.980
PTAX0033.JPG	-1.575	-5.461	172.240	100.53131	13.738070	60.230
PTAX0034.JPG	4.720	-2.966	168.920	100.53128	13.737971	60.210

4.3 Comparison of the results

The results from the experiments of both cases have shown that the images and exterior orientation parameters were recorded into SD card of camera and data logger completely. The exterior orientation parameters of the both cases which get from data logger directly and get from Agisoft PhotoScan's estimation in photo alignment and orthophoto mosaicing (see Table 4.3, 4.4 and Figure 4.5) have standard deviation of error values in case A are about 0.785 m, 0.814 m, 0.654 m, 0.891°, 1.019°, and 19.670° in X, Y, Z, roll, pitch and yaw, respectively. For the case B are about 0.611 m, 0.700 m, 0.668 m, 1.801°, 1.624°, and 19.499° in X, Y, Z, roll, pitch and yaw, respectively (see Figure 4.6).

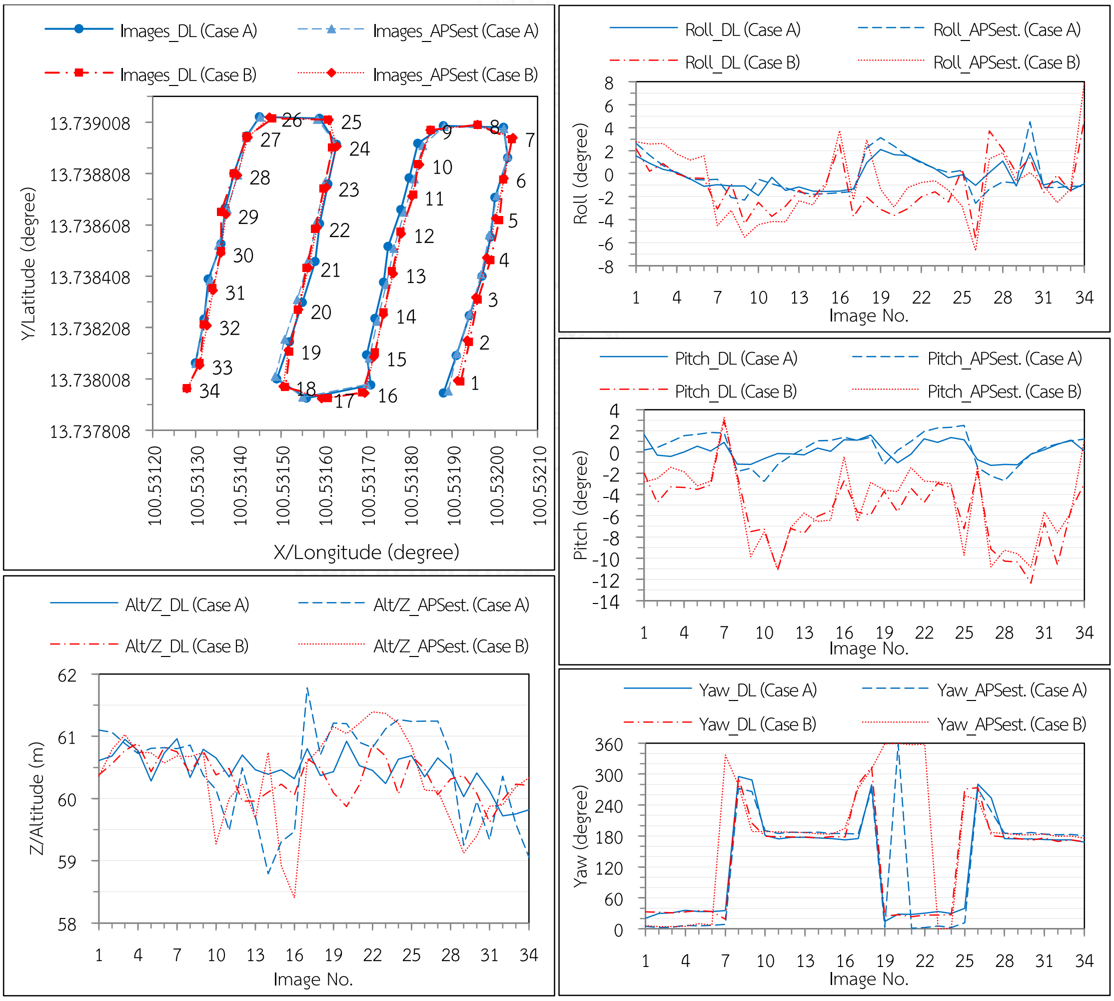


Figure 4.5 The exterior orientation parameters that get from data logger and Agisoft PhotoScan's estimation (DL - Data logger, APSest – Agisoft PhotoScan's estimation)

Table 4.3 The results of photo alignment and orthophoto mosaicing from Agisoft PhotoScan of the data in case A

Image No.	X/East (deg.)	Y/North (deg.)	Z/Altitude (m)	Yaw (deg.)	Pitch (deg.)	Roll (deg.)	Error (m)	X error (m)	Y error (m)	Z error (m)	Error (deg)	Yaw error (deg.)	Pitch error (deg.)	Roll error (deg.)	X est (deg.)	Y est (deg.)	Z est (m)	Yaw est (deg.)	Pitch est (deg.)	Roll est (deg.)
PTAX0001.JPG	100.53188	13.737953	60.610	20.250	1.625	1.554	1.597	1.239	0.881	0.490	15.254	-15.148	-1.428	1.089	100.53189	13.737961	61.100	5.102	0.197	2.643
PTAX0002.JPG	100.53191	13.738099	60.680	29.700	-0.302	0.899	0.870	0.108	0.774	0.384	27.261	-27.243	0.727	0.695	100.53191	13.738106	61.064	2.457	0.425	1.594
PTAX0003.JPG	100.53194	13.738255	60.950	30.870	-0.416	0.382	0.716	0.102	0.705	-0.068	27.690	-27.652	1.402	0.337	100.53194	13.738261	60.882	3.218	0.986	0.719
PTAX0004.JPG	100.53197	13.738408	60.770	35.790	0.014	0.115	0.752	-0.189	0.727	-0.044	29.102	-29.062	1.526	-0.074	100.53197	13.738415	60.726	6.728	1.540	0.041
PTAX0005.JPG	100.53199	13.738560	60.280	33.800	0.546	-0.474	1.214	-0.332	1.044	0.524	28.387	-28.365	1.118	-0.011	100.53199	13.738569	60.804	5.435	1.664	-0.485
PTAX0006.JPG	100.53200	13.738715	60.730	33.540	0.103	-1.099	0.915	0.571	0.710	0.087	26.687	-26.625	1.743	0.545	100.53201	13.738721	60.817	6.915	1.846	-0.554
PTAX0007.JPG	100.53203	13.738869	60.960	35.450	0.916	-0.976	0.727	-0.084	0.704	-0.157	26.619	-26.600	0.884	0.476	100.53203	13.738875	60.803	8.850	1.800	-0.500
PTAX0008.JPG	100.53202	13.738987	60.340	295.500	-1.129	-1.087	0.687	0.082	-0.446	0.516	23.033	-23.001	-0.696	-0.981	100.53202	13.738983	60.856	272.499	-1.825	-2.068
PTAX0009.JPG	100.53188	13.738994	60.790	288.600	-1.168	-1.071	0.573	0.108	-0.381	-0.414	22.683	-22.646	-0.365	-1.236	100.53188	13.738991	60.376	265.954	-1.533	-2.307
PTAX0010.JPG	100.53182	13.738925	60.650	181.250	-0.610	-1.899	1.565	1.168	-0.908	-0.509	9.507	9.155	-2.141	1.409	100.53183	13.738917	60.141	190.405	-2.751	-0.490
PTAX0011.JPG	100.53180	13.738791	60.350	174.480	-0.133	-0.329	1.608	1.269	-0.473	-0.867	10.697	10.631	-1.053	-0.542	100.53181	13.738787	59.483	185.111	-1.186	-0.871
PTAX0012.JPG	100.53178	13.738667	60.700	177.460	-0.183	-1.457	1.434	0.876	-1.116	-0.208	10.133	10.129	-0.178	0.208	100.53179	13.738657	60.492	187.589	-0.361	-1.249
PTAX0013.JPG	100.53175	13.738524	60.460	177.280	-0.254	-1.157	1.814	1.475	-0.749	-0.742	9.640	9.612	0.583	-0.434	100.53176	13.738517	59.718	186.892	0.329	-1.591
PTAX0014.JPG	100.53174	13.738384	60.390	176.170	0.372	-1.512	1.985	0.651	-0.975	-1.601	11.485	11.460	0.701	-0.279	100.53175	13.738375	58.789	187.630	1.073	-1.791
PTAX0015.JPG	100.53172	13.738243	60.460	174.740	0.056	-1.545	1.762	0.503	-1.230	-1.157	10.194	10.139	1.038	-0.188	100.53173	13.738232	59.303	184.879	1.094	-1.733
PTAX0016.JPG	100.53170	13.738101	60.320	172.590	1.158	-1.523	1.715	0.625	-1.351	-0.852	12.037	12.034	0.227	-0.126	100.53171	13.738089	59.468	184.624	1.385	-1.649
PTAX0017.JPG	100.53171	13.737984	60.800	175.250	1.131	-1.321	1.472	-1.048	0.342	0.976	8.387	8.385	-0.035	-0.216	100.53170	13.737987	61.776	183.635	1.096	-1.537
PTAX0018.JPG	100.53156	13.737933	60.370	278.880	1.605	0.949	1.106	-0.968	0.426	0.325	8.297	-8.190	-0.196	1.315	100.53155	13.737937	60.695	270.690	1.409	2.264
PTAX0019.JPG	100.53149	13.738008	60.430	14.110	0.162	2.099	1.393	-0.439	1.067	0.781	10.992	-10.861	-1.327	1.046	100.53149	13.738018	61.211	3.249	-1.165	3.145
PTAX0020.JPG	100.53152	13.738152	60.920	28.320	-1.013	1.669	1.644	-1.230	1.054	0.280	29.663	-29.631	1.162	0.743	100.53151	13.738162	61.200	358.689	0.149	2.412
PTAX0021.JPG	100.53155	13.738306	60.530	28.200	-0.208	1.574	1.824	-1.330	1.192	0.374	27.036	-27.009	1.213	-0.030	100.53154	13.738317	60.904	1.191	1.005	1.544
PTAX0022.JPG	100.53158	13.738465	60.450	30.220	1.249	1.017	1.900	-1.637	0.892	0.368	27.737	-27.729	0.681	-0.084	100.53157	13.738473	60.818	2.491	1.930	0.933
PTAX0023.JPG	100.53159	13.738613	60.240	33.640	0.910	0.427	1.375	-0.747	0.746	0.881	28.317	-28.284	1.369	0.038	100.53158	13.738620	61.121	5.356	2.279	0.465
PTAX0024.JPG	100.53161	13.738767	60.630	29.750	1.358	-0.353	1.265	-0.917	0.591	0.641	27.239	-27.217	0.985	0.446	100.53160	13.738772	61.271	2.533	2.343	0.093
PTAX0025.JPG	100.53163	13.738921	60.690	39.830	1.142	-0.112	1.239	-0.507	0.990	0.547	27.754	-27.717	1.368	0.409	100.53163	13.738930	61.237	12.113	2.510	0.297
PTAX0026.JPG	100.53159	13.739022	60.350	280.710	-0.706	-1.007	1.072	-0.513	-0.291	0.895	9.215	-9.050	-0.734	-1.577	100.53159	13.739019	61.245	271.660	-1.440	-2.584
PTAX0027.JPG	100.53145	13.739029	60.650	253.890	-1.247	0.098	0.653	0.132	-0.240	0.593	26.225	-26.166	-1.004	-1.442	100.53145	13.739027	61.243	227.724	-2.251	-1.344
PTAX0028.JPG	100.53142	13.738954	60.460	175.280	-1.152	1.113	0.739	-0.092	-0.690	0.249	10.233	9.950	-1.554	-1.813	100.53142	13.738948	60.709	185.230	-2.706	-0.700
PTAX0029.JPG	100.53139	13.738808	60.030	174.100	-1.196	-1.067	1.397	0.957	-0.653	-0.780	10.196	10.191	-0.210	0.232	100.53140	13.738802	59.250	184.291	-1.406	-0.835
PTAX0030.JPG	100.53137	13.738673	60.410	174.290	-0.210	1.817	0.738	0.430	-0.397	-0.450	12.986	12.700	0.004	2.710	100.53137	13.738669	59.960	186.990	-0.206	4.527
PTAX0031.JPG	100.53136	13.738534	60.120	172.990	0.234	-0.990	1.199	-0.669	-0.615	-0.782	10.915	10.912	0.183	-0.187	100.53135	13.738528	59.338	183.902	0.417	-1.177
PTAX0032.JPG	100.53133	13.738397	59.720	172.280	0.746	-0.684	1.079	0.448	-0.743	0.641	9.916	9.901	0.016	-0.542	100.53133	13.738390	60.361	182.181	0.762	-1.226
PTAX0033.JPG	100.53132	13.738241	59.750	172.610	1.123	-1.463	1.328	-0.141	-1.311	-0.157	10.486	10.480	-0.062	0.356	100.53132	13.738229	59.593	183.090	1.061	-1.107
PTAX0034.JPG	100.53130	13.738070	59.820	168.330	0.101	-0.891	0.814	0.096	-0.274	-0.760	11.992	11.939	1.108	-0.204	100.53130	13.738068	59.060	180.269	1.209	-1.095
Total error							1.307	0.785	0.814	0.654	19.717	19.670	1.019	0.891						

