University of Nottingham
UK | CHINA | MALAYSIA

EEE Department of Electrical and Electronic Engineering

H53PJ3
Final Year Project Report

The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

“Hands-free" control of a Quadcopter

AUTHOR:	**NANDI GUO**
ID NUMBER:	**14311495**
SUPERVISOR:	**Dr. Steve Greedy**
MODERATOR:	**Dr. James Bonnyman**

Abstract

The quadcopter, or drone, is generally controlled by a handheld remote controller which could be a challenge for beginners or those with limited mobility, or mobility issues.

Research has shown that most of the beginners do not have enough knowledge about the fly modes of a copter (Throttle, pitch, yaw, roll) and each function of the joystick on the complex handheld controller. This project aims to remotely control a Quadcopter by using a system based on gesture recognition to bring a fresh and easy control concept for users. Based on the targeted project, it asks:

Which kind of gesture sensors would be used, how the quadcopter is controlled by the sensor and how the signal could be transferred to the quadcopter?

A hand gesture sensor called Leap-Motion would be used as a control component, users could control the quadcopter easily by wave their hands, the height of the hand determines the flying height, and the plane position of the hand determines the flying direction. The signal generated by the gesture sensor would be sent to a microcontroller which is able to control the mechanical components which are fixed on the original handheld controller.

Analysis of the experimental demonstrator developed through this project showed that the gesture recognition sensors could achieve precise control of a quadcopter with a simpler learning curve than would otherwise be possible if trying to use the traditional handheld controller.

Table of Contents

Chapter 1: Introduction

In this chapter, the introduction and the aims of the project would be illustrated, the background of the components mainly used in the project would be introduced. The objectives of the project would be also listed in this chapter.

Section 1.1: An Introduction

The popularity of quadcopter has been increasing in recent years. The quadcopter is a complicated flying mechanic's vehicle that has four arms, and there is a motor attached to a propeller in every arm. The quadcopter is normally controlled by a handheld radio remote controller. Using the original remote controller to control the quadcopter will be a challenge for beginners. This project aims to remotely control a Quadcopter by using a gesture recognition sensor. Therefore achieving 'hands-free' control for the users.

The aims of the project are listed:

a. This project aims to offer a human interface based on gesture recognition instead of the traditional radio controller.

b. The traditional remote controller should be modified and connected to the gesture recognition sensor system and the sensor should be able to control the quadcopter to fly.

c. A flight control system of gesture recognition should be implemented.

d. Complete and try to upgrade the control system.

Section 1.2: Background

Gesture recognition

Gesture recognition is an active research field which tries to integrate the gestural channel in Human-Computer Interaction. It has applications in virtual environment control, but also in sign language translation, robot remote control or musical creation.

Recognition of human gestures comes within the more general framework of pattern recognition. In this framework, systems consist of two processes: the representation and the decision processes. The representation process converts the raw numerical data into a form adapted to the decision process which then classifies the data (see Figure 1.21). [1]

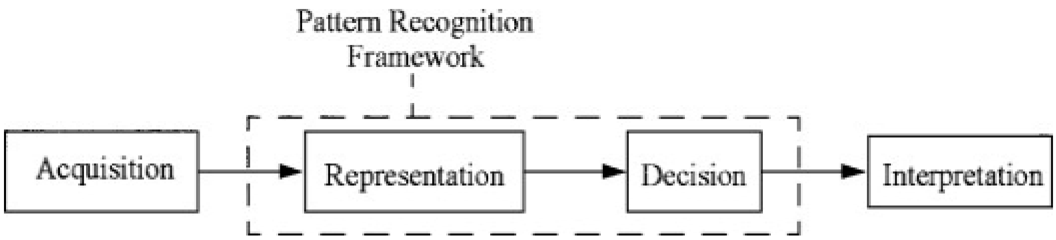


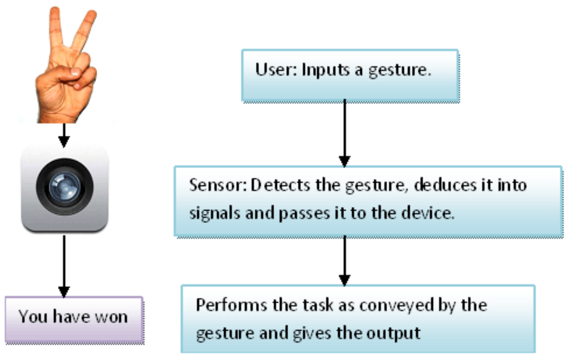
Figure 1.21: representation process of recognition [1]

A sensorial computing user interface that allows machines to Identify and calculate human gestures as commands are gesture recognition. In general, the explanation of gesture recognition is the capability of a computer to comprehend gestures then perform commands according to human gestures. Consumers are familiar with gesture recognition through VR devices and X-box or PlayStation games. In the gesture recognition area, gestures are defined as physical movement, it could be large movement or small movement, these movements can be recognized by a gesture sensor. It includes anything from the points of fingers to a nod of the head to a pinch or roundhouse kick or wave of the hand. Gestures may be contained or broad. In some instances, the definition of the gesture may also include voice.

The gestures can be classified into several types, such as Pre-emptive Gestures, Function Associated Gestures, Context-Sensitive Gestures, Global Shortcut Gestures, Natural Dialogue Gestures.

The gesture recognition system could also be classified to different types, such as Contact type, Non-Contact, Device Gesture Technologies, Vision-based Technologies, Electrical Field Sensing, etc. [2]

While there are many different types of gesture recognition technology, they all work on the same basic principle of recognizing human movement as a form of input. The device features one or more sensors — or cameras — that monitor the user's movement. When it detects a movement that corresponds with a command, it responds with the appropriate output. This may be unlocking the device, launching an app, changing the volume, etc.



A general way is cameras send picture data into a sensor that is connected to a computer. The sensor system then makes use of an infrared sensor for measuring depth. Showed in figure 1.22 [3].

Figure 1.22: A Figure Showing Working of the Gesture Recognition System

Normally, gesture recognition software identifies gestures from a predetermined library where all gestures are already matched to a computer command.

The software will correlate every real-time gesture, recognizes the gesture and take the library to identify various gestures that coincide with the library. As soon as the gesture was interpreted, the computer runs the command correlated to that particular gesture.

Currently, gesture recognition technology has three mature solutions: "structure light", "time of flight" and "multi-camera". [4]

Overall, using gesture recognition could Greatly improve the precision of human control of the machine and Simplifies the way humans interact with machines. Using gesture recognition for remote control also brings a whole new experience for humans. [4] The example can be seen below.

Figure 1.23: gesture recognition [4]

Gestures detected methods

Depending on the type of input data, the approach for interpreting a gesture could be done in different ways. However, most of the techniques rely on key pointers represented in a 3D coordinate system. Based on the relative motion of these, the gesture can be detected with high accuracy, depending on the quality of the input and the algorithm's approach. There are three basic algorithms, 3D model-based algorithms, Skeletal-based algorithms, Appearance-based models. [5]

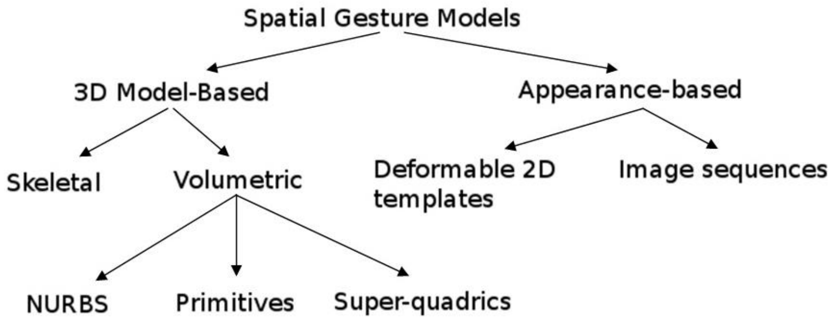


Figure 1.24: gesture recognition algorithms [5]

According to different hardware implementations, there are about the following gesture recognitions used in the industry:

a. Structure Light calculates the location and depth data of an object with infrared refraction and algorithm, thus restoring the whole three-dimensional space. Typical products of structured light are Microsoft's Kinect generation [6]. However, since the location of the refracted light was calculated to calculate the location, this technique cannot compute accurate depth data, and there are stringent requirements for the distance to be recognized.

Figure 1.25: Microsoft's Kinect generation [6]

b. Time of Flight, charging a light-emitting component, capturing the flight time of the calculated photon through the CMOS sensor, and calculate the distance of the photon flight that is based on a photon flight time, and receiving depth data of the object. A representative for the work is Intel's 3D camera [7] with gesture recognition.

Figure 1.26: Intel's 3D camera [7]

c. Multi-angle imaging is now used by Leap Motion [8], leading in gesture recognition. It uses multiple cameras to simultaneously acquire images. By comparing the differences between the images obtained by these different cameras at the same time, an algorithm is used to calculate the depth information, thereby multi-angle three-dimensional imaging.

Figure 1.27: Leap motion on a VR device [8]

Quadcopter

Quadcopters have been around for more than two decades, but their roots date back to World War I when both the U.S. and France worked on developing automatic, unmanned aeroplanes. But the last few years have been significant in terms of drone adoption, usage expansion across industries, and global awareness. [9]

From technically manning sensitive military areas to luring hobbyists throughout the world, drone technology has developed and prospered in the last few years. Individuals, commercial entities, and governments have come to realize that drones have multiple uses, which include:

1. Aerial photography for journalism and film
2. Express shipping and delivery
3. Gathering information or supplying essentials for disaster management

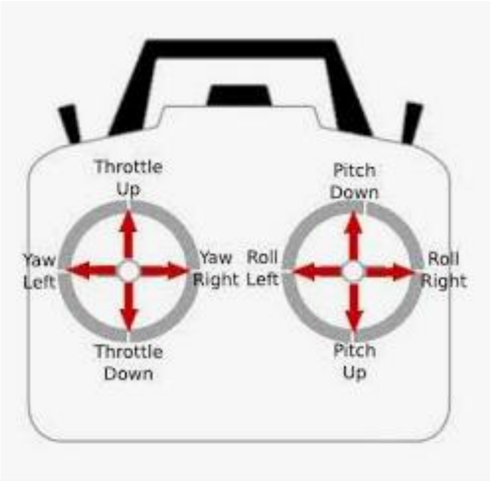
4. Thermal sensor drones for search and rescue operations
5. Geographic mapping of inaccessible terrain and locations
6. Building safety inspections
7. Precision crop monitoring
8. Unmanned cargo transport
9. Law enforcement and border control surveillance
10. Storm tracking and forecasting hurricanes and tornadoes

Development of hundreds of more uses of drones is underway due to the multiple investments pouring into this promising industry every day. [9]

A quadcopter is a complex flying mechanic's vehicle that has four arms, and there is a motor attached to a propeller in every arm. They have gained popularity in recent years because they have big advantages over fixed-wing aircraft and the traditional helicopter. The quadcopter is relatively small and omnidirectional which means it has much better manoeuvrability than the fixed-wing aircraft and helicopters.

A typical quadcopter is using a 4-channel controller that sends commands in the copter to control its throttle (up & down), pitch, yaw, roll (showed in figure 3&4). The
transportations frequency used by the majority of remote controllers is 2.4 GHz.

In general, users have to learn how to use the controller and they have to memorize all the functions before they try to fly the copter. This is a difficult process with a steep learning curve, especially for beginners. The traditional remote controller uses two joysticks that could move in four different directions respectively. Normally, one stick controls the throttle and yaw whilst the other controls the pitch and roll. Therefore, every action performed in requires the almost perfect combine effort of two hands. (Figure 1.292)



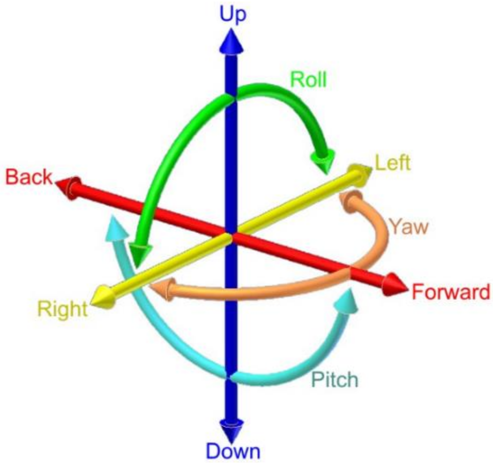


Figure 1.28: Throttle, pitch, yaw, a roll of quadcopter flight
Figure 1.29: How to fly a quadcopter

A gesture recognition system should easy the learning curve, reduce complexity and be more intuitive for the user to master.