Table 4.4 The results of photo alignment and orthophoto mosaicing from Agisoft PhotoScan of the data in case B

Image No.	X/East (deg.)	Y/North (deg.)	Z/Altitude (m)	Yaw (deg.)	Pitch (deg.)	Roll (deg.)	Error (m)	X error (m)	Y error (m)	Z error (m)	Error (deg)	Yaw error (deg.)	Pitch error (deg.)	Roll error (deg.)	X est (deg.)	Y est (deg.)	Z est (m)	Yaw est (deg.)	Pitch est (deg.)	Roll est (deg.)
PTAX0001.JPG	100.53192	13.737997	60.380	32.730	-1.975	2.145	0.893	-0.726	0.519	-0.020	26.429	-26.408	-0.839	0.620	100.53191	13.738002	60.360	6.322	-2.814	2.765
PTAX0002.JPG	100.53194	13.738152	60.560	32.130	-4.748	0.219	0.964	-0.536	0.768	0.226	27.978	-27.787	2.266	2.350	100.53194	13.738159	60.786	4.343	-2.482	2.569
PTAX0003.JPG	100.53196	13.738316	60.770	31.030	-3.287	0.877	1.228	-0.574	1.055	0.258	26.711	-26.589	1.848	1.750	100.53196	13.738326	61.028	4.441	-1.439	2.627
PTAX0004.JPG	100.53199	13.738470	60.890	32.640	-3.346	-0.026	1.474	-0.972	1.098	-0.145	27.941	-27.847	1.496	1.733	100.53198	13.738480	60.745	4.793	-1.850	1.707
PTAX0005.JPG	100.53201	13.738626	60.430	35.600	-3.528	-0.384	1.284	-0.862	0.900	0.305	25.561	-25.511	0.338	1.565	100.53200	13.738634	60.735	10.089	-3.190	1.181
PTAX0006.JPG	100.53202	13.738784	60.820	33.590	-3.106	-0.403	0.676	0.082	0.622	-0.250	26.416	-26.341	0.368	1.956	100.53202	13.738790	60.570	7.249	-2.738	1.553
PTAX0007.JPG	100.53204	13.738941	60.750	18.930	2.916	-3.078	0.819	0.260	0.773	-0.065	41.827	-41.800	0.355	-1.455	100.53204	13.738948	60.685	337.130	3.271	-4.533
PTAX0008.JPG	100.53196	13.738997	60.410	293.200	-2.150	-0.853	0.318	-0.072	0.169	0.260	15.237	-15.052	-0.458	-2.325	100.53196	13.738999	60.670	278.148	-2.608	-3.178
PTAX0009.JPG	100.53185	13.738977	60.760	205.010	-7.508	-4.354	0.264	-0.251	0.079	-0.021	16.777	-16.572	-2.336	-1.170	100.53185	13.738978	60.739	188.438	-9.844	-5.524
PTAX0010.JPG	100.53182	13.738843	60.380	179.520	-7.238	-2.496	1.206	0.468	0.030	-1.110	9.146	8.928	-0.303	-1.962	100.53182	13.738843	59.270	188.448	-7.541	-4.458
PTAX0011.JPG	100.53181	13.738725	60.480	178.470	-11.155	-3.719	0.901	-0.461	-0.606	-0.483	9.476	9.464	0.169	-0.434	100.53181	13.738720	59.997	187.934	-10.986	-4.153
PTAX0012.JPG	100.53178	13.738581	59.960	177.980	-7.200	-2.822	0.853	0.135	-0.795	0.276	9.275	9.174	0.084	-1.360	100.53178	13.738574	60.236	187.154	-7.116	-4.182
PTAX0013.JPG	100.53176	13.738427	59.960	177.920	-7.658	-1.474	1.018	0.256	-0.945	-0.280	9.148	8.901	1.916	-0.888	100.53176	13.738418	59.680	186.821	-5.742	-2.362
PTAX0014.JPG	100.53174	13.738266	60.100	176.170	-6.060	-2.119	0.942	-0.097	-0.686	0.639	8.145	8.108	-0.485	-0.596	100.53174	13.738260	60.739	184.278	-6.545	-2.715
PTAX0015.JPG	100.53172	13.738111	60.230	178.890	-5.528	-0.772	2.284	-0.310	-1.844	-1.311	4.764	4.677	-0.878	-0.221	100.53172	13.738094	58.919	183.567	-6.406	-0.993
PTAX0016.JPG	100.53169	13.737955	60.060	177.800	-2.699	2.527	1.793	0.658	-0.194	-1.657	17.483	17.293	2.265	1.218	100.53170	13.737953	58.403	195.093	-0.434	3.745
PTAX0017.JPG	100.53161	13.737933	60.640	280.320	-5.653	-3.740	1.609	-1.602	-0.140	-0.048	7.940	-7.753	-0.869	1.475	100.53160	13.737932	60.592	272.567	-6.522	-2.265
PTAX0018.JPG	100.53151	13.737976	60.490	313.800	-5.919	-2.042	0.716	-0.570	0.284	0.326	6.847	-3.578	3.051	4.978	100.53151	13.737979	60.816	310.222	-2.868	2.936
PTAX0019.JPG	100.53152	13.738113	60.090	24.810	-3.703	-3.101	1.346	-0.286	0.770	1.066	26.136	-26.078	0.133	1.737	100.53152	13.738120	61.156	358.732	-3.570	-1.364
PTAX0020.JPG	100.53154	13.738276	59.870	27.130	-5.628	-3.624	1.199	0.148	0.191	1.174	28.043	-27.971	1.884	0.711	100.53154	13.738278	61.044	359.159	-3.744	-2.913
PTAX0021.JPG	100.53156	13.738440	60.220	23.750	-3.405	-3.052	1.110	0.476	0.175	0.987	27.019	-26.886	1.951	1.835	100.53156	13.738442	61.207	356.864	-1.454	-1.217
PTAX0022.JPG	100.53158	13.738592	60.870	26.560	-4.752	-2.010	0.847	0.414	0.525	0.520	28.432	-28.333	2.029	1.237	100.53158	13.738597	61.390	358.227	-2.723	-0.773
PTAX0023.JPG	100.53160	13.738749	60.680	26.900	-2.948	-1.572	1.158	0.622	0.694	0.688	26.560	-26.543	0.123	0.939	100.53161	13.738755	61.368	0.357	-2.825	-0.633
PTAX0024.JPG	100.53162	13.738909	60.070	27.340	-3.341	-2.456	1.731	1.189	0.541	1.136	25.955	-25.933	0.351	1.017	100.53163	13.738914	61.206	1.407	-2.990	-1.439
PTAX0025.JPG	100.53161	13.739017	60.690	271.330	-7.227	0.396	0.334	0.262	0.150	0.143	13.717	-13.093	-2.494	-3.239	100.53161	13.739018	60.833	258.237	-9.721	-2.843
PTAX0026.JPG	100.53148	13.739022	60.470	273.330	-1.858	-5.707	1.001	-0.749	0.573	-0.338	23.549	-23.522	0.542	-0.994	100.53147	13.739027	60.132	249.808	-1.316	-6.701
PTAX0027.JPG	100.53142	13.738952	60.060	180.760	-9.150	3.720	0.432	0.312	-0.292	0.063	6.596	5.899	-1.661	-2.439	100.53142	13.738949	60.123	186.659	-10.811	1.281
PTAX0028.JPG	100.53139	13.738807	60.310	178.040	-10.273	2.127	1.197	0.819	-0.577	-0.654	7.132	7.050	1.039	-0.292	100.53140	13.738802	59.656	185.090	-9.234	1.835
PTAX0029.JPG	100.53136	13.738658	60.370	174.930	-10.359	0.068	2.073	1.226	-1.116	-1.244	7.322	7.244	0.777	-0.727	100.53137	13.738648	59.126	182.174	-9.582	-0.659
PTAX0030.JPG	100.53136	13.738505	60.080	172.150	-12.364	1.382	0.920	-0.061	-0.612	-0.684	10.393	10.198	1.531	-1.292	100.53136	13.738499	59.396	182.348	-10.833	0.090
PTAX0031.JPG	100.53134	13.738361	59.620	176.300	-6.649	-1.687	0.973	0.161	-0.925	0.258	7.542	7.444	1.006	0.677	100.53134	13.738353	59.878	183.744	-5.643	-1.010
PTAX0032.JPG	100.53132	13.738219	59.980	169.140	-10.631	-0.095	0.791	0.721	-0.315	-0.086	10.688	9.957	3.022	-2.439	100.53133	13.738216	59.894	179.097	-7.609	-2.534
PTAX0033.JPG	100.53131	13.738070	60.230	172.240	-5.461	-1.575	0.839	-0.029	-0.837	-0.048	7.906	7.898	-0.252	0.260	100.53131	13.738062	60.182	180.138	-5.713	-1.315
PTAX0034.JPG	100.53128	13.737971	60.210	168.920	-2.966	4.720	0.135	-0.052	-0.032	0.121	8.369	6.477	4.125	3.327	100.53128	13.737971	60.331	175.397	1.159	8.047
Total error							1.144	0.611	0.700	0.668	19.649	19.499	1.624	1.801						

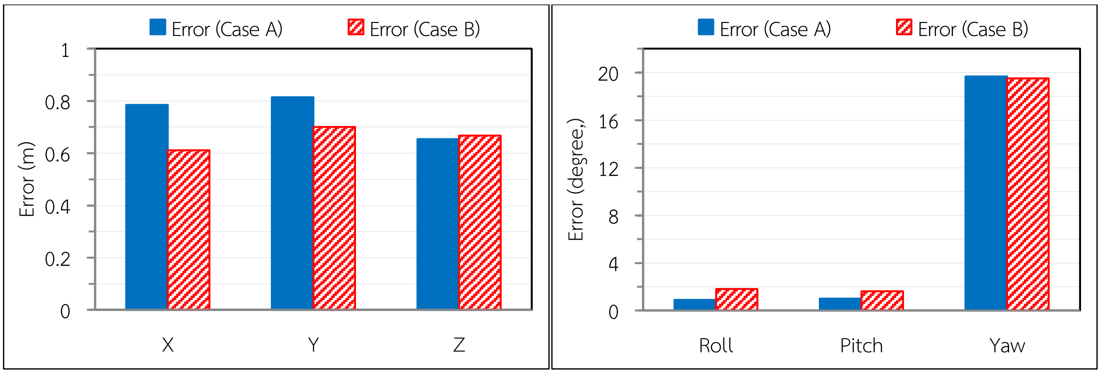


Figure 4.6 Standard deviation of error values of exterior orientation parameters from the both cases

After the orthophoto mosaicing from aerial images, the results have shown that the orthophoto that get from the case A has a better overlapping area than the case B when look at the overview of their orthophoto. In addition, when zoom in 200 percent, an orthophoto of case A has a better objects detail than case B in terms of image qualities which can see clearly from a Figure 4.7 and 4.8, respectively.