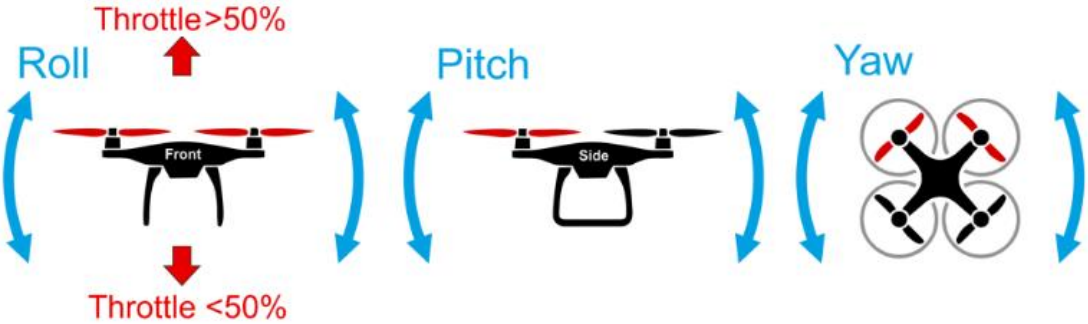


Figure 1.292: How to fly a quadcopter [10]

Section 1.3: Objective of the Project

Looking further from the aims of the project, the objectives of this project can be categorised into four major areas to occur in chronological order:

Research

- Become familiar with a microcontroller with enough knowledge to complete project
- Become familiar with C programming language and "Processing 3 IDE" software
- Become familiar with gesture sensors to choose a suitable one
- Become familiar with the control operation of Quadcopter and the functions of the handheld controller

- Understand and become familiar with microcontroller and its General-purpose Input/Output (GPIO) pins

Hardware Design and Development

- Choose a gesture/motion sensor and do a test through example programs of the sensor driver.
- Select a suitable microcontroller
- Interfacing the sensor with PC and the microcontroller
- Choose digital potentiometers/Special drive integrated circuit and design a method to drive the handheld remote-control unit.
- Design the circuit to make the whole system have a stable power supply
- Integrate gesture recognition and quad controller into a single system
- Simulating gesture recognition and control

Software Development

- Develop codes for the microcontroller to control circuits
- Develop codes for gesture sensor (include test codes)
- Develop codes to transmit the signals to make the functions of microcontroller and gesture sensor are combined

Whole system evaluation

- Evaluate and test the system’s software and hardware as a whole
- Make sure the system can at least precisely control the throttle of the quadcopter

Chapter 2: Components selection

In this chapter, the expected components are listed and illustrated, the information and reasons for choosing each component are also introduced.

Expected components

1	Gesture sensor
2	4+ Channel DSM2 Transmitter
3	Blade BLH8680G Quadcopter
4	Microcontroller
5	Potentiometers
6	SG90 servo motor/5v torque motor
7 8	Drive integrated circuit Power supply

The components are selected

1	Leap motion
2	DX8 8-Channel DSMX Transmitter
3	Inductrix blade small copter
4	Orange pip kona328
5	Mcp4725
6	SG90 servo motor
7	L298N
8	6V battery pack

All the selected components would be introduced in the following sections.

Section 2.1: Gesture sensor

During this project, two gesture sensors are considered, Microsoft's Kinect and leap motion.

Leap motion	Kinect
1. Most accurate, Embedded in PCs and tablets, better suited for PC games and apps, limited capability, awkward placement. 2. Cheap. 3. Portable and easy to connect. 4. Easy to be programmed and developed.	1. Most powerful, most popular, more games and apps, better suited for Console games and Fitness purpose, Bigger than other sensors and hence less portable. 2. Expensive 3. Harder to be developed.

Leap motion is getting popular these days. It is the most accurate sensor in all of these and it's size is very small which is an advantage. It has started coming embedded in PCs and this is revolutionary. It works for a very small range though (up to 3 feet) and does only hands and finger tracking. It is better suited for Apps and PC games. [11] Thus, the Leap motion, a hand gesture sensor, will be used as a remote controller, figure 2.1.

Leap Motion is a device that detects and tracks hands, fingers and finger-like tools. The device operates close to the user and is able to track movements with high accuracy and high tracking frame rate.

The Leap Motion field of view is an inverted pyramid centred above the device. The effective range of Leap Motion detection is about 25 mm to 600 mm (1 inch to 2 feet).

Leap Motion can effectively analyze the objects observed in its field of view. It can distinguish between hands, fingers and tools; provide a series of actions and position information; it can identify four specific actions: Circle, Swipe, Key Taps, Screen Taps; by continuously tracking the action flow, Leap Motion can Action in a region are understood as three basic elements: scaling, translation, and rotation. [12]

Leap Motion can effectively analyze the objects observed in its field of view. It can distinguish between hands, fingers and tools; provide a series of actions and position information; it can identify four specific actions: Circle, Swipe, Key Taps, Screen Taps; by continuously tracking the action flow, Leap Motion can interpret the movement of the user and represent them as three basic elements: scaling, translation, and rotation. [13]

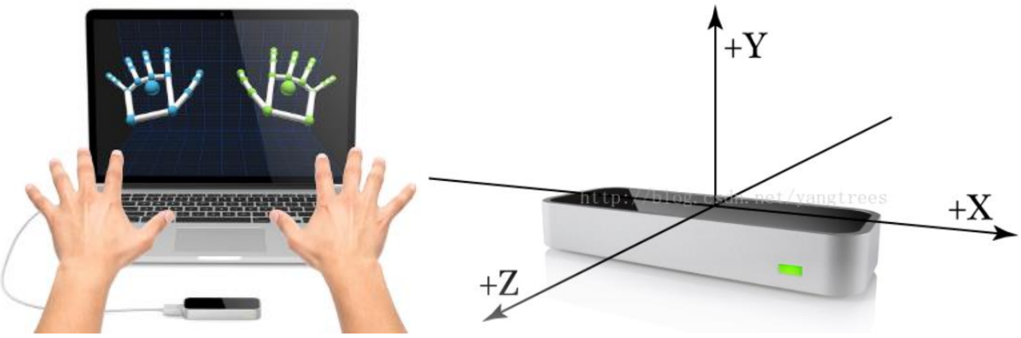


Figure 2.1: Leap motion [13]

Advantages of the Leap Sensor:

a. Easy to be connected and programmed, the leap can be connected directly to the computer by using a USB and programmed through the manufacturer's software development kit (SDK)

b. LEAP can quickly and accurately identify the position and movement of the user's hands and fingers, even for very subtle movements.

c. LEAP has a long recognition distance and is not easy to get out of control when controlling a quadcopter.

d. Lower price compared to Kinect.

Therefore, the leap motion gesture sensor was chosen to be connected with other devices mentioned in context to achieve the project.

Section 2.2: Quadcopter and Transmitter

Inductrix blade small copter

Parameters:
Required: 4+ channel DSM2/DSMX transmitter
Product Dimensions: 26.9 x 20.8 x 8.6 cm ; 18.1 g
Boxed-product Weight: 612 g
Batteries 1 Lithium Polymer batteries required.
Item model number: BLH8700
ASIN: B011UDQYSC
[14]

Figure 2.2: Blade Inductrix FPV Ultra Micro Quadcopter Drone

Reasons for selection:

1. With the Blade® Inductrix ™ quadcopter, users can fly with little effort. Exclusive Horizon Hobby SAFE® technology combines small pitch and roll angle limits with electronic self-levelling to help users stay in control and have more fun.
2. Each cylindrical rotor housing that ducts air through the high-speed rotors provides an inherently durable structure. Users can bump the Inductrix drone into practically anything with little risk.
3. Brilliant LED lights are located to provide instant orientation reference. The clear rotor housings keep your view unobstructed and add an unmistakable glow to the airframe.

Therefore, the Inductrix blade small copter was chosen to achieve the task in this project as it is small, safe and easy to use.

Dx8 transmitter

Figure 2.21: DX8 8-Channel DSMX Transmitter

Dx8 transmitter is an 8-Channel DSMX Transmitter which is able to bind with the Blade® Inductrix ™ quadcopter, and this transmitter is strong for transmitting and receiving signals. This transmitter gives pilots a powerful combination of features and capabilities no other 8-channel can match.
The DX8 lets the user keep tabs on important functions without ever taking your eyes off what you're flying. Users can program voice alerts to call out what flight mode you've chosen or report telemetry information on demand. If an alarm goes off, the user won't have to look at the transmitter display to see what's happening. The DX8 can tell you.
This exclusive Spektrum technology gives DX8 users the ability to wirelessly link up with other Spektrum transmitters and specially-equipped Spektrum FPV headsets. Once bound to another transmitter or component, ModelMatchT technology will allow the DX8 to re-link without having to go through the binding process again.
The enormous capacity of the DX8's on-board memory means you will rarely if ever, need to juggle models between transmitter memory and your SD card. The model memory menu is easier to navigate too. Only memory slots with model settings saved to them will appear. The user won't have to scroll through empty slots or move models around if one is deleted. Because the DX8 uses the same AirWareT software as many other Spektrum transmitters, user can share model setups across platforms with almost any Spektrum pilot.
The built-in telemetry feature gives you real-time information on things like your model's battery voltage, signal quality, engine or motor temperature, airspeed, altitude and more. [15]

The bind speed of the transmitter and Blade® Inductrix ™ quadcopter is very fast, binding-time less than 10s. This is important because once the quadcopter loses connection while flying, it still has time to reconnect, this may reduce a lot of losses.

The only drawback is that it is expensive.

Section 2.3: Microcontroller

Figure 2.3: Orange PIP Kona 328

The Orange PIP Kona 328 (in figure 2.3) development board centres around the ATmega328 microcontroller and comes with full Arduino ™ UNO compatibility.

Reasons for selection:

1. It is cheap.
2. It comes with an open supply hardware feature that permits users to develop their kit.
3. The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
4. Easy to learn for beginners: The programming environment is easy for beginners to use, but flexible enough for advanced users to use. Another big advantage of Arduino is the sample library included in the Arduino IDE software.
5. Its biggest advantage is the connection of the board to the computer via a USB cable which does a dual purpose of supplying power and acting as a Serial port to interface the Arduino and the computer. It can also be powered by a 9V-12V AC to DC adapter.

Thus, the orange pip kona328(Arduino UNO) was chosen to control the circuit and play a role to transmit and receive the signals from leap motion.

Section 2.4: Potentiometers selection

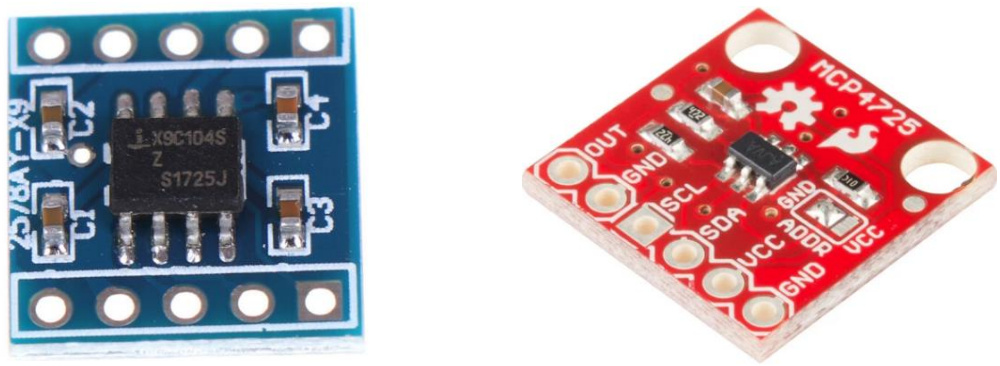


Figure 2.4: the digital potentiometer x9c104 and mcp4725 [16]

A digital potentiometer (also called a resistive digital-to-analogue converter, or informally a digipot) is a digitally-controlled electronic component that mimics the analogue functions of a potentiometer.(figure 2.4). It is often used for trimming and scaling analogue signals by microcontrollers.[16]

• It is referred to as mixed-signal device as it enables digital control of a variable voltage.
• It is an IC chip which emulates the function of the analogue potentiometer.
• It is digitally programmable as its internal resistance can be varied using input control. Hence it is used in conjunction with the microcontroller to control internal resistance of the component.

Reasons for selection:

1. It is smaller in size and hence can be placed very close to other chips on the board. Wide range of resolution
2. Much more writing cycles than mechanical parts
3. It offers a digital interface
4. It can be programmed

The application of the digital potentiometer would be introduced in Chapter 3, the x9c104 digital potentiometer is a key component for Project design concept 2.
[17]

Section 2.5: SG90 servo motor/5v torque motor selection

Figure 2.5: 5v torque motor [18]

Parameters:

Voltage range: 3V-6V DC. No-load rotating speed: 40 ±10%RPM. Ambient temperature: -5°C - 40°C.

Load torque: 0.5kg. cm. Load current: 0.06A (0.09A Max.). Load rotation: 31.4 ±10% RPM.

Rotor current: 0.29A. Rotor torque: 2.3kg. cm.

Axial play: 0.3mm or less. Motor idle current: 0.030A. Motor idle rotation speed: 11500±10 %RPM. [18]

Reasons for selection:

Small, easy to be placed and have enough torque to be used in project design concept 1 (in Chapter 3).

The application for this component would be discussed in Chapter 3.

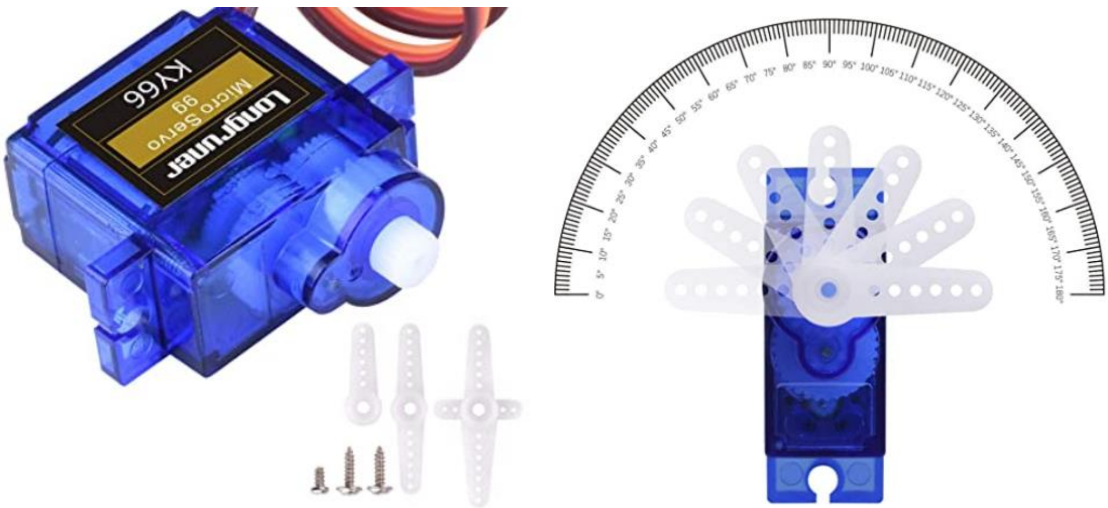


Figure 2.51: SG90 servo motor [19]

Parameters [19] :

Operating speed: 0.12second/ 60degree (4.8V no load)

Stall Torque (4.8V): 17.5oz /in (1kg/cm)
Operating voltage: 3.0V~7.2V
Temperature range: -30 to +60
Dead band width: 7usec

Reason for selection:

1. Small
2. Easier to be programmed
3. The rotation degree can be controlled
4. Hight accuracy
5. The circuit is also easier to be constructed.

The application for this component would be discussed in Chapter 3.

Section 2.6: Drive integrated circuit

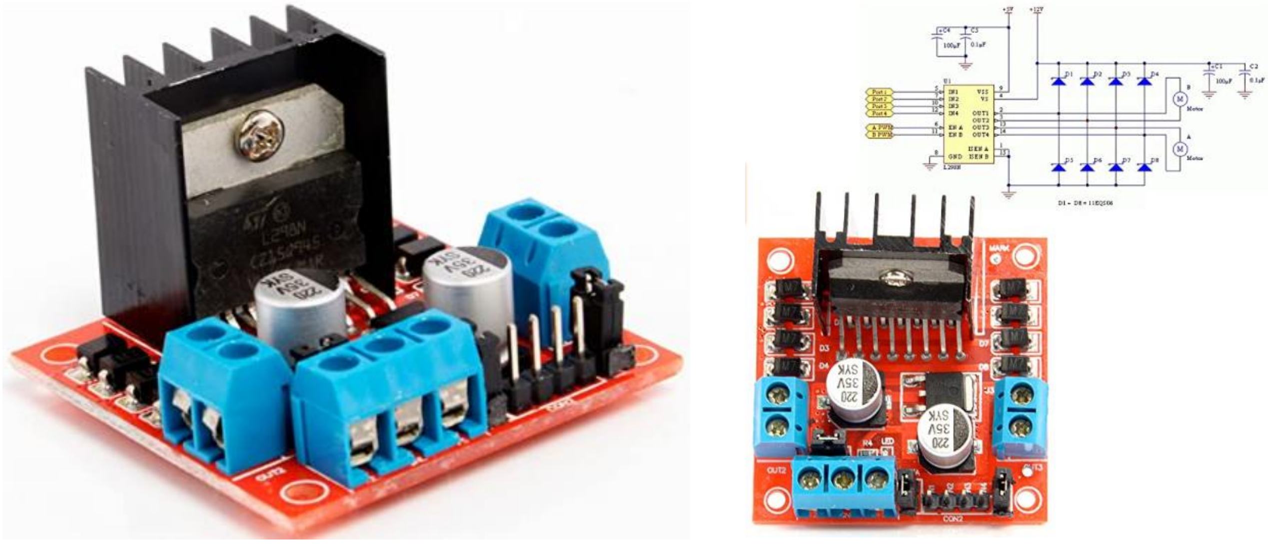


Figure 2.6: L298N DC stepper motor controller drive [20]

Feature:

1. The output pins have connections with tongs and OUT1, OUT2 and OUT3 of the first motor, OUT4 marked for the second motor
2. Ensure sufficient voltage and current for your engine at pins VCC and GND
3. This module integrated with one internal 5V power supply, meaning when the driving voltage is 7V-35V, it makes 5V of energy board power supply logically; when 5V of energy board power on, please not input voltage to +5V power supply in the interface but you can lead 5V voltage for external usage
4. Built-78M05, the drive power receiving power, but when the drive power exceeds 12V, use the power of 5V external power

Reasons for selection

1. L298N as main driver chip makes strong driving ability, low heating, strong anti-interference

2. The module is driven by Dual-channel H-bridge, it can drive two motors simultaneously that creates higher working efficiency
3. The speed and direction can be easily controlled using Arduino
4. Large-capacity filter capacitors and diode with freewheeling protection function, increasing reliability
5. This module can use built-in 78M05 for electric work via a driving power supply part
6. It can be used to control the motor mentioned in figure 2.5
7. It can be used to provide a constant power supply

The specific application for this component would be discussed in Chapter 3.

Section 2.7: Power supply

The whole system is built on a stripboard, the system is required to have a stable power supply and the whole stripboard should be portable to achieve the whole system can be used anywhere and anytime.Thus, a battery pack which can be fixed at the back of the stripboard is considered as a stable and portable power supply.[21]

Figure 2.7: 6V battery pack [21]

Reasons for selection:

1. High-quality battery holder case with wire leads for easy soldering and connecting.
2. Battery Type: 4x 1.5V AA battery；Battery Cable: 15cm; Material: Durable plastic and metal.
3. Each battery holder comes with a case cover for keeping the battery in place.
4. Great items for keeping your standard or rechargeable batteries together and safe.

This component provides a stable 5v voltages for the microcontroller mentioned in Section 2.3, it has to be connected with the drive mentioned in section 2.6 first.

The specific application for this component would be discussed in Chapter 3.

Chapter 3: Project design concept and analysis

In this chapter, three project design concepts would be illustrated. The concept 1 and concept 3 have a similar idea, they don't have to modify the inside of the handheld controller, they are only required to build mechanical components on the shell of the handheld controller. The concept 2 needs to take the handheld controller apart and modify the circuit inside the controller.

The concepts would be introduced detailly, advantages and disadvantages of them would be listed respectively. The concept 3 was selected as the final implementation plan, this part is illustrated in section 3.4.

Section 3.1: Project design concept 1

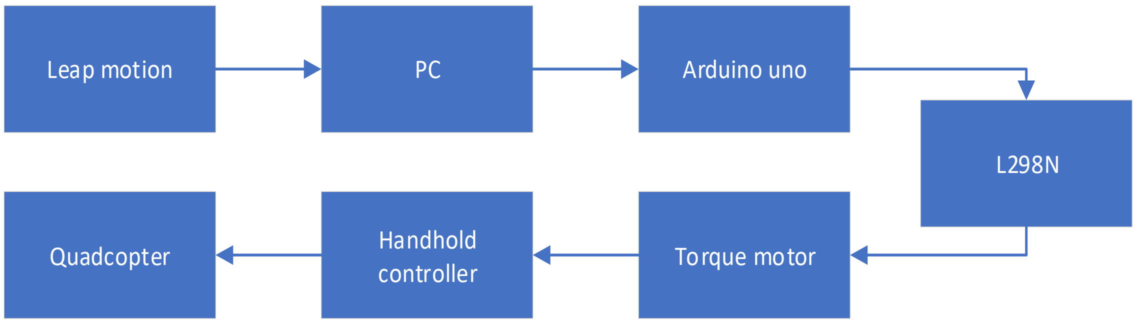
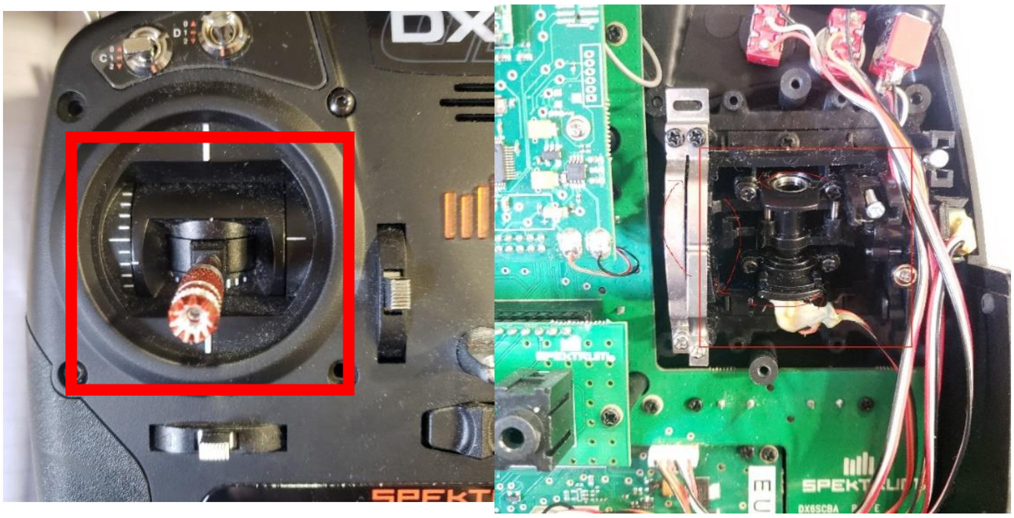


Figure 3.1: Project design concept 1

In the design concept 1, the leap motion is connected to PC first, then the signal and data from the Leap motion should be collected and programmed by using a software called Processing 3 (it will be introduced in section 5.3) on PC, after that the Arduino IDE on PC would transmit the signal to Arduino board.

The L298N motor drive would be connected to Arduino, then the signal from Arduino should make the L298N be able to control the rotation of the motor, the method of the motor to control the handhold controller will be shown in Figure 3.11.