Figure 4.7 Orthophoto of the case A that get from orthophoto mosaicing

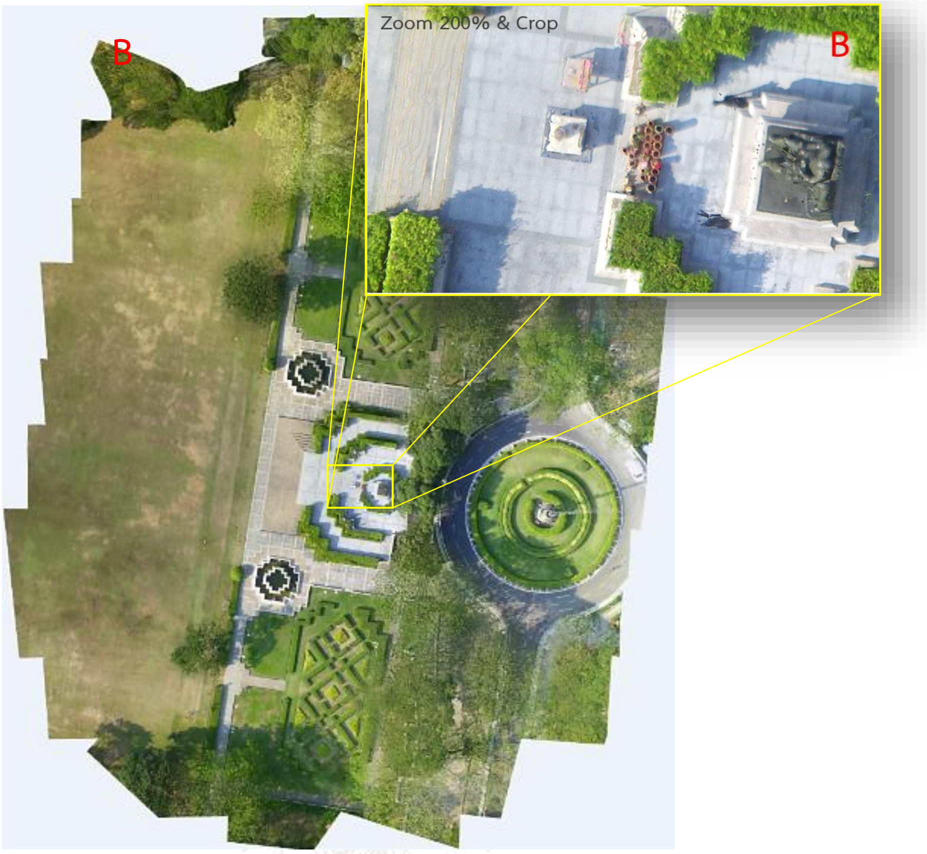


Figure 4.8 Orthophoto of the case B that get from orthophoto mosaicing

CHAPTER 5
CONCLUSION

5.1 Conclusion of experimental results

As the results of testing the developed 2-axis brushless gimbal and data logger, the images controlled by the gimbal have a better orientation than images which have not use gimbal. The quality of image mosaic from images that controlled by the gimbal is also better.

The standard deviation of error for X, Y, and Z occurred from both cases are equivalent and less than a meter. Moreover, the standard deviation of error for roll and pitch are less than 2 degrees which less than 3 degrees that firstly determined in the testing. Therefore, the exterior orientation parameters that get from data logger are acceptable range.

In addition, small standard deviation of their error values making the exterior orientation parameters that get from developed data logger had enough precision and reliability, except yaw (or heading) which has more error than others. An error value of yaw may be influenced by magnetic fields within the UAV due to electronic equipment and other factors.

The developed brushless gimbal and data logger have shown a great potential of vertical imaging, precision of data, and providing nearly vertical images and exterior orientation parameters for aerial mapping. From orthophoto mosaicking result, the better image orientation can be generates better orthophoto.

5.2 Issues and obstacles in research

1) Design and creating of gimbal frame, the trial and error cannot be avoid due to this is fundamental of prototyping. The process is typically time consuming.

2) According to tuning of PID and PWM of the gimbal controller in the laboratory which seen that it works perfectly for stabilizing camera, calibrating gimbal in term of tuning PID and PWM consume much times. Since environment that use to test the vibration of gimbal in laboratory is different from real flying in the field.

3) If the speed of their recording are not related with the imaging, it may cause the acquired images and exterior orientation parameters they have not equal number. Which can cause error, if the data have been used in the image processing without consideration of this problem.

4) IR transmitter cannot work in the strong sunlight.

5.3 Suggestions

1) To reduce the inclination of the imaging on UAV, the gimbal controller should be perfectly tuned and tested in various conditions that similar with the conditions occurred on UAV as much as possible.

2) To reduce the error that may occurred in data logging during the imaging, the time setting of interval recording should be set lower than the default self-timer in a camera. IR transmitter should be placed closely to the IR receiver of a digital camera to ensure the camera shutter will absolutely work.

3) As the result of orthophoto mosaicing, yaw (heading) angle of the recorded images had large error value. To reduce these error, should be added the third axis (Z-axis/yaw) or third motor into the next prototype of the gimbal in order to control a direction (yaw) of the imaging which it might reduce error of image direction and get better image mosaic result.

4) In this research, the orthophoto maps that get from the developed 2-axis brushless gimbal and data logger have not been evaluated for an accuracy of the position. Therefore this research will be focused in term of quality of data that get from the developed tools and image mosaic.

REFERENCES

3DRobotics. (2007). "3DRobotics, ArduPilot/APM development." Retrieved 9 September, 2014, from http://dev.ardupilot.com/.

Aleksander, N., J. e. Karol and B. Damian (2013). "The Prototype of Gyro-Stabilized UAV Gimbal." Springer-Verlag Berlin Heidelberg 107-115.

Aleksey, M. (2012). "Simple brushless gimbal controller." Retrieved 31 August, 2014, from http://www.simplebgc.com/eng/about/.

Dale, D. (2002). "NMEA data." Retrieved 2 September 2014, from http://www.gpsinformation.org/dale/nmea.htm#nmea.

Elnima, E. E. (2013). "A solution for exterior and relative orientation in photogrammetry, a genetic evolution approach." Journal of King Saud University - Engineering Sciences **27**(1): 108-113.

Henri, E. (2009). UAV_Photogrammetry. Doctor of Sciences Dipl.-Ing., University of Technology Dresden

InvenSense. (2011). "InvenSense, MPU-9150 Product Specification Revision 4.0." Retrieved 12 September 2014, from http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/IMU/PS-MPU-9150A.pdf.

James, S. A., M. Irene and B. R. Johannes (2010). Small-Format Aerial Photography Principles, techniques and geoscience applications by (Elsevier Science, 2010), Elsevier.

Jeremy, B. (2013). Exploring Arduino Tools and Techniques for Engineering Wizardry. 10475 Crosspoint Boulevard, Indianapolis, IN 46256 John Wiley & Sons, Inc.

Lee, D.-J., I. Kaminer, V. Dobrokhodov and K. Jones (2010). "Autonomous feature following for visual surveillance using a small unmanned aerial vehicle with gimbaled camera system." International Journal of Control, Automation and Systems **8**(5): 957-966.

Lee, D. w., J.-h. Kim, K.-r. Cho, S.-y. Jo, J.-h. Kim, C.-o. Min, D.-i. Han and S.-j. Cho (2010). "Development of an electro-optical system for small UAV." Aerospace Science and Technology **14**(7): 505-511.

Legat, K. (2006). "Approximate direct georeferencing in national coordinates." ISPRS Journal of Photogrammetry and Remote Sensing **60**(4): 239-255.

Martin, E., N. Joshua and H. Jordan (2013). Arduino in Action, MANNING.

Michael, S. (2007). "IMPROVEMENT OF INTERIOR AND EXTERIOR ORIENTATION OF THE THREE LINE CAMERA HRSC WITH A SIMULTANEOUS ADJUSTMENT." ISPRS Journal of Photogrammetry and Remote Sensing: 161-166.

Nielsen, M. Ø. (2004). True Orthophoto Generation. Master, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark.

Qadir, A., W. Semke and J. Neubert (2013). "Vision Based Neuro-Fuzzy Controller for a Two Axes Gimbal System with Small UAV." Journal of Intelligent & Robotic Systems **74**(3-4): 1029-1047.

Sebastian, S. (2011). "Multi Camera IR Control." Retrieved 29 August 2014, from http://sebastian.setz.name/arduino/my-libraries/multi-camera-ir-control/.

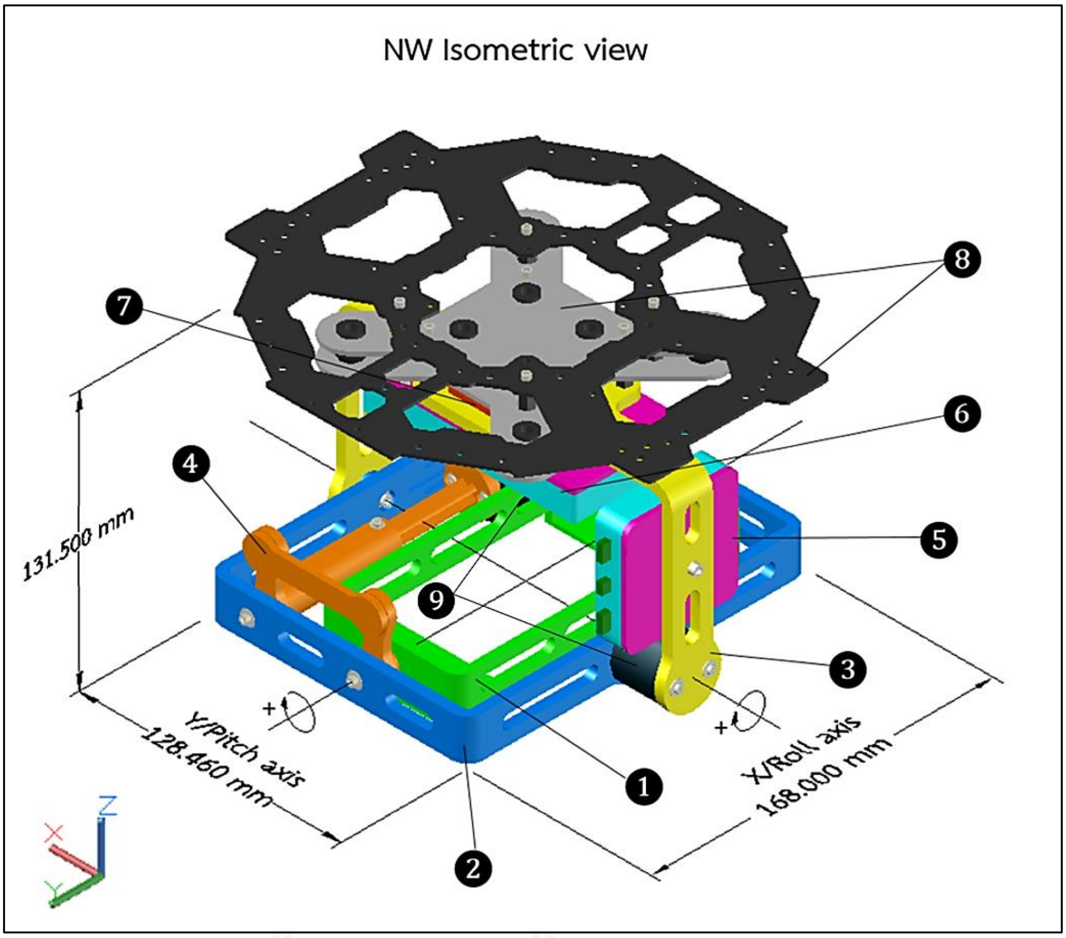
Siebert, S. and J. Teizer (2014). "Mobile 3D mapping for surveying earthwork projects using an Unmanned Aerial Vehicle (UAV) system." Automation in Construction **41**: 1-14.

Xiang, H. and L. Tian (2011). "Method for automatic georeferencing aerial remote sensing (RS) images from an unmanned aerial vehicle (UAV) platform." Biosystems Engineering **108**(2): 104-113.