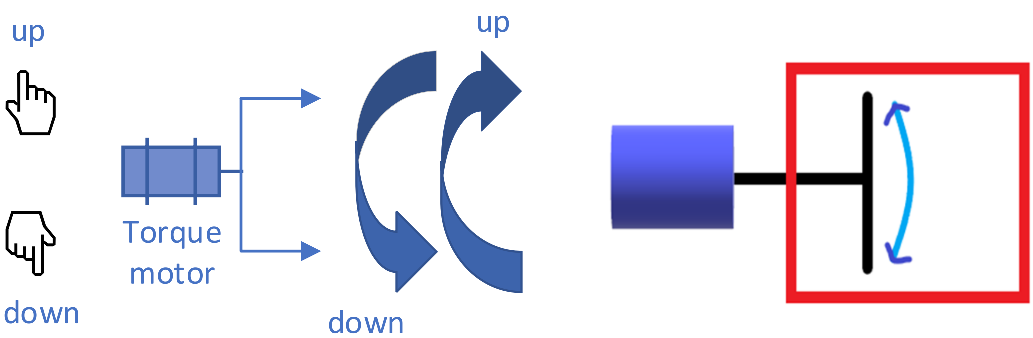
Figure 3.11: Handhold controller and its Internal structure



In concept 1, the red part should be added a 5V torque motor (mentioned in Section 2.5), the gear for the 5V torque motor should be connected to the joystick, therefore when the motor is rotating, the gear can pull or push the mechanical part to control the throttle.

The motor should be controlled by the L298N drive (mentioned in Section 2.6) which is connected to the Arduino board. As mentioned the Arduino is connected to a gesture sensor and the Arduino could receive the gesture signal from the Leap motion, therefore if the user's hands move up, the motor should push the joystick up, if the user's hand move low, the motor should pull the joystick down, thus, the throttle joystick of the handheld controller can be controlled by hand gestures.

Figure 3.12: control concept



The figure showed if the hands move up, the torque motor (mentioned in Section 2.5) rotates up, the throttle joystick will be pulled high by the gear of the motor, if the hands move down, the torque motor (mentioned in

Section 2.5) rotates down, the throttle joystick will be pulled low by the gear of the motor.
The circuit of the L298n and Arduino played the main role to control the 5V torque motor.

The detailed circuit design and specifications are shown below

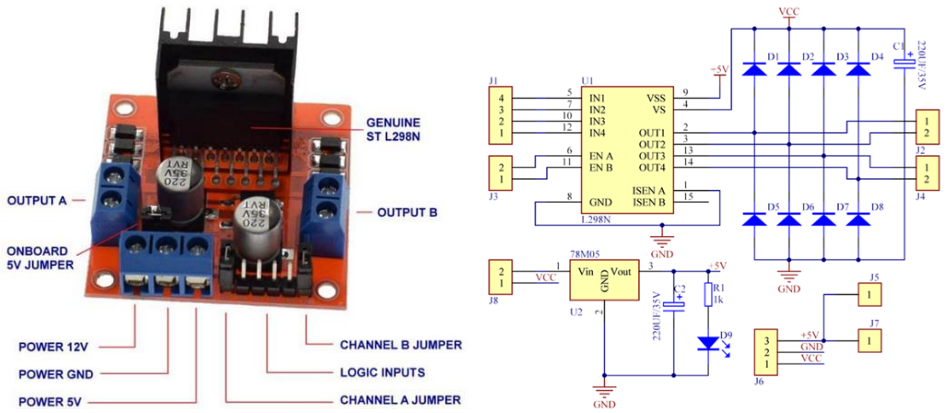


Figure 3.13: L298n stepper motor drive board and circuit [20]

Specifications of L298N Motor Driver

a. The module will allow users to control the speed and direction of two DC motors.

b. It can control motors that operate between 5 to 35V and up to 2A.

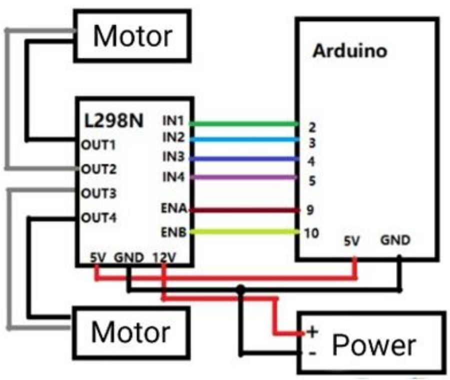
c. The module has an onboard regulator which helps in giving the output of 5V.

d. The module can be powered from 5 to 35V from Arduino or external power supply. It is recommended to always use the external voltage supply.

e. It can also control a stepper motor.

f. It is inexpensive and perfect for robotic projects.

In order to control the direction of rotation of the motor, L298n has to be connected to the Kona328 board (Arduino Uno).



direction	IN1	IN2
positive	high	low
negative	low	high
stop	low	low
stop	low	low

Figure 3.14: circuit of L298n with Arduino

The 6V battery pack should be connected to the 12V pin of the L298N, the 5V pin of the L298N should be connected to 5V pin of Arduino board, thus the regulator of the L298N would provide a constant 5V power supply to Arduino board.
The two pins of the motor would be connected to out1/out2, the IN1/2 pins should be connected to the pins 1/2 on Arduino board for signal transmission.
Once the circuit is constructed, the L298N should be able to control the torque motor rotation by correct codes in Arduino board.
As figure 3.13 showed, if the voltage of IN1 is high, and the voltage of IN2 is low, the Motor rotation direction would be positive. If the voltage of IN1 is low, and the voltage of IN2 is high, the Motor rotation direction would be negative. If the voltages are the same in IN1 and IN2, the motor will stop rotating.

Example codes

```
void loop() {
  //forward
  digitalWrite(input1,HIGH);  //high
  digitalWrite(input2,LOW);  //low
  delay(3000);   //time
}
```

As shown in the example codes, IN1 and IN2 (mentioned in figure 3.14) are defined as input1 and input2, delay(3000) means the motor will rotate positive for 3000ms.
The concept 1 has been constructed on a stripboard to test performance, it can be seen in Figure 3.15.

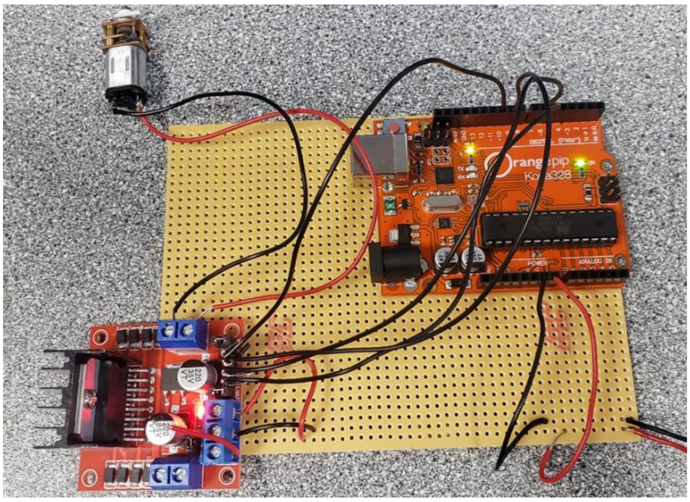


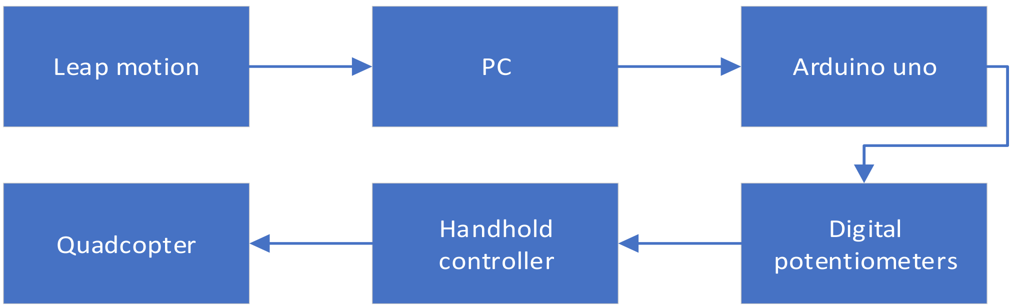
Figure 3.15: Basic torque motor circuit on stripboard

Firstly, the system operation needs to be tested without leap motion connecting to make sure the gear of the torque motor is able to pull or push the throttle joystick smoothly and accurately. If the signal transmission between the leap motion and Arduino is done, but the motor gear can't pull

or push the joystick on handhold controller properly, it will waste a lot of time and budget.
The torque motor is fixed on the handhold controller, an iron wire (doing the gear job) is used to connect the motor rod with the joystick.
The advantages and disadvantages of the concept 1 would be listed in following table.

Advantages	Disadvantages
1. Have enough torque to pull and push the joystick on handhold controller. 2. Small, easy to be fixed on the handhold controller 3. Can be fixed on the outside of the handhold controller, no need to open up the expensive DX8 controller.	1. There is no suitable gear for the torque motor, the wire wound on it is easy to slip. 2. The surface of the motor is metal which is hard to be fixed on the plastic case of the DX8 controller. 3. Although the rotation speed of the torque motor is already very slow, it is still too fast for the joystick. 4. The rotation angle cannot be controlled, the throttle can only be pushed to the largest or smallest, which is dangerous for the Quadcopter.

Section 3.2: Project design concept 2



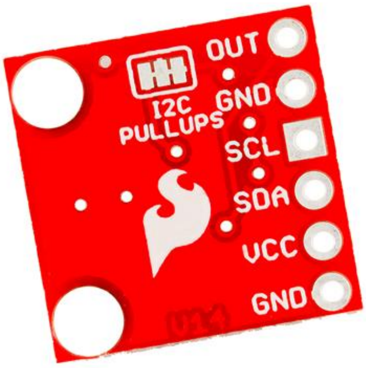
In the design concept 2, the leap motion should be connected to PC first, then the signal should be handled and transmitted by using a programming Software (processing 3, in section 5.3) on PC, the signal then will be transmitted to Arduino board, after that the Arduino would send the signal to the digital potentiometer circuit (can be seen in figure 3.22).

The digital potentiometer circuit need to be connected to handhold controller by modifying the wires inside the DX8 controller, therefore the joysticks now are replaced by “voltage signal” from the digital potentiometer, and the "voltage signal" would be able to send the control commands to the Quadcopter.

Digital potentiometers will be used in this project, because this kind of potentiometers can be programmed, the voltage signal can be set flexibly by using Arduino board, while the mechanical potentiometers could not.

A digital potentiometer (also called a resistive digital-to-analogue converter, or informally a digipot) is a digitally-controlled electronic component that mimics the analogue functions of a potentiometer. It is often used for trimming and scaling analogue signals by microcontrollers. So the Digital potentiometers here is to simulate joystick signal and transmit it to the quadcopter. Digital potentiometers are a new type of CMOS digital alternative to traditional mechanical potentiometers (analogue potentiometers) that simulate mixed-signal processing integrated circuits. The digital potentiometer is controlled by a digital input to generate an analogue-digital potentiometer. The numerical value is used to adjust the resistance value. It has flexible, high precision and non-polar output. Output. Depending on the digital potentiometer, the tap current can be shorted from a few hundred microamps to a few milliamps. low noise, not easy to foul, anti-vibration, anti-interference, it has a small size, long life and other significant advantages. [16]

Figure 3.21: digital potentiometer MCP4725



Pins of MCP4725	Use
OUT	Outputs Analog Voltage
GND	GND for Output
SCL	I2C Serial Clock line
SDA	I2C Serial Data line
VCC	Input Reference Voltage 5V or 3.3V
GND	GND for input

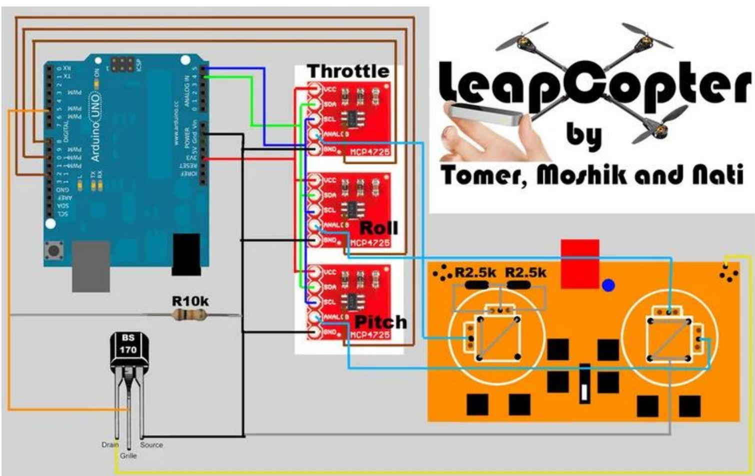


Figure 3.22: example circuit of the whole system [22]

1. Connect the VCC output from each of the 3 A/Ds (analogue-to-digital converter) together to the Arduino's 3V3 input (The RED wires on the circuit picture)
2. Connect the SDA output from each of the 3 A/Ds together to the Arduino's analogue 4 input (The GREEN wires on the circuit picture)
3. Connect the CSL output from each of the 3 A/Ds together to the Arduino's analogue 5 input (The BLUE wires on the circuit picture)
4. Connect the ANALOG output from each of the 3 A/Ds and choose one for the throttle, one for the roll, and one for the pitch. (The light blue wires on the circuit picture)
5. Connect the GND output from each of the 3 A/Ds together to the Arduino's GND input (The BLACK wires on the circuit picture)
6. Connect a wire to the lower-left pin in each of the main chips in the 3 A/Ds:

- in the throttle, A/D connect the wire to the Arduino's digital 8 (The brown wires on the circuit picture)
- in the roll A/D connect the wire to the Arduino's digital 10 (The brown wires on the circuit picture)
- in the pitch, A/D connect the wire to the Arduino's digital 12 (The brown wires on the circuit picture)

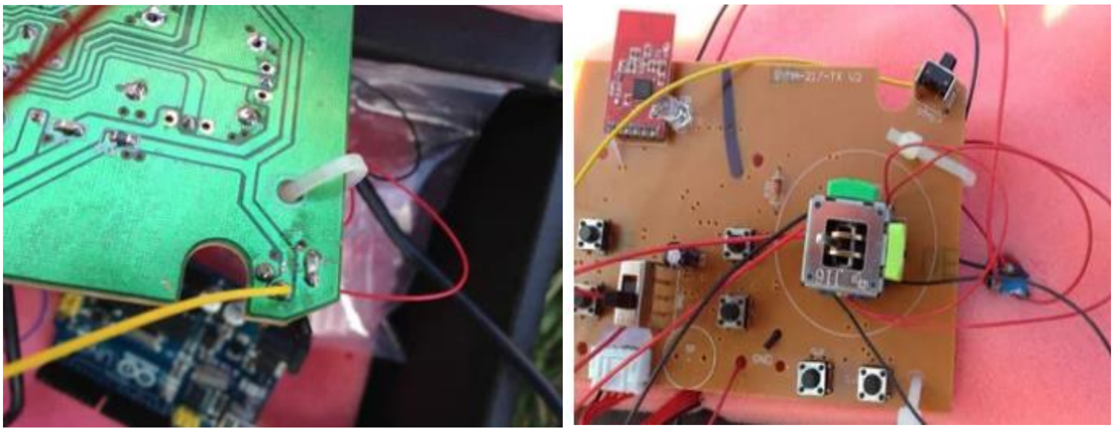


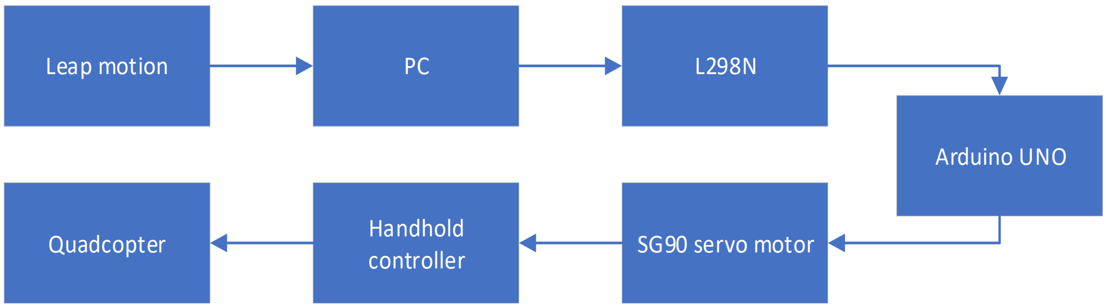
Figure 3.23: motherboard of the handheld controller

As the figure shows, the original circuits need to be modified to be connected with the potentiometer circuit.

The advantages and disadvantages of the concept 2 would be listed in following table.

Advantages	Risks and disadvantages
1. high control accuracy 2. Fast response speed 3. Take up less space	1. The more complex circuit structure 2. Difficult conceptual understanding 3. Easy to damage an expensive controller, once the controller is broken, the entire project cannot continue constructing. 4. The internal circuit of the controller is complicated, there is very low fault tolerance. 5. expensive, many of the Mcp4725 are broken

Section 3.3: Project design concept 3



In the design concept 3, the leap motion should be connected to PC first, then the signal should be handled and transmitted by using programming Software on PC (processing 3, in section 5.3). After that, the signal is sent to the Arduino board. The SG90 servo motor (can be seen in figure 3.31) is connected to Arduino, therefore the codes in Arduino should be able to control the rotation of the SG90 servo motor. The gear of the SG90 servo motor should be able to pull or push the throttle joystick on the handheld controller. Thus, the hand gesture could control the Quadcopter to fly high or low.

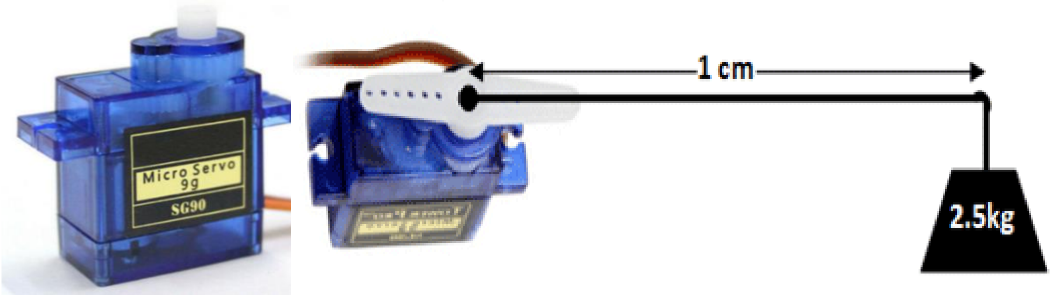
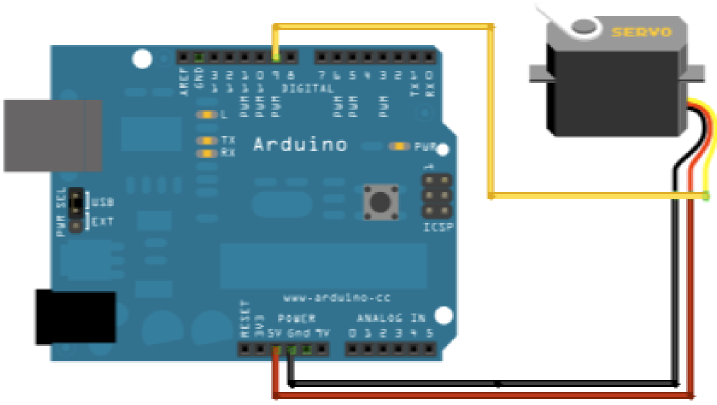


Figure 3.31: SG90 servo motor [23]

The method of the motor gear to control the joystick on the handheld controller can be seen in Figure 3.11 and 3.12. This part is very similar to the design concept 1.
L298n motor drive is not required to control the rotation of SG90 servo motor, however, the L298N motor drive here is still can be used as a Voltage regulator, the power supply is a 6V battery pack which is connected to L298N 12V pin, therefore the L298N component is on duty to provide a constant 5V voltage to Arduino board. This part can be seen in Figure 4.12.

Figure 3.32: Example circuit [23]



Firstly, the system operation needs to be tested without leap motion connecting to make sure the SG90 motor is able to control the throttle smoothly and accurately. The example circuit is shown in figure3.32. If the signal transmission between the leap motion and Arduino is done, but the motor gear could not pull or push the joystick on the handhold controller properly, it will waste a lot of time and budget. The SG90 servo motor is fixed on the handhold controller, original gear which is suitable enough for servo motor is used to connect the motor rod with the joystick.

The advantages and disadvantages of the concept 3 would be listed in the following table.

Advantages	Disadvantages
1. Have enough torque to pull and push the joystick on handhold controller. 2. The plastic shell is easy to be fixed on the handhold controller 3. Can be fixed on the outside of the handhold controller, no need to open up the expensive DX8 controller. 4. High accuracy and fast response speed, the rotation degree can almost follow the height of the user's hand. 5. Simple circuit and concept required. 6. Cheap and fewer components required	1. Need at least four to achieve full functions, they would require a large space to be fixed on the handhold controller.

Section 3.4: Project design concept analysis

After comparison,
The project design concept 3 shows a better performance, it doesn't need to modify wires and circuits inside the handheld controller, therefore it won't damage the expensive transmitter. Concept 3 is also easier to be constructed compared to concept 1 and 2. The codes of Concept 3 is also easier to be developed. Concept 3 is a relatively good design that combined mechanical components and embedded programming.
This concept almost includes both advantages of concept 1 and concept 2 and only has a few disadvantages.
Thus, concept 3 would be selected as a final implementation plan, all the detailed work could be seen in Chapter 4.

Chapter 4: Project design concept 3 implementation

In this chapter, the hardware implementation of Concept 3 will be illustrated in detail. The hardware implementation includes power supply circuit design, L298n circuit design, SG90 servo motor circuit design and whole system construction.

Section 4.1: Hardware design

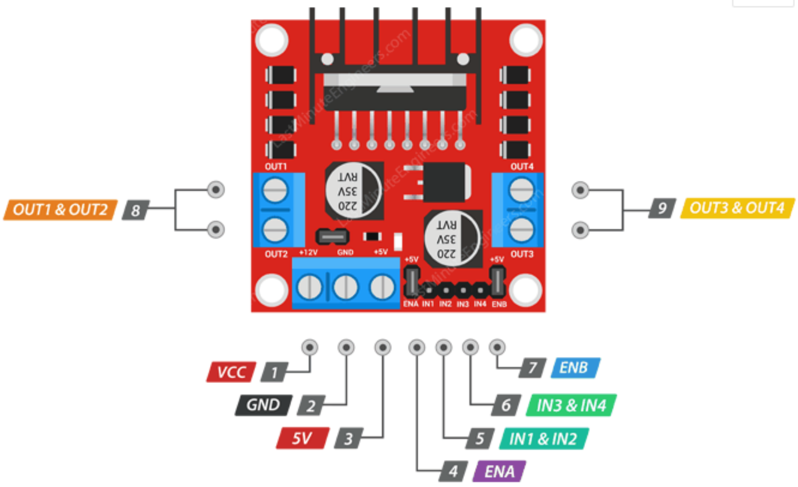


Figure 4.11: L298N circuit [24]

In concept 3, the L298N component is only used as a voltage stabilizer, because the rotation of SG90 motor can be directly controlled by Arduino, so only the VCC, GND, 5V pins are needed to provide a constant 5V power to Arduino.

VCC pin supplies power for the board. It can be anywhere between 5 to 35V.

GND is a common ground pin.

5V pin supplies power for the switching logic circuitry inside the L298N IC. If the 5V-EN jumper is in place, this pin acts as output and can be used to power up your Arduino. If the 5V-EN jumper is removed, you need to connect it to the 5V pin on Arduino.

The battery pack mentioned in section 2.7 provides a 6V voltage, the red wire(positive) of the battery pack should be connected to the VCC pin on L298N, the black wire and GND pin should be connected to the ground. The 5V pin on L298N should be connected to the 5V pin on Arduino.

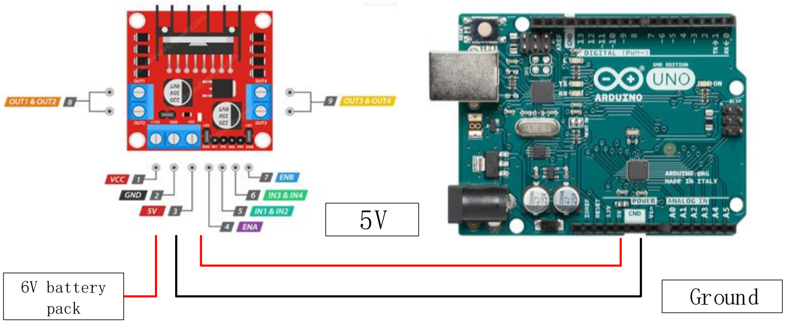


Figure 4.12: Power supply to Arduino

After the circuit constructed, the Arduino now should have a stable 5V power supply, no matter how large the voltage provided by battery pack is, the power supply will always be 5V, which meet the required voltage limit for SG90 servo motor.
Then the SG90 servo should be connected to Arduino. The circuit is shown in figure 4.13.