APPENDIX

Appendix A

Drawing of developed 2-axis brushless gimbal frame



<u>Legend:</u>

❶ Ring 1 (Pitch frame)

❷ Ring 2 (Roll frame)

❸ Ring 3 (Yaw frame)

❹ Steering arm

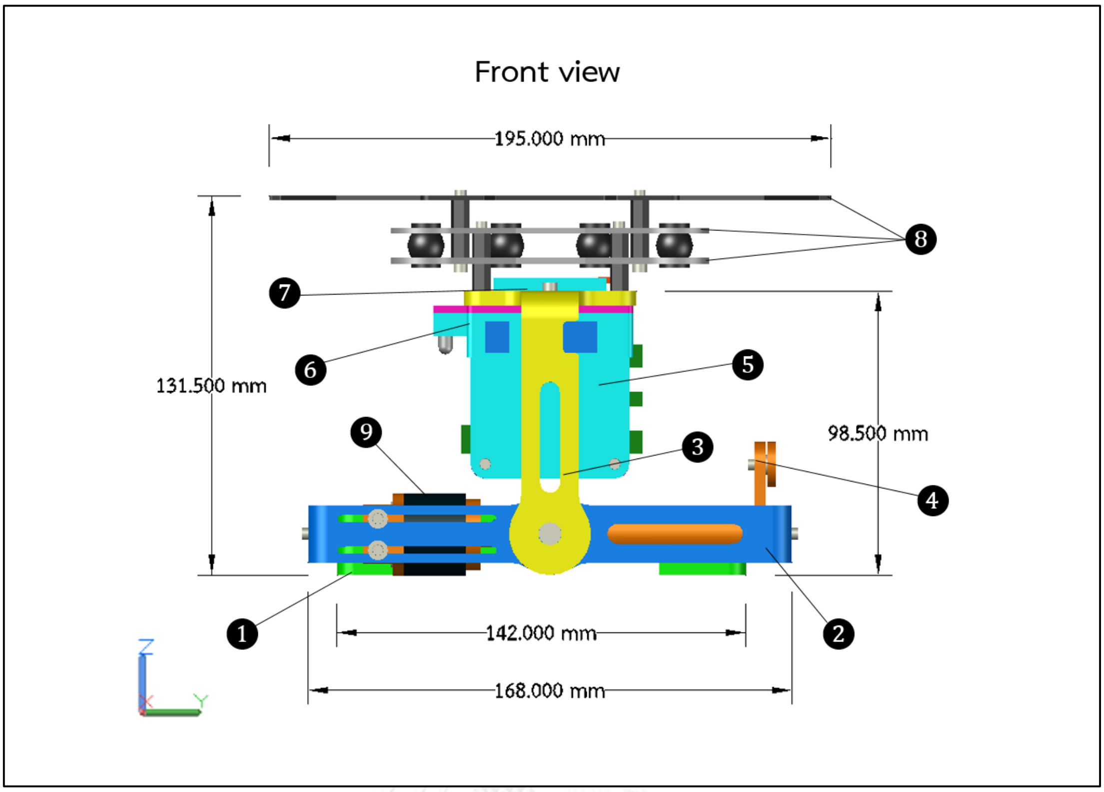
❺ Case of RCTIMER 2-axis brushless gimbal controller board

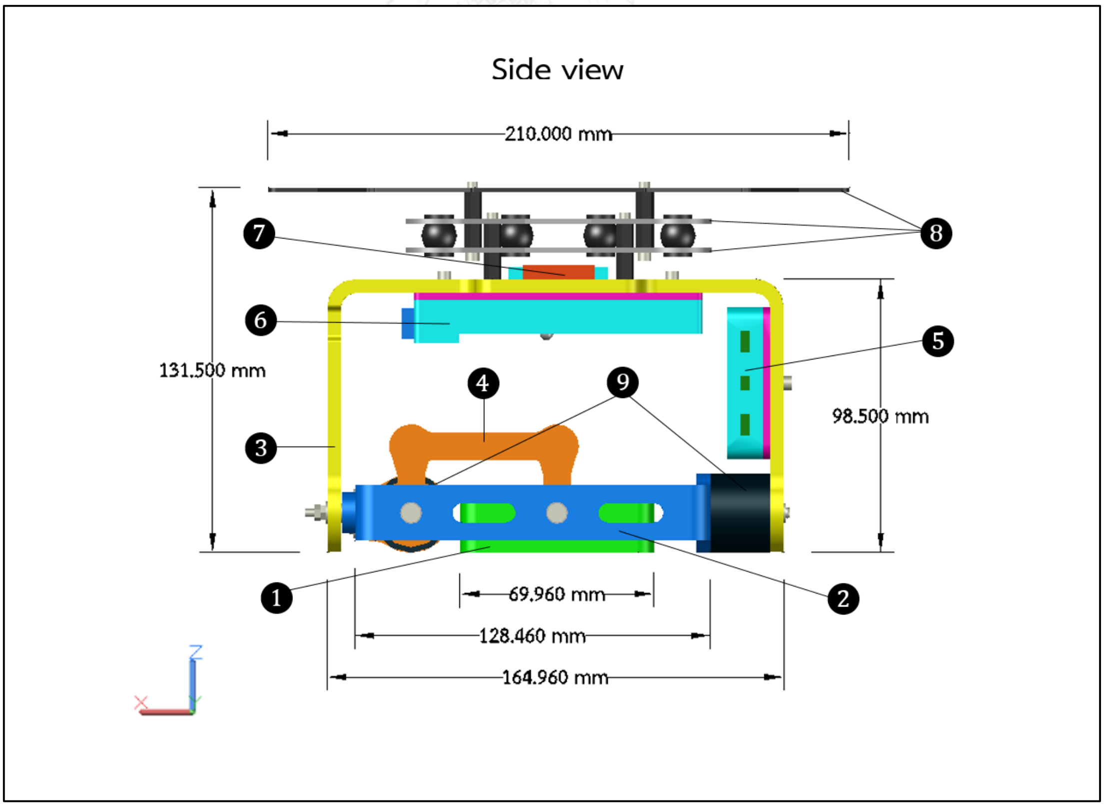
❻ Case of Arduino MEGA 2560 and IR transmitter

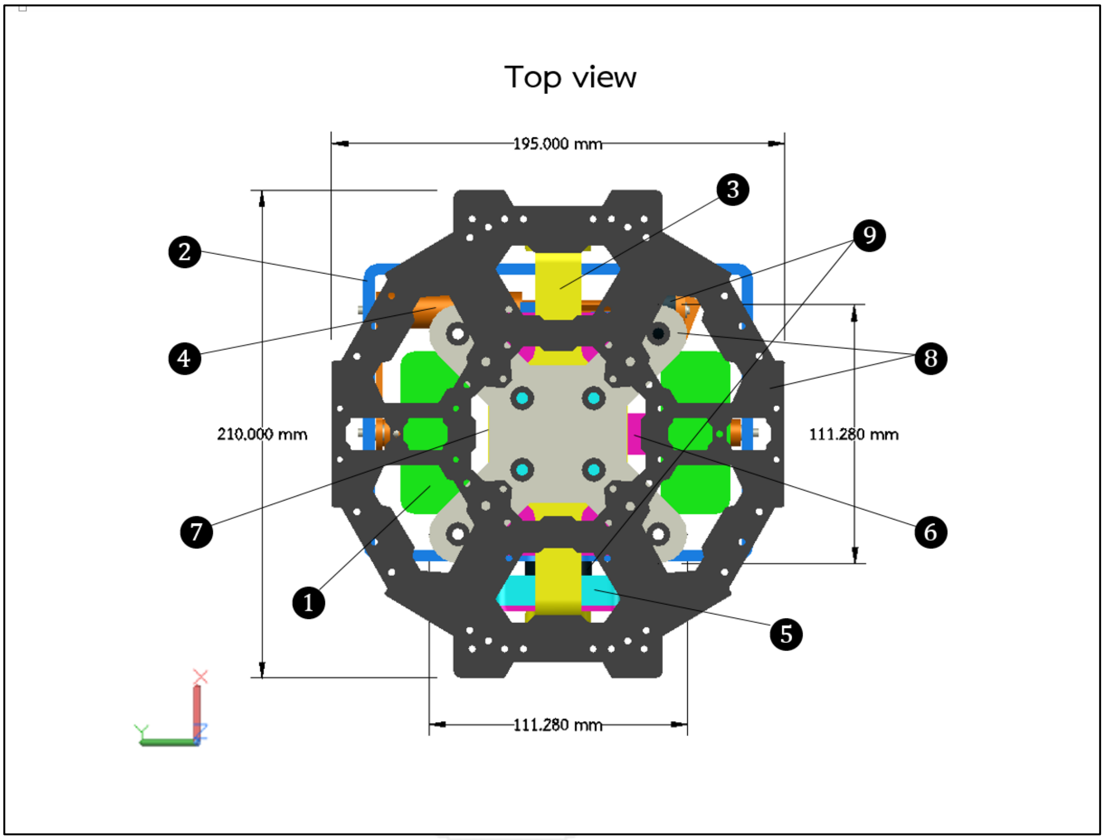
❼ Case of SD-MMC card

❽ Anti-vibration

❾ Brushless DC motor


Front view
195.000 mm
131.500 mm
98.500 mm
142.000 mm
168.000 mm
1
2
3
4
5
6
7
8
9
Z
Y

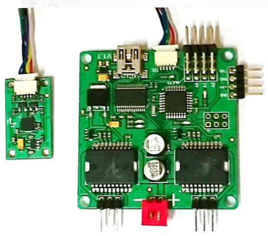

Side view
210.000 mm
131.500 mm
98.500 mm
69.960 mm
128.460 mm
164.960 mm
1
2
3
4
5
6
7
8
9
Z
X


Top view
195.000 mm
210.000 mm
111.280 mm
111.280 mm
1
2
3
4
5
6
7
8
9

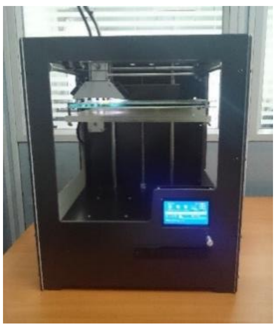
Appendix B
List of the hardware and their specifications

Item No.	Figure	Specifications
1). ArduPilot controller board (APM 2.5)		- Support ArduPlane, Multicopter, ArduCopter, helicopter, ArduRover - Arduino Compatible - Comes presoldered (with straight pins) and tested - Include 3axis gyro, accelerometer and magnetometer, along with a highperformance barometer - Onboard 4 MP Data flash chip for automatic data logging - Digital compass powered by Honeywell's HMC5883LTR chip, now included on the main board - Optional offboard GPS, uBlox NEO6M GPS module - One of the first open source autopilot system to use Invensense's 6 DoF Accelerometer/Gyro MPU6000 - Barometric pressure sensor upgraded to MS561101BA03, from Measurement Specialties - Atmel's ATMEGA2560 and ATMEGA32U2 chips for

		processing and USB functions respectively - Dimension: 70.5x45x13.5mm - Weight: 31g
2). 3DR UBlox GPS + Compass Module		- ublox LEA-6H module - 5 Hz update rate - 25 x 25 x 4 mm ceramic patch antenna - LNA and SAW filter - Rechargeable 3V lithium backup battery - Low noise 3.3V regulator - I2C EEPROM for configuration storage - Power and fix indicator LEDs - Protective case - ArduPilot Mega compatible 6-pin JST connector - Exposed RX, TX, 5V and GND pad - Dimension: 38x38x8.5mm - Weight: 16.8g
3). Compact digital camera (Pentax Optio WG-2)		Camera: Water proof, shock proof, cold proof, crush proof digital compact Sensor: 1/2.3” CMOS Effective Pixels: Approx. 16.0megapixels Focal Length: 5 - 25mm, approx. 28-140mm in 35mm

		Interface: PC / AV terminal (PAL/NTSC), USB 2.0 (Hi-Speed), HDMI terminal (Type D) Micro port Dimension: 122.5x61.5x29.5mm Weight: 173g (without battery and memory card) 194g (loaded and ready)
4). Brushless DC motor for gimbal (EMAX GB 2210)		- KV: 110 - Framework: 12N14P - hole distance: 16mm - hole diameter: 2.5mm - Dimension: 28.5X28.5X21.5mm - Li-Poly Battery (cell): 3 cells Li-poly - Weight: 48g
5). Gimbal controller board and IMU (RCTIMER 2-axis brushless gimbal controller & IMU v1.0).		- Microcontrolle ATMega328P, onboard FT232RL for debug - Li-Poly Battery (cell): 3-6 cells Li-poly - Port: Multiport, UART port - IMU: MPU-6050 6-axis gyro/accel Dimension: 50x50mm Weight: 25g (include IMU board)

6). Arduino controller board (Arduino Mega 2560)		- Microcontroller: ATMega2560 - Operating Voltage: 5V - Input Voltage (recommended): 7-12V - Input Voltage (limits): 6-20V - Digital I/O Pins: 54 (of which 14 provide PWM output) - Analog Input Pins: 16 - DC Current per I/O: Pin: 40mA - DC Current for 3.3V Pin: 50mA - Flash Memory: 256 KB of which 8 KB used by bootloader - SRAM: 8KB - EEPROM: 4KB - Clock Speed: 16MHz - Dimension: 101.98x53.63x 15.29mm - Weight: 34.9g
7). Breakout board IMU (MPU-9150).		- Digital-output 9-axis MotionFusion data in rotation matrix, quaternion, Euler Angle, or raw data format

		- Tri-Axis angular rate sensor (gyro) with a sensitivity up to 131 LSBs/dps and a full-scale range of ±250, ±500, ±1000, and ±2000dps - Tri-Axis accelerometer with a programmable full scale range of ±2g, ±4g, ±8g and ±16g - Tri-axis compass with a full scale range of ±1200μT - VDD Supply voltage range of 2.4V-3.46V; VLOGIC of 1.8V±5% or VDD - Dimension: 28x15.5x2.5mm
8). Breakout board for SD-MMC and SD card		- Standard socket for SD/MMC .1" 10-pin header. - Dimension: 38x33x4.5mm
9). Infrared (IR) transmitter		- DIY Infrared transmitter for camera shutter control, which comprises of infrared (IR) LED and resistor
10). 3D printer		- Build Volume: 20x20x20cm - Speed: 200mm/s - Extruder: Single - Nozzle diameter: 0.4mm - Material: ABS, PLA (1.75mm) - Bed material: Glass (5mm)

		- Heated bed: PCB - Printing: Print from SD card

Appendix C
Programing of controller

C.1. Brushless gimbal controllers

C.1.1. Connection of controller

The RCTIMER V1 2-axis brushless gimbal controller board was used in this research is based on controlling of Arduino programing. It has a similar components of the Martinez V3 2-axis brushless gimbal controller board including the FTDI USB controller chip. In, addition, this board use the FTDI 2.8.24 windows driver to communicate with a computer which can download from this link (http://www.itsqv.com/QVW/files/CDM-2.08.24-WHQL-Certified.zip). RCTIMER V1 board is a well-designed copy of the original Martinez V3 board and comes completely pre-soldered with all the pin-outs needed. Moreover, it still use the same firmware of the Martinez V3 board for tuning of PWM and PID parameters in order to control a BLDC motor.