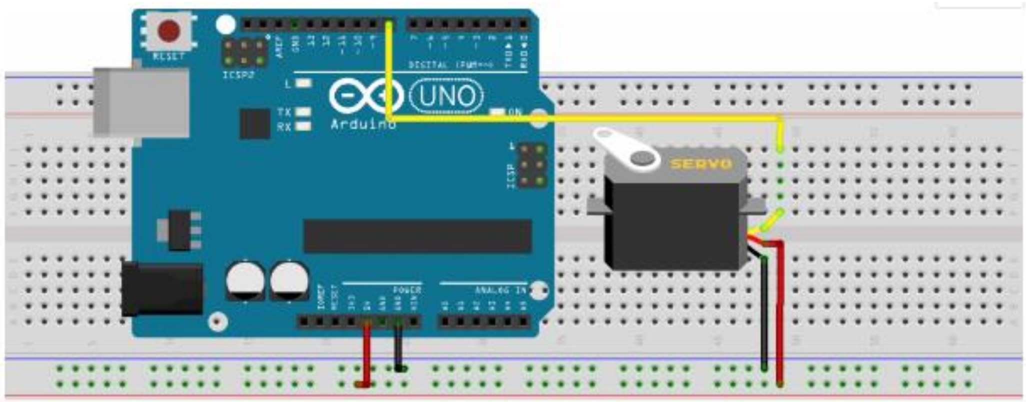


Figure 4.13: SG90 connection with Arduino [23]

As the Figure shows, the circuit for connection between the SG90 servo motor and Arduino is quite easy.
Servo motors generally have three pins/wires, this includes the VCC, GND, and the Signal pin. The Signal pin is the one used to feed the control signal from the microcontroller to the servo, to get the servo to rotate to a particular angle. The yellow wire in figure 4.13 is the signal transmission wire, it could be connected to any digital pin on Arduino because it is a PWM pin (Pulse width modulation is a powerful technique for controlling analogue circuits with a microprocessor's digital outputs. PWM is employed in a wide variety of applications, ranging from measurement and communications to power control and conversion.). Servo directions are sent from the microcontroller to the servo motor as PWM pulses.

After the connection processes are done, the test codes should be implemented to test the working performance of the SG90 servo motor, this process could be seen in Section 5.1.

The whole circuit now is shown in figure 4.14.

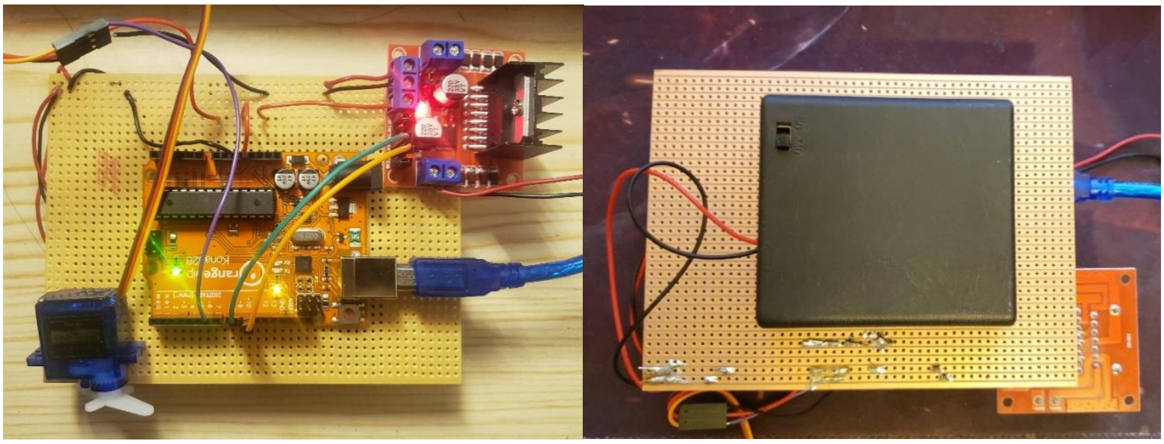


Figure 4.14: whole system circuit of SG90-Arduino-L298N

Chapter 5: Software design

In this chapter, the software implementation of Concept 3 will be illustrated in detail. The software implementation includes SG90 servo motor test, leap motion test, communication between leap motion and Arduino, and final system communication test.

The software design is the most significant part in the whole project, the signals from the leap motion must be transmitted to Arduino through PC, in this way, users' hand gestures could control the rotation of SG90 servo motor, otherwise, the hardware part would be useless without a valid code.

Section 5.1: SG90 servo motor test

The code for testing is quite simple thanks to the servo.h library developed by the Arduino team to facilitate the use of servo motors in Arduino projects. The library makes it easy to turn the servo at different angles using a single command. The library comes pre-installed in the Arduino IDE removing the need for users to download and install. [25]

The example codes are shown below, Figure 5.11 and Figure 5.12

```
Servo servo;
int angle = 10;

void setup() {
  servo.attach(8);
  servo.write(angle);
}
```

Figure 5.11: example code 1

This part defined the digital pin 8 as the signal pin and set the rotation degree to 10.

```
void loop()
{
 // scan from 0 to 180 degrees
  for(angle = 10; angle < 180; angle++)
  {
    servo.write(angle);
    delay(15);
  }
  // now scan back from 180 to 0 degrees
  for(angle = 180; angle > 10; angle--)
  {
    servo.write(angle);
    delay(15);
  }
}
```

Figure 5.12: example code 2

In the loop, the code means, servo motor rotates from degree 10 to 180, after 15ms delay, the servo motor rotates back to 10.

This code is for testing whether the SG90 servo motor circuit is working or not, if it is working, the servo motor should start turning as above illustrated.

Section 5.2: leap motion test

Figure 5.21: leap motion and drive

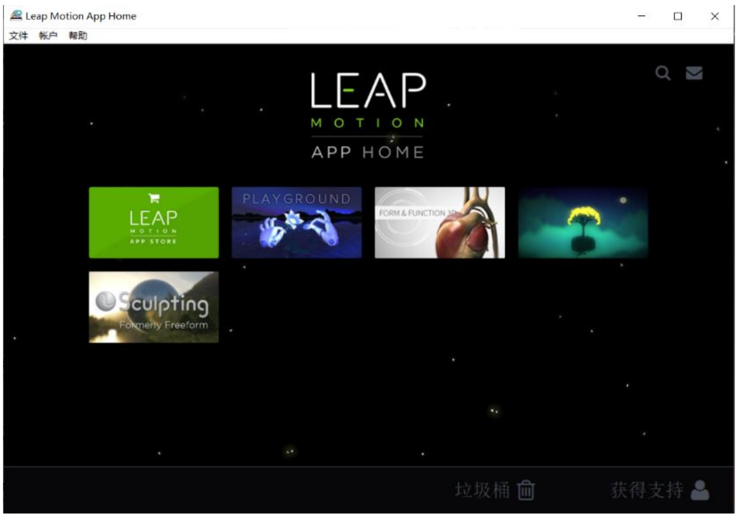


Figure 5.22: Leap motion APP home

Firstly, the drive for leap motion needs to be download from the official website. After downloading, two software called "Leap Motion control panel" and "Leap Motion App home" should be able to find on the desktop. There is already some software in "Leap Motion App home", users can use the software to test the leap Motion is working or not. At the first time the software may not run properly, the general solution is:
Go to the directory of the leap motion, find a File named "LeapSvc64", click to install it as a 64-bit version.

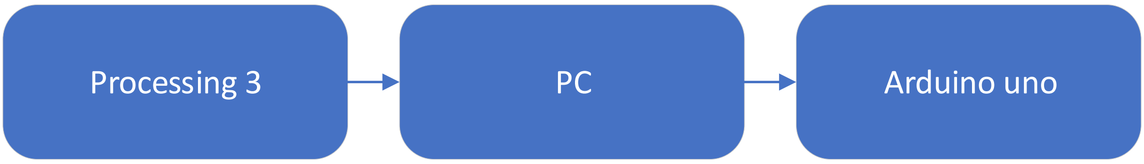
Section 5.3: "Processing 3" IDE with Leap Motion

Figure 5.31: Processing 3 IDE [26]

Processing is a flexible software sketchbook and a language for learning how to code within the context of the visual arts. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology.

Free to download and open source
Interactive programs with 2D, 3D or PDF output
OpenGL integration for accelerated 2D and 3D
For GNU/Linux, Mac OS X, and Windows. [26]

Processing 3 here is for handling the signal from leap motion on PC, then transmitted the signal to Arduino board. C programming language would be used to develop the code.
Communications between Arduino and leap motion can be simply shown below.



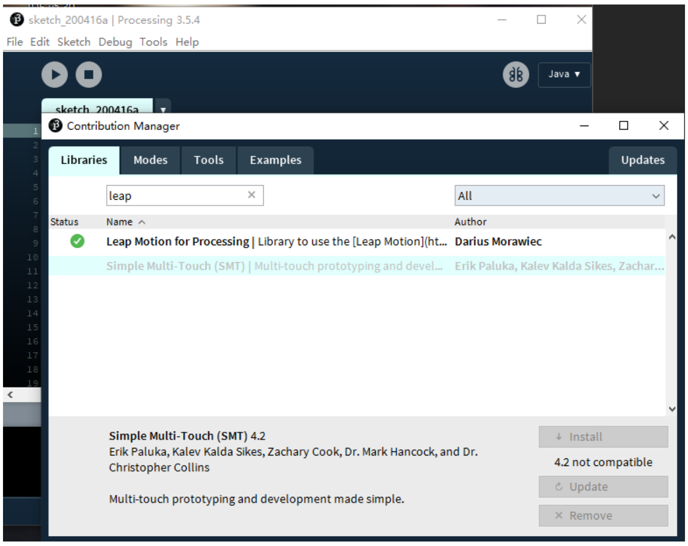


Figure 5.32: Leap Motion library in processing 3

For processing 3, there is already an example code for testing the connection between the Processing 3 IDE and Leap motion, firstly the library needs to be added, then the code can be found in Example codes.

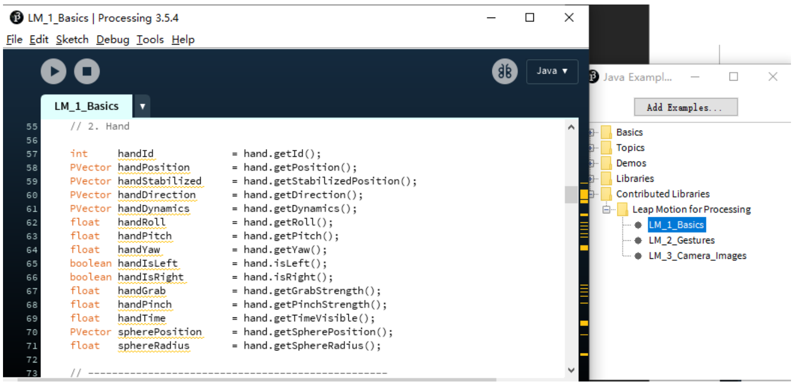


Figure 5.33: Leap Motion library in processing 3

As shown in Figure 5.33, LM_1_Basics provides the example code for testing the Leap Motion, there are also lots of useful codes for learning to develop the code later.
Connected the Leap Motion to PC, then compile and run the example code.