The firmware of Martinez V3 brushless gimbal controller board can download from this website (http://sourceforge.net/projects/brushless-gimbal-brugi/files) e.g. BruGi_050_r217. For uploading a firmware into the RCTIMER V1 board, the open source Arduino software has been used. Which it can download from this website (http://www.arduino.cc/en/Main/Software).

Furthermore, for the uploading steps of brushless gimbal controller's firmware are listed as follows:

1) Go to File/Preferences and select the folder of new firmware files are located and click “OK”.

2) Go to Tools/Board Type and select the “Arduino Mini w/ATmega328”.

3) Go to Tools/Programmer and select the “USPasp”.

4) Connect the IMU (MPU-6050) to brushless gimbal controller board.

5) Connect the USB to brushless gimbal controller board and computer.

6) Start Arduino and select the COM port assigned to brushless gimbal controller board (from Device Manager in windows).

7) If brushless gimbal controller board has an older firmware should be cleared the eeprom before upload a new firmware by going to File/Examples/EEPROM/eeprom_clear (Verify and Upload).

8) In Arduino, go to File/Open and select the *.INO file from firmware package, e.g. _BruGi.INO.

9) Define the MPU-6050 by going to "definitions.h" tab, on the line as below and then change HIGH to LOW.

```
#define MPU6050_DEFAULT_ADDRESS MPU6050_ADDRESS_AD0_HIGH
```

10) Click on "Verify" (Check mark button) and when it completes click "Upload" (Arrow button) and wait until finished (see Figure C.1).

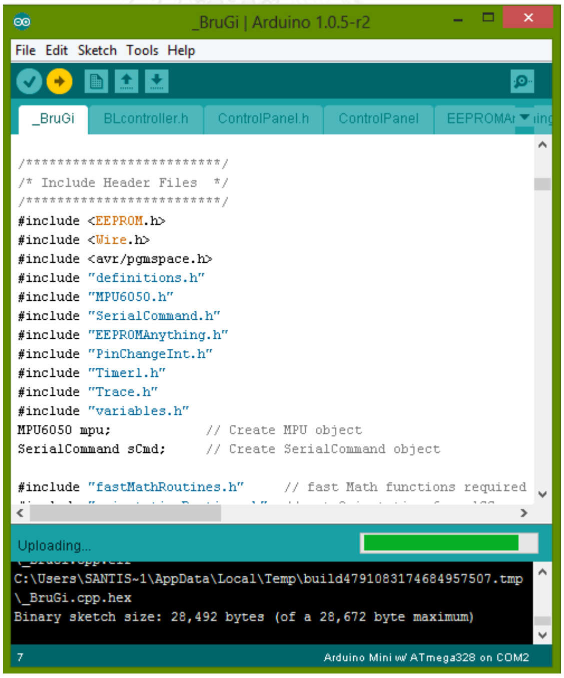


Figure C.1 The uploading of brushless gimbal controller's firmware by using Arduino software

C.1.2. Controller tuning

A tuning of brushless gimbal controller is comprised of two important parameters have to tune which they are PWM and PID parameters. For a tuning of their parameters in order to control the rotational direction of BLDC motor will be used brushless gimbal (BruGi) software that comes with BruGi firmware. Furthermore, before tune these parameters should firstly check as follows:

1) Board Power.

The RCTIMER V1 board can be inputted with a power of Li-Po battery of 2 to 6 cells. For the best result, should be used a separate battery and adjusted the voltage regulator at least 9 volts.

2) Gimbal Motion.

It is critical that all movement of the gimbal components are completely unrestrained. That is, perfectly smooth with no resistance.

3) Balance.

A gimbal with camera installed and all cables connected must be balanced near perfectly. That is, without any power applied, if camera was moving to any position, but it stays there and does not fall over (side to side or front to back). It may move slightly due to magnetic motor "cogging" and this is acceptable.

4) Cable Routing.

All motor, IMU and/or video cable routing must be free from applying any tension (pulling) on the gimbal across the full range of desired motion. It takes very little cable resistance to throw the gimbal balance off and cause bad behavior. Never route IMU cables close to motor wires or any other source of EMI as this can cause erratic behavior and errors.

5) IMU Mount Method.

The mounting method of the IMU board is very important. It cannot be loose or free to vibrate as this will play damage with the sensors. The recommendation is to mount it with machine screws solidly to the top or bottom camera plate.

6) IMU Mounting Orientation.

In General, IMU boards have the mark of positive ‘X’ and ‘Y’ direction arrows printed on the board. If these arrows are not printed on the board, use the following directional map (see Figure C.2) to orient the board. The chip has a dimple on one corner which is used to set these directions.

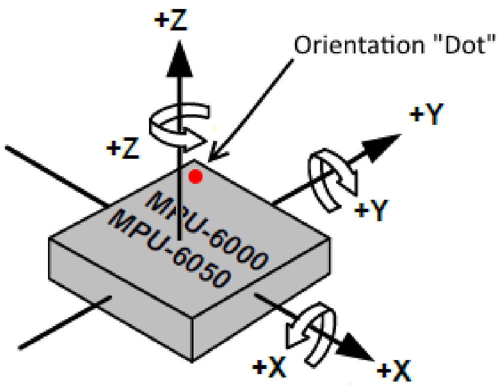


Figure C.2 The orientation map of IMU

For the setting of IMU orientation in BruGi software is shown in Table C.1 With the default Martinez/BruGi software settings, the IMU board (the really small one) is mounted such that the 'X' arrow shown on the board component side is pointed forward and the 'Y' arrow points to the right side. This means the components will be on the downside of the board. Which can swap the X-Y orientation or the Z/-Z orientation in the GUI if would like to mount it differently (but still horizontally).

Table C.1 The setting of IMU orientation on brushless gimbal (BruGi) software

Axis	Orientation mode of IMU			
	Standard (direction)	Swap X-Y (direction)	Swap Z only (direction)	Swap X-Y-Z (direction)
X	Forward	Right	Forward	Left
Y	Right	Forward	Left (check reverse motor in Pitch tab)	Forward
Z	Down	Down	Up	Up

7) Drivers of Brushless gimbal controller.

All necessary drivers must be installed and functioning correctly. On Windows PC's, the 2.8.24 driver is the most common working driver.

For the tuning steps of PWM and PID parameters on BruGi software has listed as below, which these parameters may be differences in the values and/or percentages (see Figure C.3).

1) For Live tuning, that is with motors connected, always follow this sequence:

a) With camera installed and balanced, connect battery power to the brushless gimbal controller board.

b) Connect the USB to brushless gimbal controller board and computer.

c) Start the BruGi software and click connect.

2) Set zero of max PWM and PID values in both Pitch and Roll.

3) Beginning with Roll, start with a value of 5 on P, I & D (on v048, start with P=5, I=0.1, & D=5)

4) Raise the PWM value until the Roll motor begins to vibrate then back-off enough to completely stop the vibration

5) Raise the ‘P’ value until the motor just begins to vibrate.

6) Raise the ‘D’ value until the motor becomes quiet.

7) Continue going back and forth between ‘P’ and ‘D’ until ‘P’ value can reach to the highest value.

8) Add the values for ‘P’ and ‘D’ together and divide by 2 to get a starting value for ‘I’ (049 and up firmware, for 048, change ‘I’ to 0.5).

9) Give the gimbal a light tap on one side while watching the GUI trace to make sure it immediately recovers.

a) If it starts to hum or takes a long time to settle down, ‘I’ is too high, reduce by 10%.

b) If it returns immediately turn ‘I’ up by 10% and recheck. The goal is the highest value of ‘I’ without causing it to vibrate after a "tap"

10) Roll Tuning should be complete calibrate the Gyro and save these settings to brushless gimbal controller board. This is very important if a firmware is r161 or lower follow these 2 steps:

a) Click on the options tab and then click "Save to Board"

b) Click on options tab and then click "Save to Flash".

If use a newer firmware than r161, simply click on options and "Save to Flash".

11) Now, go to Pitch settings and repeat steps 3 through 10

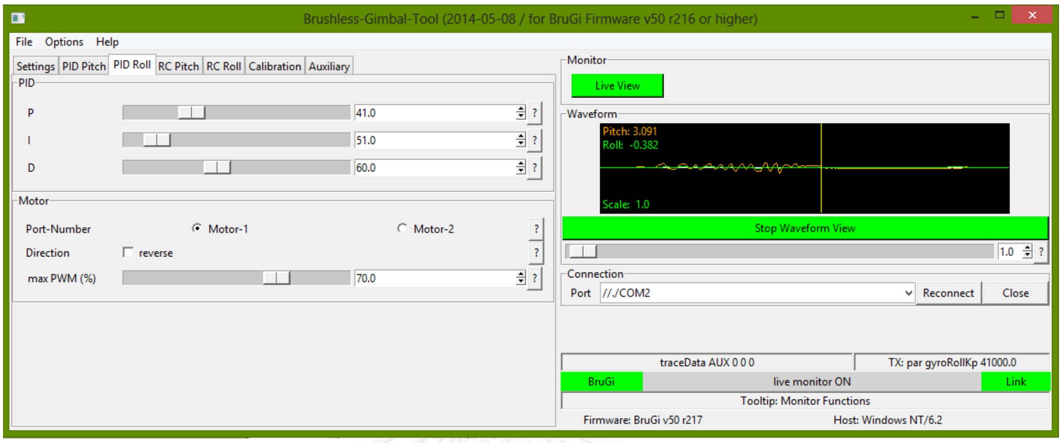


Figure C.3 The tuning of brushless gimbal controller

C.2. Data logger

C.2.1. Sending of data

The steps are used for sending of data will be described as follows:

1) Install the Mission planer software, which can download from this link (http://ardupilot.com/downloads/?did=82)

2) Install the driver of APM 2.5 controller board by connecting to the computer via micro USB cable. Then the driver will be installed automatically.

3) Download ArduCopter firmware for APM 2.5 controller board from this link (https://github.com/diydrones/ardupilot) and then select one of the ArduCopter firmware version that is available to download e.g. ArduCopter-2.9

4) Go to File/Preferences and select the folder of ArduCopter firmware files are located and click “OK”.

5) Go to Tools/Board Type and select the “Arduino Mega 2560 or Mega ADK”.

6) Go to Tools/Programmer and select the “USPasp”.

7) Solder the serial pins of APM 2.5 board such as, TX2, RX2, 5V and GND

8) Connect the 3DR UBLOX GPS + compass module to APM 2.5 controller board by using 5 pins wire.

9) Connect the USB to APM 2.5 controller board and computer.

10) Start Arduino and select the COM port assigned to AMP 2.5 controller board (from Device Manager in windows).

11) If APM 2.5 controller board has an older firmware should be cleared the eeprom before upload a new ArduCopter firmware by going to File/Examples/EEPROM/eeprom_clear (Verify and Upload).

12) In Arduino, go to File/Open and select the *.INO file from firmware package, e.g. ArduCopter.INO.

13) Go to "ArduCopter" tab for modification of ArduCopter firmware in order to send the required data to serial port on APM 2.5 controller board. In this tab should be edited the following:

Note: this method supports the ArduCopter 2.9x and lower.

a) Add new line of code for defining of a serial port in order to send the data by typing a "FastSerialPort2(Serial2)" under the line "FastSerialPort1(Serial1)"

```
FastSerialPort1(Serial1);       // GPS port
FastSerialPort2(Serial2);       // Data send port
```

b) Add new line of code for defining of baud rate or speed of serial port 2 by typing a "Serial2.begin(115200)" under the line "void setup() {"

```
void setup() {
   Serial2.begin(115200);
```

c) Add new line of code for defining of variables that would like to send by typing a "Log_SendParam();" under the line "static void fast_loop()"

```
static void fast_loop()
{
     Log_SendParam();
```

14) Go to "APM_Config.h" tab and then delete "//" in front of "#define CONFIG_APM_HARDWARE APM_HARDWARE_APM2" in order to enable the configuration of APM 2.XX hardware

15) Next, go to "Log" tab and type the code below (in the bold text) for sending of data over the serial port 2 such as, yaw (heading), X/longitude, Y/latitude, and Z/altitude (height above the ground).

```
// Write a control tuning packet. Total length: 26 bytes
static void Log_Write_Control_Tuning()
{
    DataFlash.WriteByte(HEAD_BYTE1);
    DataFlash.WriteByte(HEAD_BYTE2);
    DataFlash.WriteByte(LOG_CONTROL_TUNING_MSG);

    DataFlash.WriteInt(g.rc_3.control_in);          // 1
    DataFlash.WriteInt(sonar_alt);                  // 2
    DataFlash.WriteInt(baro_alt);                   // 3
    DataFlash.WriteInt(next_WP.alt);                // 4
    DataFlash.WriteInt(nav_throttle);               // 5
    DataFlash.WriteInt(angle_boost);                // 6
    DataFlash.WriteInt(climb_rate_actual);          // 7
    DataFlash.WriteInt(g.rc_3.servo_out);           // 8

    DataFlash.WriteByte(END_BYTE);
}
//Send Parameter to Serial2
static void Log_SendParam(){Serial2.printf("%4.3, %4.6f, %4.6f, %4.3f\n",
                (float)ahrs.yaw_sensor/100,
                (float)current_loc.lng/10000000,
                (float)current_loc.lat/10000000,
                (float)current_loc.alt/100);
}
```

16) Final, click on "Verify" (Check mark button), when it completes click "Upload" (Arrow button) and wait until finished.

C.2.2. Receiving of data

The steps are used for receiving of data will be described as follows:

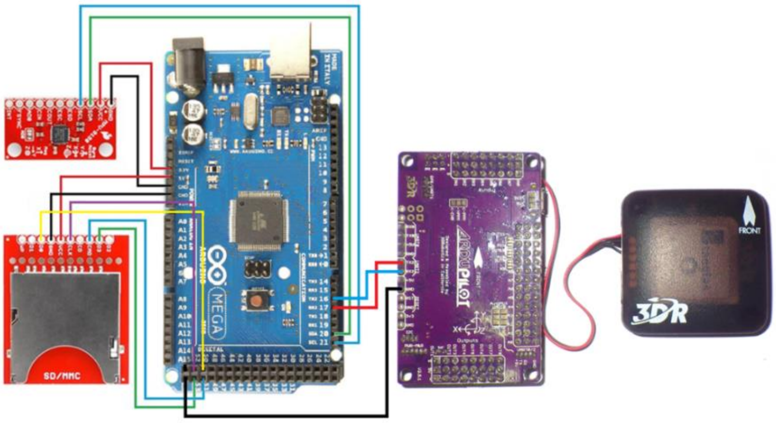


Figure C.4 The connection of all devices.

Table C.2 The pin connection of Arduino Mega 2560 with MPU-9150, Breakout board for SD-MMC, and APM 2.5 controller board

No.	Arduino MEGA 2560 (Pin No.)	MPU-9150 (Pin No.)	SD-MMC card breakout board (Pin No.)	APM 2.5 (Pin No.)
1	3.3V	VCC	-	-
2	5V	-	VCC	5V
3	GND	GND	GND	GND
5	SDA 20	SDA	-	-
6	SCL 21	SCL	-	-
7	50	-	D0	-
8	51	-	CMD	-
9	52	-	CLK	-
10	53	-	D3	-
11	TX2 16	-	-	RX2
12	RX2 17	-	-	TX2

1) Connect Arduino Mega 2560 controller board, MPU-9150, and Breakout board for SD-MMC card together via male to female jumpers – ribbon wire. Including APM 2.5 controller board, which have shown in Figure C.4 and Table C.2

2) Download MPU-9150 library from this link https://github.com/richards-tech/MPU9150Lib and then move it to default libraries folder.

3) Go to File/Preferences and select the Arduino folder in C:\Program Files (x86)\Arduino, and then click “OK”.

4) Go to Tools/Board Type and select the “Arduino Mega 2560 or Mega ADK”.

5) Go to Tools/Programmer and select the “USPasp”.

6) Connect the USB to Arduino Mega 2560 controller board and computer.

7) Start Arduino and select the COM port assigned to Arduino Mega 2560 controller board (from Device Manager in windows).

8) If Arduino 2560 controller board has an older firmware should be cleared the EEPROM before upload a new firmware.

9) In Arduino, go to File/New for creating a new firmware in order to test the receiving of data from APM 2.5 by typing the code as below.

```
#include <FastSerial.h>

FastSerialPort0(Serial);
FastSerialPort2(Serial2);
String inData;

void setup() {
  // initialize both serial ports:
  Serial.begin(115200);
  Serial2.begin(115200);
}
```

```
void loop() {
 while (Serial2.available())
 {
   char recieved = Serial2.read();
   inData += recieved;
    if( recieved == '\n')  // is this the terminating carriage return
    {
          Serial.print("Rx: ");
          Serial.print(inData);
       inData="";
    }
   }
}
```

After that save a firmware. Then verify and upload into Arduino Mega 2560. For testing should follow steps as below.

a) Connect APM 2.5 to a computer via USB cable and then open a Mission planer and click connect by selecting of baud rate at 115200 and com port that assigned to AMP 2.5 controller board. After that wait until GPS is fixed.

b) In Arduino, go to Tools/Serial Monitor and then select the baud rate at 115200 for Monitoring of data transfer from APM 2.5 to Arduino MEGA 2560.

10) In Arduino, go to File/Open and select the ArduinoDual9150.INO file in MPU-9150 library folder in order to test the MPU-9150 for tracking of data. After that verify and upload the MPU-9150 firmware into Arduino Mega 2560. Then go to Tools/Serial Monitor and select the baud rate at 115200 for Monitoring of data tracking from MPU-9150.

11) In Arduino, go to File/Sketchbook/Libraries/ReadWrite in order to load the SD card firmware for testing of reading and writing a data into SD card. In SD card fimware or "ReadWrite" tab change the pinMode(10, OUTPUT) and SD.begin(4) to pinMode(53, OUTPUT) and SD.begin(53) respectively. Then verify and upload the firmware into Arduino Mega 2560. For starting to write the data into SD card should

format SD card first (FAT or FAT 16), after that go to Tools/Serial Monitor and select the baud rate at 9600.

12) Next, for recording of data according to defined conditions such as interval recording. The code in step 9-11 have to integrate together including time conditions for interval imaging and data logging.

After that save a firmware. Then verify and upload into Arduino Mega 2560. For testing should follow steps as below.

a) Connect APM 2.5 to a computer via USB cable. Then open a Mission planer and click connect by selecting of baud rate at 115200 and com port that assigned to AMP 2.5 controller board. After that wait until GPS is fixed.

b) In Arduino, go to Tools/Serial Monitor and then select the baud rate at 115200 in or der to start the data recording from all sensors to SD card (see Figure C.5).

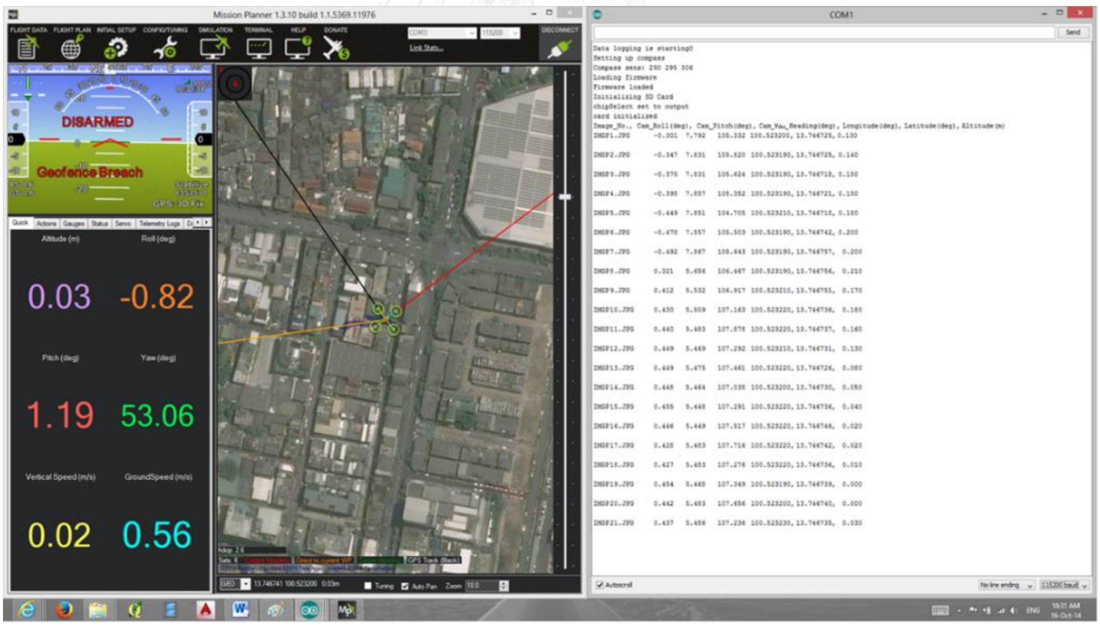
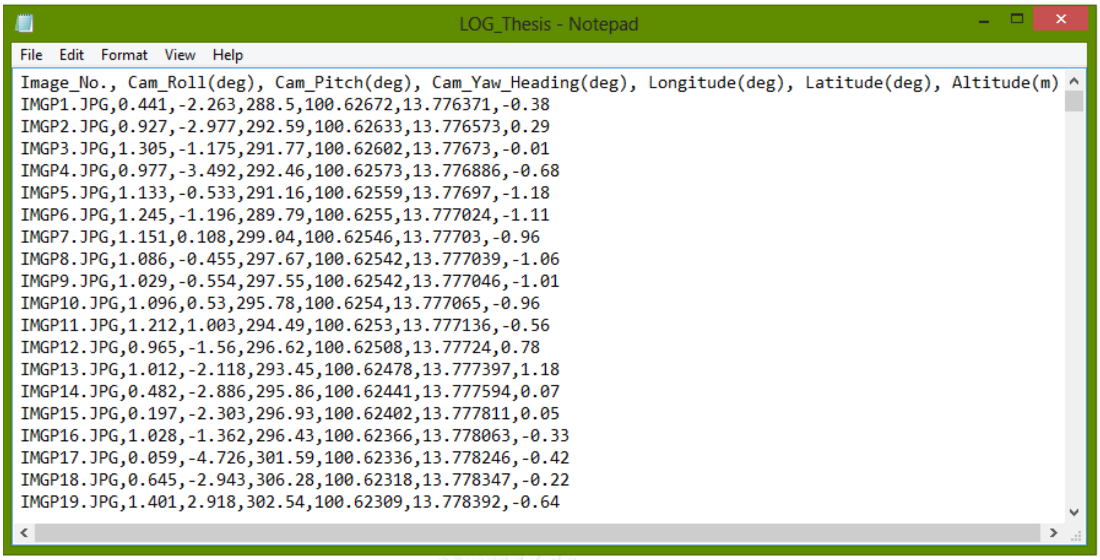


Figure C.5 Show the monitoring of data recording from all sensors to SD card.

13) Final, open the file “LOG.txt” from root of SD card for checking of data. The recorded data will be comprised the image number, camera roll, pitch, camera yaw (heading), longitude, latitude, and altitude (height above the ground) which have shown in Figure C.6.