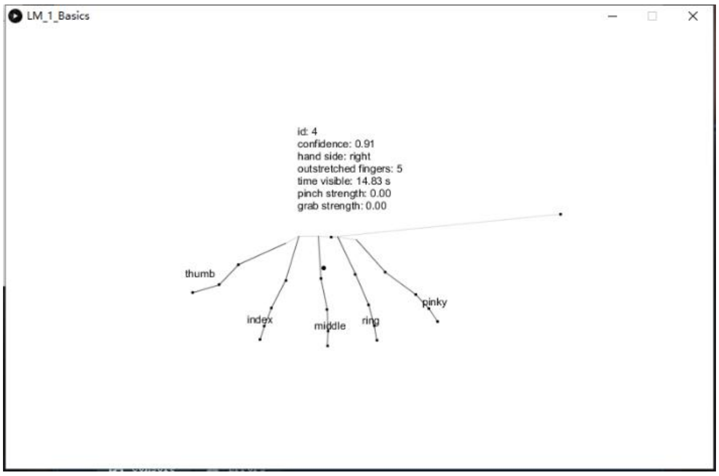


Figure 5.34: The example gesture recognition program

As shown in Figure 5.34, if the leap motion and the codes are working properly, the structure of hand would appear in the window.
The coordinates of the hands are also shown in the window, the structure of hand in the window will change with the movement of users' hand at the same time. Leap Motion can detect all fingers, wrist angles, palm, pitch, yaw, roll and waving of hands these all the process that can be used for controlling servo motors with a leap motion controller with Arduino. With the help of the cameras motion detector sensor captures the images and process them to see which gesture exactly moves. It can only work if gestures moving in a limit. Limit of motion control sensor is from 1 inch to 2*2 feet.
After all the test parts are done, the communication between the leap motion and Arduino board is ready to start, the code for leap motion will be developed on Processing 3 first, the process would be shown in Section 5.4.

Section 5.4: Communication code

In this section, the codes for Leap motion (Processing 3) and SG90 servo motor (Arduino) would be developed.

Firstly, the leap motion and Arduino need to be connected to the same PC by USB cables.

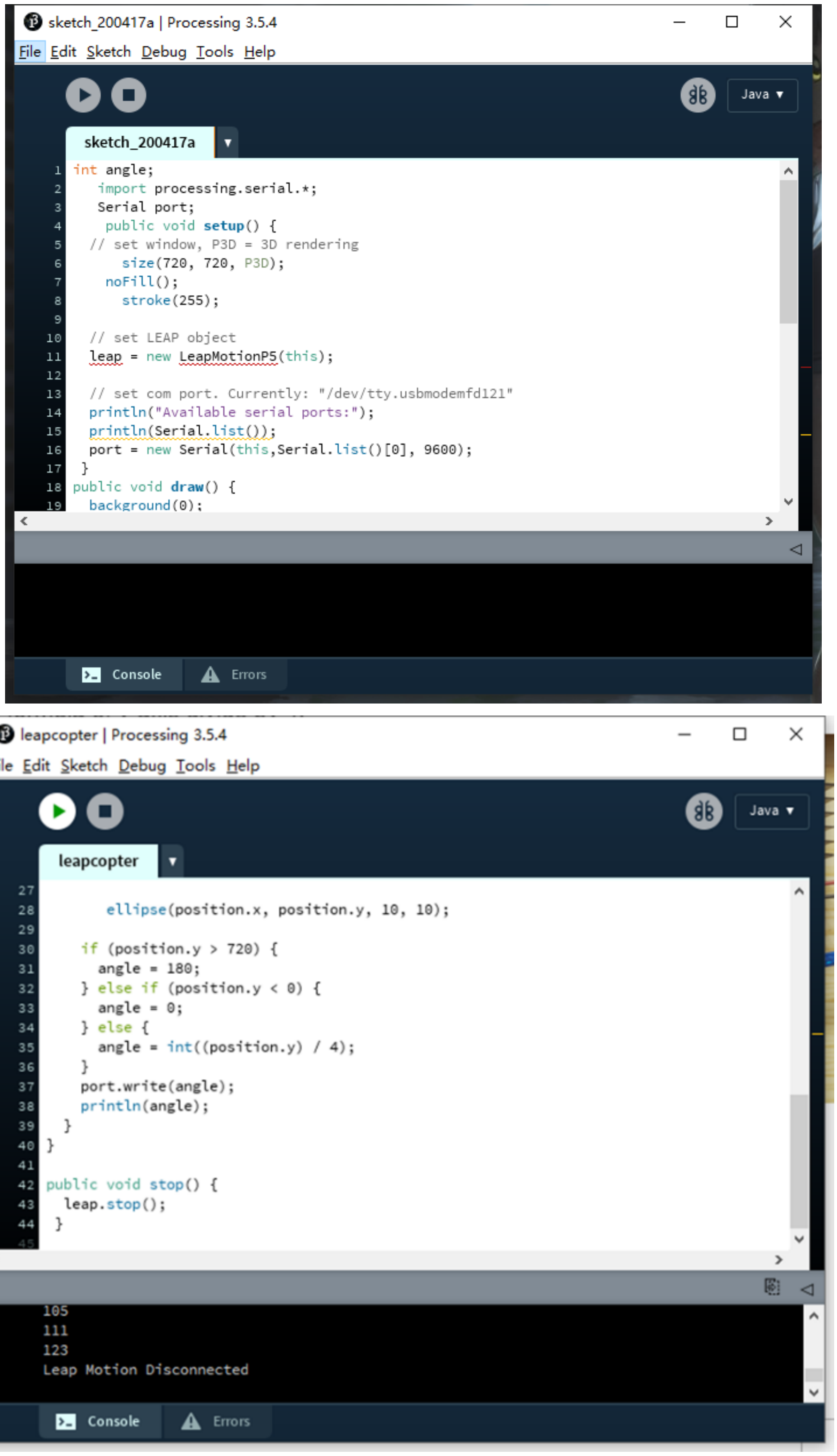


Figure 5.41: The example code on processing 3

The codes for figure 5.41 are explaned below.
First of all, a library called LeapmotionP5 is required for Processing 3 IDE, the library is provided in programme files.
Creating a class of leap and making an integer of name angle which is used further for storing the angles of the servo motor and transmitting it to Arduino. The code" import processing.serial.*;" used to communicate the processing software commands and sends it to Arduino via the serial port.
At detection of gestures in motion sensor along Y-AXIS the particular value assigned to servo motor via leap motion controller then processing and Arduino interchanged the value with each other by serial-communication.
In setup when running the program in processing new windows open. The size of the new window is 720*720. In these windows, 5 fingers dots could be shown. If users wave their hand gestures after recognizing the movements of gestures their gestures shown in that window.

As mention above when users run the program, a new window open which is 720 height and width. This window is used for controlling the position of the servo with 3D hand recognition technology. If hands above the 720 size the servo angle move to 180 degrees. If the position of hand gestures is below zero the angle turns to 0 degrees. In between the windows, the angle has a formula of the y-axis divide by 4.

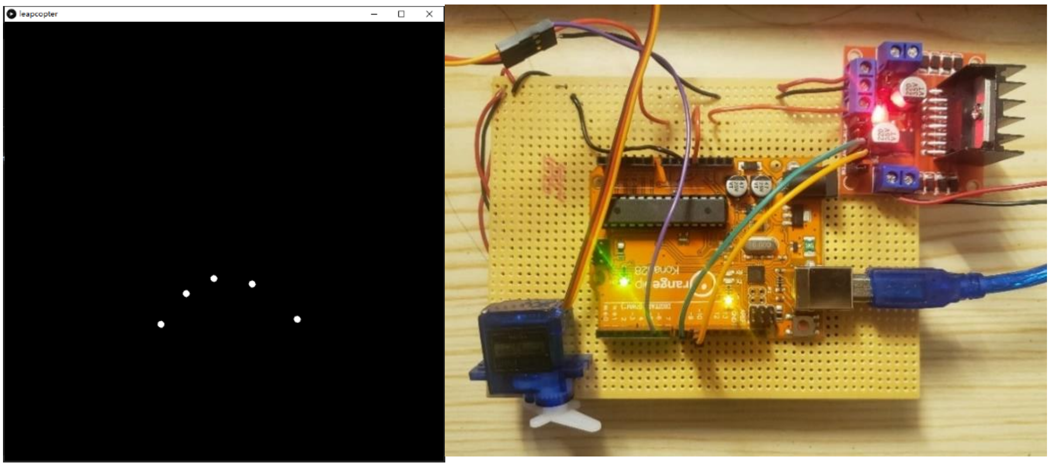


Figure 5.42: processing 3 window and servo circuit

```
#include <Servo.h>
Servo myServo;
//int handPos;
//int angle;
void setup() {
  myServo.attach(9);
  Serial.begin(9600);
  myServo.write(0);
}
void loop() {
  byte angle;

  if (Serial.available()) {
    // Read angle from Processing
    angle = Serial.read();
    Serial.println(angle);
    // If fingers in window, read servo angle
    myServo.write(angle);

  }
}
```

Figure 5.43: The example code on processing 3

In setup attach the signal wire of servo motor is connected to Pin 9, Communicate serially with serial. begin command with the baud rate of 9600. Baud rate can also be changed if needed.

Read serial if available means read the angle from processing or whatever values coming from processing via hand detection. Saved in variable named angle and write it to the servo motor. Remember that values are within 0 to 720 range. whatever gestures come to leap motion controller the processing tracks and send data to Arduino that moved servo motor at some special angles.

One thing is important, the processing 3 IDE and Arduino IDE must be opened at the same time, if only one of them is activated, the signals can not be transmitted due to no transmitter/receiver.

Once the codes compile successfully, the system should be able to use the leap-motion to control the rotation of the SG90 servo motor properly.

Chapter 6: system evaluation

In this chapter, the development of hardware and software will be combined into a single system. The whole system is built on a single stripboard, therefore the system is portable and would be easy to be connected to the PC or handheld hold controller.
The performance and result of the whole system and the problems of the current system would be discussed.

Section 6.1: System mechanism

At the beginning of the project development, the system mechanics of how the whole control system is going to work out as planned. The section 6.1 was expected to get an summary of how the whole system is going to operate. Concerns for the project were addressed, including the system's flow, how the mechanics work and the ability to identify the hand gesture and control the rotation of SG90 motor.

The system's mechanic is planned to work according to the following points:

1. Detect hand gestures from the Leap Motion.
2. If gestures are detected, the coordinates of fingers will be read by processing 3 IDE, the structure of hands will be displayed on the window on PC.
3. If the hand is high, the signal (make the servo motor rotates from 0 -180 degree) will be sent to Arduino board. If the hand is low, the signal (make the servo motor rotates from 180 - 0 degree) will be sent to Arduino board.
4. The Arduino will receive the signals then send the rotation signal to the SG90 servo motor.
5. The SG90 motor starts to rotate, which will pull or push the throttle joystick on the handheld controller to low or high.
6. The Quadcopter then will fly high or low.

The flow chart of the whole process could be seen in Figure 6.11 below.

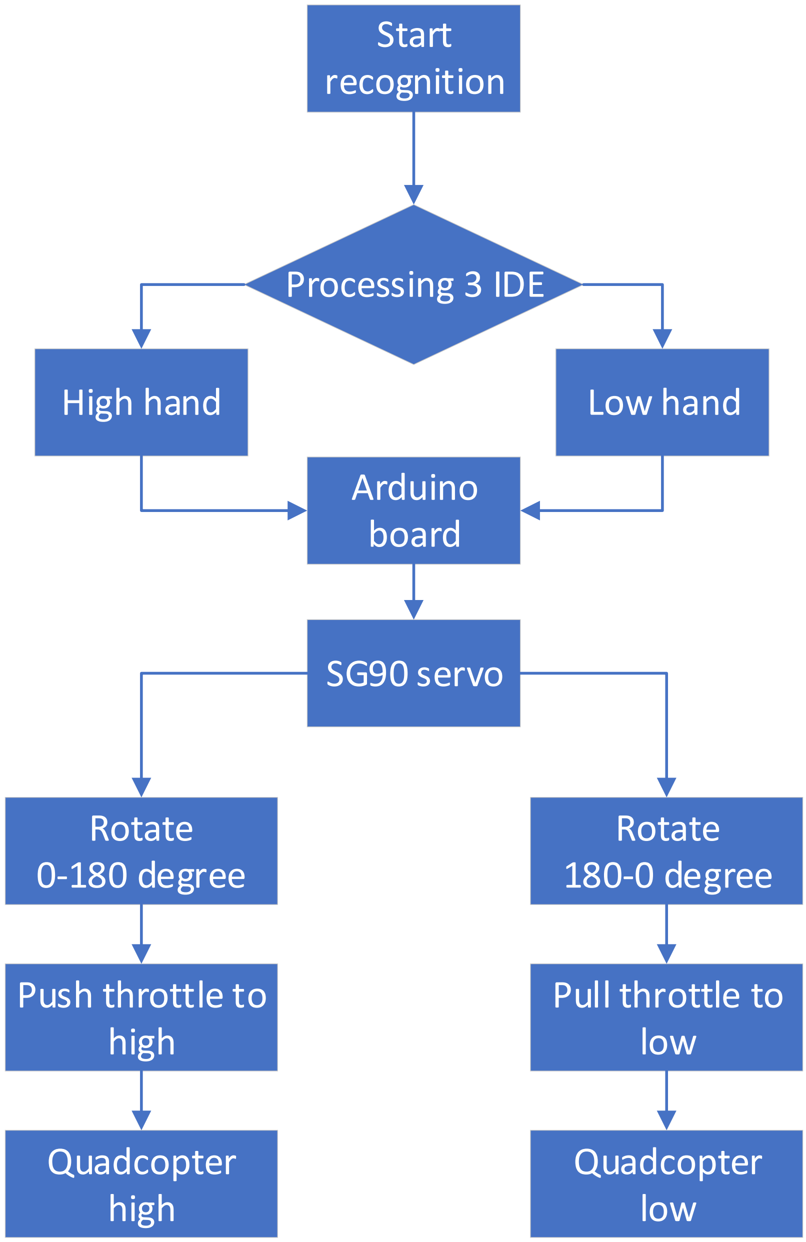


Figure 6.11: Flowchart of the Throttle control

The flow chart shows how the throttle joystick of the handheld controller is controlled by the Leap Motion gesture sensor.