LOG_Thesis - Notepad

File Edit Format View Help

```
Image_No., Cam_Roll(deg), Cam_Pitch(deg), Cam_Yaw_Heading(deg), Longitude(deg), Latitude(deg), Altitude(m)
IMGP1.JPG,0.441,-2.263,288.5,100.62672,13.776371,-0.38
IMGP2.JPG,0.927,-2.977,292.59,100.62633,13.776573,0.29
IMGP3.JPG,1.305,-1.175,291.77,100.62602,13.77673,-0.01
IMGP4.JPG,0.977,-3.492,292.46,100.62573,13.776886,-0.68
IMGP5.JPG,1.133,-0.533,291.16,100.62559,13.77697,-1.18
IMGP6.JPG,1.245,-1.196,289.79,100.6255,13.777024,-1.11
IMGP7.JPG,1.151,0.108,299.04,100.62546,13.77703,-0.96
IMGP8.JPG,1.086,-0.455,297.67,100.62542,13.777039,-1.06
IMGP9.JPG,1.029,-0.554,297.55,100.62542,13.777046,-1.01
IMGP10.JPG,1.096,0.53,295.78,100.6254,13.777065,-0.96
IMGP11.JPG,1.212,1.003,294.49,100.6253,13.777136,-0.56
IMGP12.JPG,0.965,-1.56,296.62,100.62508,13.77724,0.78
IMGP13.JPG,1.012,-2.118,293.45,100.62478,13.777397,1.18
IMGP14.JPG,0.482,-2.886,295.86,100.62441,13.777594,0.07
IMGP15.JPG,0.197,-2.303,296.93,100.62402,13.777811,0.05
IMGP16.JPG,1.028,-1.362,296.43,100.62366,13.778063,-0.33
IMGP17.JPG,0.059,-4.726,301.59,100.62336,13.778246,-0.42
IMGP18.JPG,0.645,-2.943,306.28,100.62318,13.778347,-0.22
IMGP19.JPG,1.401,2.918,302.54,100.62309,13.778392,-0.64
```

Figure C.6 Show the sample of recorded data in text file

C.3. Digital camera shutter

The steps of controlling of a camera shutter will be listed as follows:

1) Connect the Infrared (IR) transmitter to Arduino Mega 2560 controller board via male jumpers – ribbon wire which have shown in Figure C.7 and Table C.3.

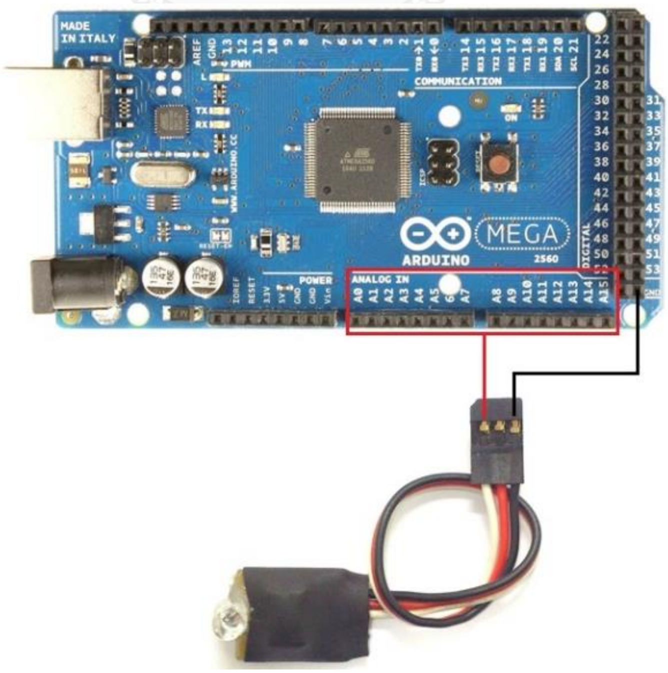


Figure C.7 The connection of Arduino Mega 2560 with IR transmitter

Table C.3 The pin connection of Arduino Mega 2560 controller board with IR transmitter

No.	Arduino MEGA 2560 (Pin No.)	IR transmitter (Pin color)
1	A0-A15	White
2	-	Red
3	GND	Black

2) Download Multi camera IR control library from this link http://sebastian.setz.name/wp-content/uploads/2010/12/multiCameraIrControl_1-91.zip and then move it to default libraries folder.

3) In Arduino, go to File/Preferences and select the Arduino folder in C:\Program Files (x86)\Arduino, and then click “OK”.

4) Go to Tools/Board Type and select the “Arduino Mega 2560 or Mega ADK”.

5) Go to Tools/Programmer and select the “USPasp”.

6) Connect the USB to Arduino Mega 2560 controller board and computer.

7) Start Arduino and select the COM port assigned to Arduino Mega 2560 controller board (from Device Manager in windows).

8) If Arduino 2560 controller board has an older firmware should be cleared the eeprom before upload a new firmware.

9) In Arduino, go to File/Open and select the “Pentax.INO” file from multi camera IR control firmware and change the code in “Pentax” tab to the code as shown below.

```
#include <multiCameraIrControl.h>

// Define IR shutter PIN (Analog pin)
#define irPin A0
// Library of a camera
Pentax WG_2(irPin);


void setup(){
```

```
}
void loop(){
  WG_2.shutterNow();
  delay(5000);   //delay 5000 millisecond
}
```

After that save a firmware. Then verify and upload into Arduino Mega 2560. For testing should follow steps as below.

10) Next, test the IR signal by sighting digital camera or smartphone camera to the IR LED (cannot see the light of IR directly) if it was shining and blinking like a Figure C.8 is meaning that the IR transmitter is ready to use.

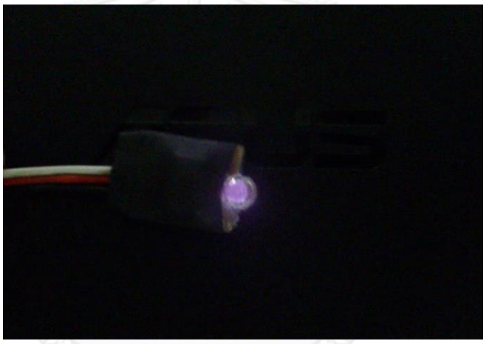


Figure C.8 Light detection of IR LED by using a digital camera

11) Final, test the IR transmitter to control the digital camera shutter by moving IR transmitter to the in front of digital camera closely to the IR receivers (see Figure C.9) of camera, if a camera was starting to take a photo by itself automatically is meaning that the control of cameras shutter by using the IR transmitter was successful.

Figure C.9 Infrared (IR) receivers of Pentax Optio WG-2 digital camera

Appendix D
Data logger firmware

The following code are used for developed data logger, which based on based on C/C++ programming

```
#include <Wire.h>
#include "I2Cdev.h"
#include "MPU9150Lib.h"
#include "CalLib.h"
#include <dmpKey.h>
#include <dmpmap.h>
#include <inv_mpu.h>
#include <inv_mpu_dmp_motion_driver.h>
#include <EEPROM.h>
#include <SD.h>

String inData;

//  DEVICE_TO_USE selects whether the IMU at address 0x68 (default) or 0x69 is used
//    0 = use the device at 0x68
//    1 = use the device at ox69

#define  DEVICE_TO_USE    0

MPU9150Lib MPU;   // the MPU object

//  MPU_UPDATE_RATE defines the rate (in Hz) at which the MPU updates the sensor data and DMP output

#define MPU_UPDATE_RATE  (20)
```

```
//  MAG_UPDATE_RATE defines the rate (in Hz) at which the MPU updates the magnetometer data
//  MAG_UPDATE_RATE should be less than or equal to the MPU_UPDATE_RATE

#define MAG_UPDATE_RATE  (10)

//  MPU_MAG_MIX defines the influence that the magnetometer has on the yaw output.
//  The magnetometer itself is quite noisy so some mixing with the gyro yaw can help
//  significantly. Some example values are defined below:

#define  MPU_MAG_MIX_GYRO_ONLY          0    // just use gyro yaw
#define  MPU_MAG_MIX_MAG_ONLY           1   // just use magnetometer and no gyro yaw
#define  MPU_MAG_MIX_GYRO_AND_MAG       10   // a good mix value
#define  MPU_MAG_MIX_GYRO_AND_SOME_MAG  50   // mainly gyros with a bit of mag correction

//  MPU_LPF_RATE is the low pas filter rate and can be between 5 and 188Hz

#define MPU_LPF_RATE   40

const int chipSelect = 53;       // CS pin (Arduino boards,4 on the Uno, 53 on the Mega)
long Number = 1;

//  SERIAL_PORT_SPEED defines the speed to use for the debug serial port
#define  SERIAL_PORT_SPEED  115200

unsigned long previousMillis  = 0;
unsigned long currentMillis   = 0;
unsigned long fMillis  = 0;
unsigned long sMillis   = 0;
int interval = 3000; //set interval (milliseconds)
int DL = 30000;  //set delay (milliseconds)
```

```
void setup()
{
  Serial.begin(SERIAL_PORT_SPEED);
  Serial2.begin(SERIAL_PORT_SPEED);
  Serial.print("Data logging is starting"); Serial.println(DEVICE_TO_USE);
  Wire.begin();
  MPU.selectDevice(DEVICE_TO_USE);     // only really necessary if using device 1
  MPU.init(MPU_UPDATE_RATE, MPU_MAG_MIX_GYRO_AND_MAG, MAG_UPDATE_RATE,
 MPU_LPF_RATE);  // start the MPU
  // intialize SD Card
  Serial.println("Initializing SD Card");
    // Make sure that the default chip select pin is set to
  // output, even if you don't use it:
  pinMode(chipSelect, OUTPUT);
  Serial.print("chipSelect set to output\n");

  if (!SD.begin(chipSelect))
  {
    Serial.println("Card failed, or not present\n");
    // don't do anything more;
    return;
  }
  Serial.println("card initialized");

  File logFile = SD.open("LOG.txt", FILE_WRITE);
  if (logFile)
  {
    String header = "Image_No., Cam_Roll(deg), Cam_Pitch(deg),
    Cam_Yaw_Heading(deg), Longitude(deg), Latitude(deg), Altitude(m)";
    logFile.println(header);
```

```
    logFile.close();
    Serial.println(header);
  }
  else
  {
    Serial.println("Couldn't open log file");
  }
}
void loop()
{
  while (Serial2.available())
  {
  char recieved = Serial2.read();
  inData += recieved;
    if( recieved == '\n' )  // is this the terminating carriage return
    {
      MPU.selectDevice(DEVICE_TO_USE);
      if (MPU.read())
      {
        fMillis = millis();
        // check to see if the interval time is passed
        if (fMillis - sMillis >= DL == true)
        {
          currentMillis = millis();
          // check to see if the interval time is passed
          if (currentMillis - previousMillis >= interval == true)
          {
            File dataFile = SD.open("LOG.txt", FILE_WRITE);
            if(dataFile)
            {
             // print and write image number
```

```
                dataFile.print("PTAX");dataFile.print(Number);dataFile.print(".JPG");
                Serial.print("PTAX ");Serial.print(Number);Serial.print(".JPG");
               dataFile.print(",");
               Serial.print("\t");
             // print and write calculated gyro data
               dataFile.print((MPU.m_dmpEulerPose[0]*-180/PI),3);
               Serial.print((MPU.m_dmpEulerPose[0]*-180/PI),3);
               dataFile.print(",");
               Serial.print("\t");
               dataFile.print((MPU.m_dmpEulerPose[1]*180/PI),3);
               Serial.print((MPU.m_dmpEulerPose[1]*180/PI),3);
               dataFile.print(",");
               Serial.print("\t");
            // print and write data from serial2
               dataFile.print(inData);
               Serial.print(inData);
               Serial.println();
               dataFile.close();
            }
            else
            {
              Serial.println("Couldn't access file");
            }
            Number++;
            previousMillis = currentMillis;
          }
        }
      }
      inData = "";
    }
  }
```

Appendix E
Calibration of camera, gimbal and data logger

E.1. Camera calibration

For calibration of camera lens by using the Agisoft Lens software have steps are as follow:

1) Capture the photos of the calibration pattern.

a) Select “Show chessboard” command from the Tools menu to display the calibration pattern.

b) Capture a series of photos of the displayed calibration pattern with Pentax Optio WG-2 digital camera from slightly different angles, according to the guidelines below (see Figure E.1). Minimum number of photos for a given focal length is 3.

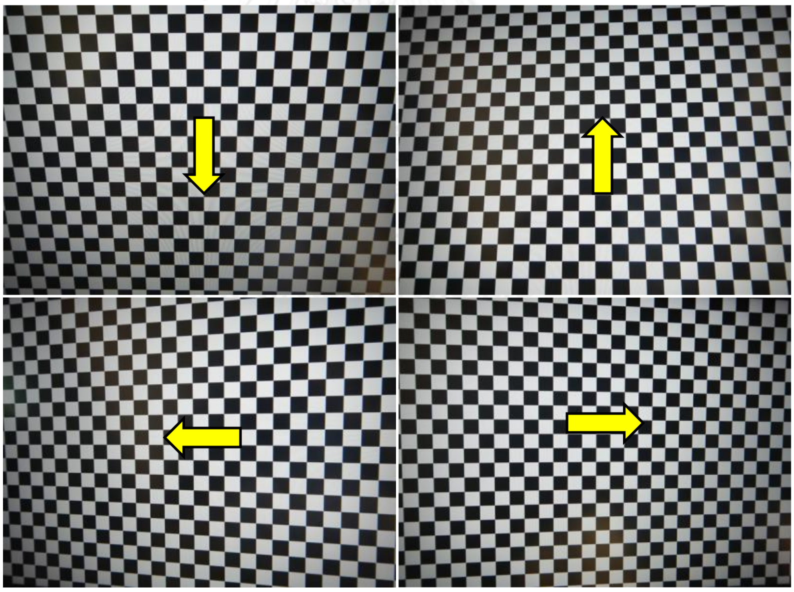


Figure E.1 Sample photos of a calibration target

When capturing photos of the calibration pattern, try to fulfill the following guidelines:

- Avoid glare on the photos. Move the light sources away if required.

- Preferably, the whole area of the photos should be covered by calibration pattern. Move the camera closer to the LCD screen if required.

c) Click anywhere on the calibration pattern or press Escape button to return to the program.

d) Upload the captured photos to the computer.

2) Load the photos of the calibration pattern.

a) Select “Add Photos...” command from the Tools menu

b) In the "Open File” dialog box, browse to the folder, containing the photos, and select files to be processed. Then click Open button.

3) Select “Calibrate...” command in the Tools menu.

4) In the “Calibration” dialog box, select the desired calibration parameters. Click OK button when done. After finished calibration, the results will appear in the report window. The following calibration parameters are estimated during calibration and printed in the report window:

- fx - horizontal focal length, in pixels
- fy - vertical focal length, in pixels
- cx - X coordinate of the principal point
- cy - Y coordinate of the principal point
- K1, K2, K3, P1, P2 – radial lens distortions in Brown’s model

After calibration is finished, it will be presented with the following information:

- Detected chessboard corners which have shown in Figure E.2).

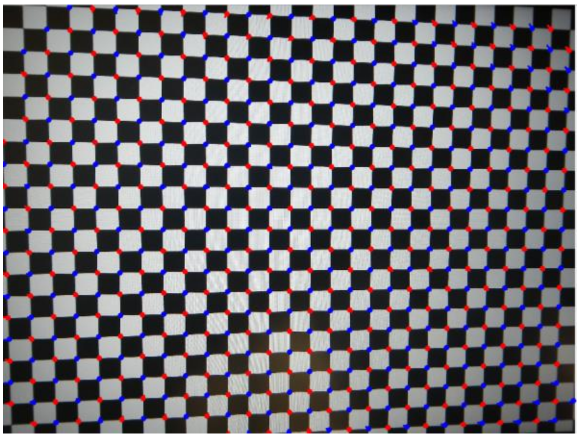


Figure E.2 Detected chessboard corner from a photo

- In the report window estimated calibration of interior orientation parameters and their errors are displayed such as, radial and tangential distortion. Which have shown in Figure E.3.

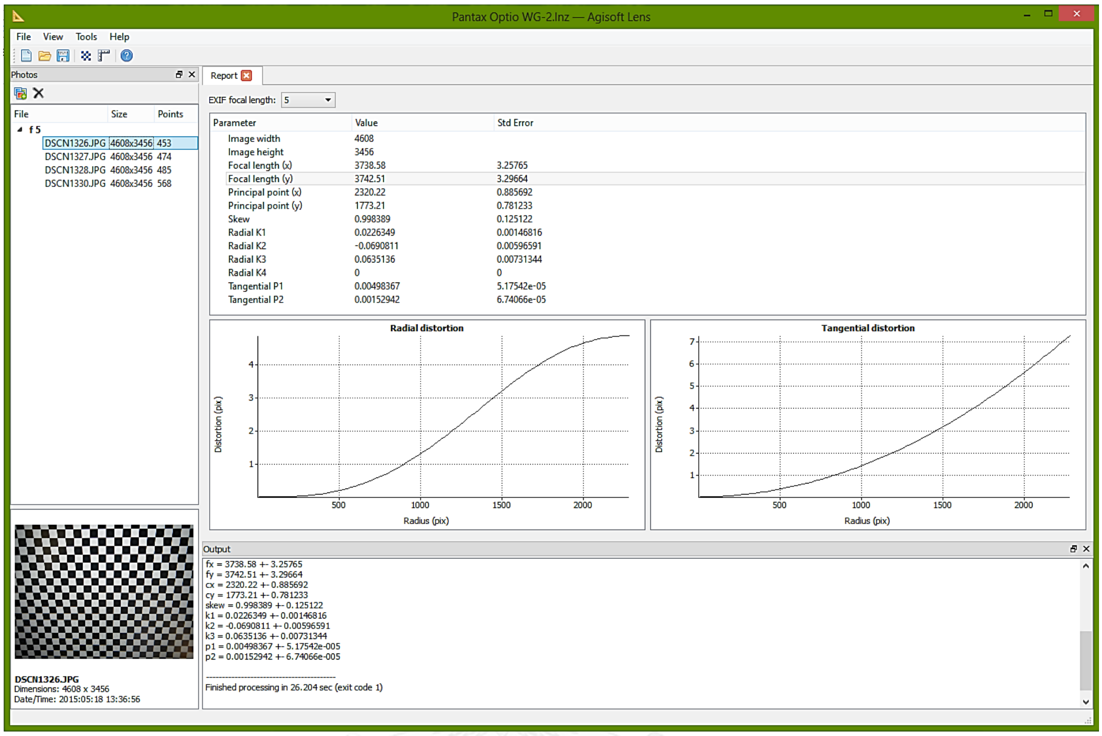


Figure E.3 The estimated calibration of interior orientation parameters their errors

5) Final, select "Save calibration..." command in the File menu.

E.2. Gimbal calibration

The steps of gimbal calibration are as follows:

1) Prepare the gimbal that finished tuning in section C.1.2

2) Use the pond bubble which shows in Figure E.4 in order to adjust the orientation of a camera to vertically by attaching on the screen of a camera, which a camera was installed on the gimbal frame and face down to the floor for vertical imaging (see Figure E.5).

Figure E.4 A pond bubble tube

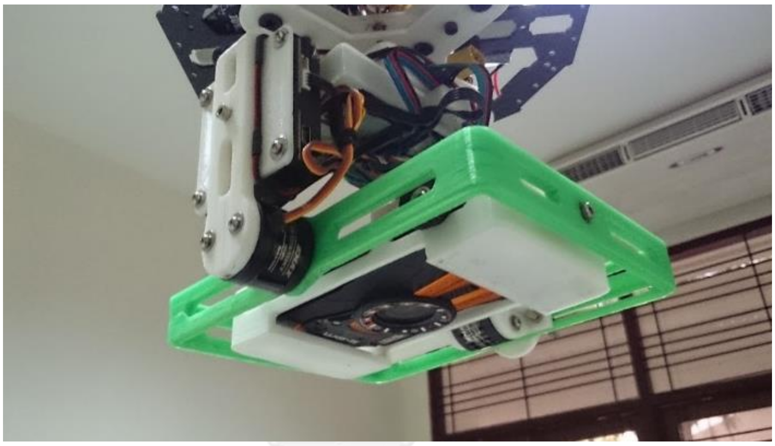


Figure E.5 The installation of a camera on gimbal frame

3) Hang the gimbal or attach on UAV (Hexacopter).

4) Connect the battery to brushless gimbal controller board.

5) Connect the brushless gimbal controller board to the computer via USB cable

6) Start the BruGi software and then select the port and click connect.

7) In the “Settings” tab click on “Get from EEPROM” for loading of exiting of values from brushless gimbal controller board that finished tuning.

8) Go to “Calibration” tab. In ACC Sensor, click ACC Calibration button and wait until finished. Then do the same for GYRO Sensor, which do not mark on Calibration “at Startup ON”

9) Next, adjust the values of “Offset X” and “Offset Y” in GYRO Sensor by adding and reducing their values (see Figure E.6) in order to bring the pond bubble into the center of the pond bubble tube.

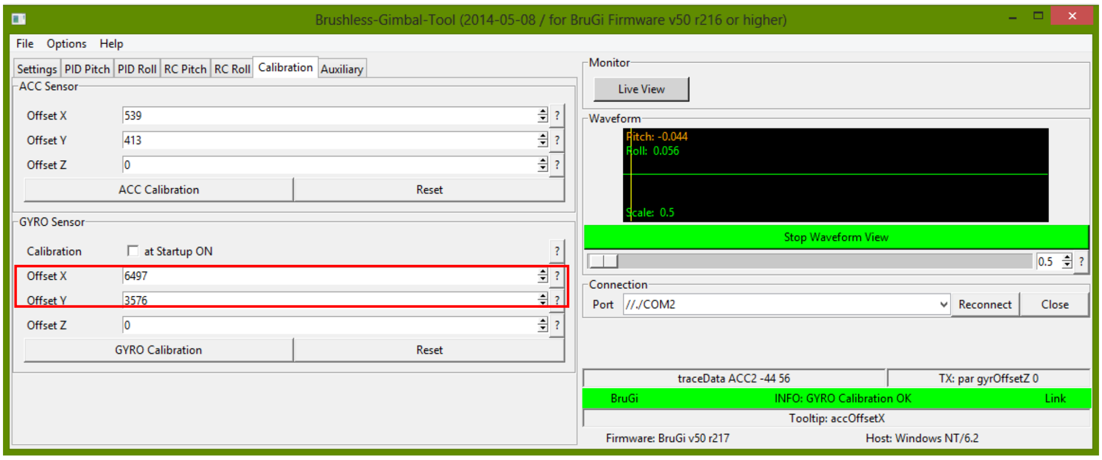


Figure E.6 The offset calibration of a gyroscope in BruGi software

10) After finished adjusting of “Offset X” and “Offset Y” in step 9. Then go to “Settings” tab again and click on Save to EEPROM button for saving of adjusted values into the brushless gimbal controller board.

11) Final, disconnect the USB cable and battery from brushless gimbal controller board. After that, reconnect a battery for checking of calibrated gimbal by using a pond bubble in order to estimate the vertical of a camera

E.3. Data logger calibration

The steps of data logger calibration will be described as follows:

1) Prepare the gimbal that finished tuning and calibrating in section C.1.2 and E.2 respectively.

2) Connect the Arduino Mega 2560 to the computer.

3) In Arduino, go to File/Open and load the data logger firmware from section D. Then set the time of interval recording by starting at 3000 milliseconds. Because it is a default self-timer in camera shutter control mode of Pentax Optio WG-2 digital camera. For change time see code line as below.

```
int interval = 3000; //set interval (milliseconds)
```

Then save a firmware, verify and upload into Arduino Mega 2560.

4) Connect the battery to brushless gimbal controller board. Then it will be operated automatically to stabilize a camera.

5) Connect the battery to APM 2.5 controller board.

6) Power on a camera and set their shutter control by using infrared. Then set the focus to infinity.

7) Connect the APM 2.5 controller board to the computer via micro USB cable. Then open a Mission planer and click connect by selecting of baud rate at 115200 and com port that assigned to AMP 2.5 controller board. After that, wait until data logger has been record the data and take a photo.

8) Next, to observe the data, it should be take time about an hour or more (not fixed). Due to UAV cannot fly longer than this. Particularly, multi-rotor.

9) After that, check the data from SD card of camera and data logger whether the images and exterior orientation parameters have equal number of images and logging data record or not. If equal do the next step, if not repeat the step 2-9 by adding time for data logging about 500 milliseconds.

10) Next, calculate the average of roll and pitch angles from log file in order to calibrate the IMU orientation to close zero by adding or reducing of their values to the data logger firmware.

11) In Arduino, go to File/Open and load the data logger firmware from step 3. Then edit the following code:

- For roll angle

```
dataFile.print((MPU.m_dmpEulerPose[0]*-180/PI) ”+ or – Average of roll angle“,3);
Serial.print((MPU.m_dmpEulerPose[0]*-180/PI) “+ or – Average of roll angle”,3);
```

- For pitch angle

```
dataFile.print((MPU.m_dmpEulerPose[1]*180/PI) ”+ or– Average of pitch angle“,3);
Serial.print((MPU.m_dmpEulerPose[1]*180/PI) ”+ or – Average of pitch angle“,3);
```

Final, save a firmware. Then verify and upload into Arduino Mega 2560.

VITA

Mr. Santisouk Kongkeo was born in Vientiane, Laos P.D.R. He was graduated the bachelor degree in Civil Engineering in 2012 from National University of Laos (NUOL). He got a CU-ASEAN scholarship from Chulalongkorn University and then he was studied in the field of survey engineering in 2013.