Section 6.2: Integration of SG90 circuit board with a handheld controller

At first, a mechanical structure is considered to be 3D printed to connect the SG90 servo motor to the joystick to control movement. This design could be seen in Figure 6.23.

Since the signal transmission part was completed, the leap motion now is able to control the rotation of the SG90 servo motor.
Then only the mechanical structure between the gear of SG90 servo motor and the joystick of the handheld controller needs to be designed and constructed.

Due to the COVID-19 situation, the 3D print method is hard to be achieved, thus, there are two more methods are considered to connect the gear of SG90 servo motor and the joystick of the handheld controller.

Which method would be the most efficient still needs to be discussed.
The 3 methods are illustrated below.

1. Use Rubberband between the SG90 motor gear and the joystick of the handheld controller. The example could be seen in Figure 6.21.

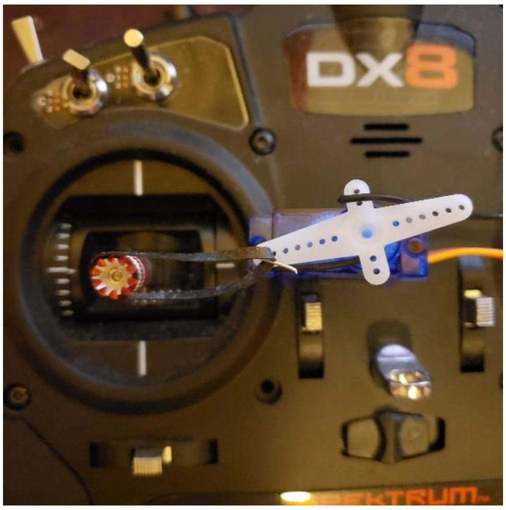


Figure 6.21: Rubber band connection

As the picture shows, both ends of the rubber band were fixed at the joystick and the gear of the SG90 servo motor respectively.
With this method, when the servo motor is rotating, the rubber band could pull the joystick to high and low.
However, due to the rubber band is elastic there is always a delay between the gear and joystick.
And the effective push-pull distance is too short, the joystick can only move a little.
This method may also cause the joystick is pulled to right.
There is a solution to make this method effective, for example, two SG90 servo motors can be fixed on the high and low position, two rubber bands are used to connect the two gears of servo motors to joystick. The two motors would always keep the same rotating direction, thus whether the joystick is up or down, the two SG90 motors are able to pull the joystick fast.

This solution can be seen in figure 6.212.

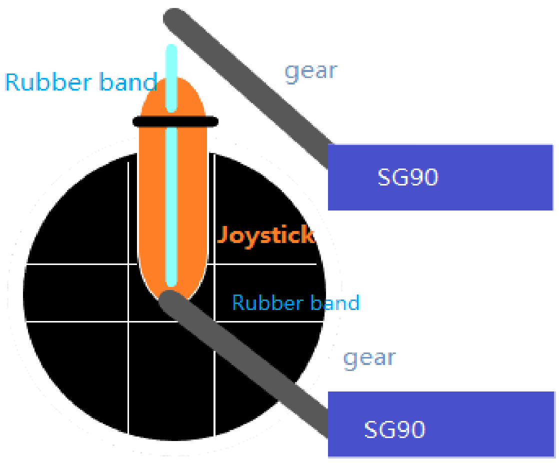


Figure 6.212: method 1 design

However, the shortcoming of method 1 is also obvious, two directions need 2 SG90 servo motors, this method would need many SG90 servo motors and take up a lot of space to fix them on the handheld controller.

2. Use iron wire between the SG90 motor gear and the joystick of a handheld controller.

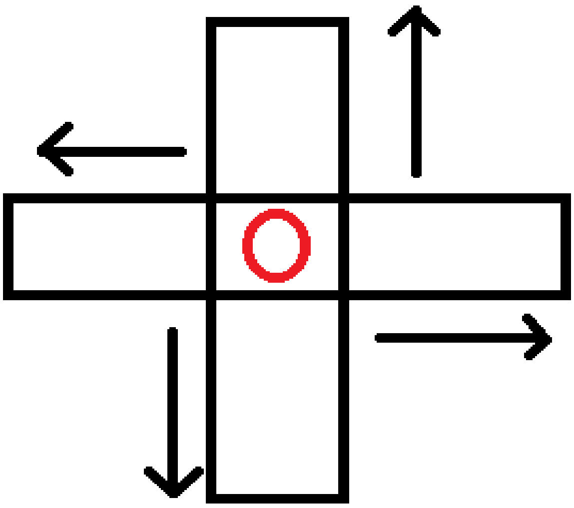
Figure 6.22: Iron wire connection

As the figure shows, both ends of the iron wire were fixed at the joystick and the gear of the SG90 servo motor respectively.

The position of the SG90 servo motor should be placed low, method 2 can be regarded as an improvement measure for method 1.

With this method, there is no delay between the gear and the joystick of the handheld controller because the iron wire provides a hard connection.
The effective push-pull distance can be modified to long or short by modifying the length of iron wire between the gear and joystick, therefore the throttle control limit can be modified to large or small.
This method has another advantage: if the throttle joystick reaches the highest limit, it will be pulled from high to low, this could be regarded as a protection for the Quadcopter.

3. Use mechanical structure which is 3D printed between the SG90 motor gear and the joystick of the handheld controller.



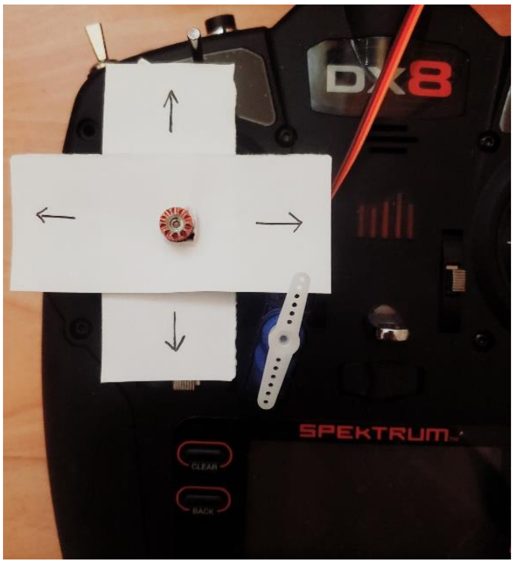


Figure 6.23: 3D printed mechanical part structure
Figure 6.24: Paper model for 3D printed mechanical structure

As figure 6.23 shown, the red circle in the middle is the throttle joystick. The two black squares (on the Horizontal axis and Vertical axis) around the joystick are the mechanical parts which can be 3D printed.

In order to explain the concept clearly, the model is made by paper can be seen in Figure 6.24. The joystick is in the centre of the mechanical structure, the gear of the SG90 servo motor should be connected to the edge of the mechanical structure.
This way, the mechanical structure can be pulled up/down, left/right, to control the joystick.

Method 3 can be regarded as an improvement measure for method 2.

In theory, method 3 can provide a stable and effective experience, it is also more convenient for subsequent upgrades, however, due to the current situation, this method is hard to be achieved, it could be regarded as an advanced plan in future.

To sum up, the performance of method 3 is better than method 2, method 2 is better than method 1.
Method 2 is current using, the performance is relatively stable and the structure is simple, it is easy to be achieved. There would further development for it later.

Section 6.3: Increasing the recognize efficiency of processing 3 (Leap motion)

As Section 5.4 mentioned, the leap motion will recognize the hand gesture then send the signal to Processing 3 IDE. If the gesture is more complex or more fingers are used, the transmission speed would get slow, there will also have a small delay between fingers and the motor gear.

For example, if the user waves his/her whole hand on the top of leap motion, there would be a small delay, when the hand up/down, the SG90 servo motor may start to rotate after 400 ms, the delay is short, however, for controlling the quadcopter it is very dangerous, because when users already try to make the quadcopter fly lower it may still keep fly upper for a while. Sometimes if the gesture changes too fast, the gear of SG90 servo motor may shake several times, this will also cause quadcopter unstable.

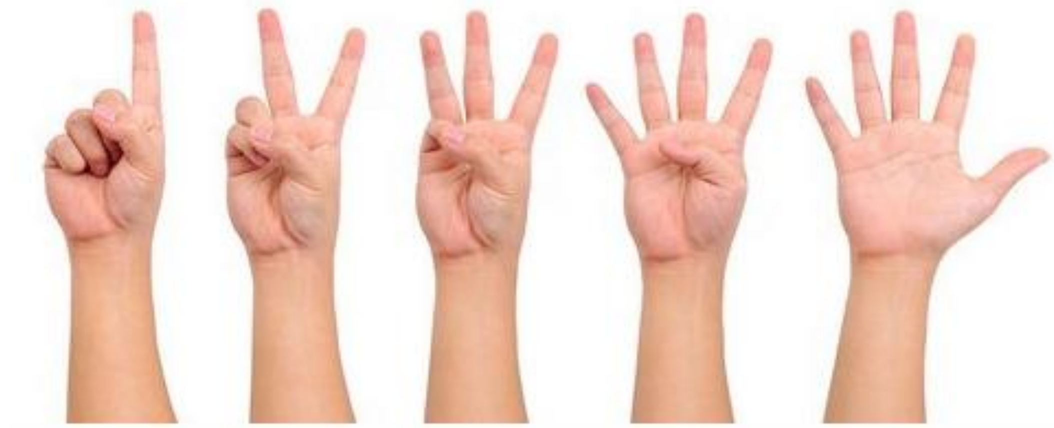


Figure 6.31: control precision when using 1-5 fingers [27]

Throttle control by Leap Motion		
Fingers	delay(ms)	status
1	50	stable
2	100	stable
3	200	normal
4	400	normal
5	450	unstable

Figure 6.32: Number of fingers used to control the quadcopter

As the experimental data showed, when using full hand to control the quadcopter, the signal transmission might be slow, SG90 servo motor rotates unstable, therefore the quadcopter will have an unstable fly experience.

By contrast, when using 1 finger to control, the SG90 servo motor almost has a real-time response, it rotates smoothly and stably, thus, the quadcopter would fly stably and follow every instruction from users' hand immediately. The user will also feel more relaxed when using fewer fingers to move on the top of the Leap Motion.

Therefore, it is recommended that users use fewer fingers on the top of the Leap motion to control the throttle of the quadcopter.

Section 6.4: System evaluation

In this section, the results of the final testing where the hardware and software sections are combined into a single system will be evaluated. The final system is shown in figure 6.41.

1. battery-powered and power plug are successful.
2. Leap Motion recognition successful.
3. Signal transmission successful.
4. SG90 servo motor rotates properly.
5. Throttle joystick moved properly.
6. Quadcopter flew up/down successfully.

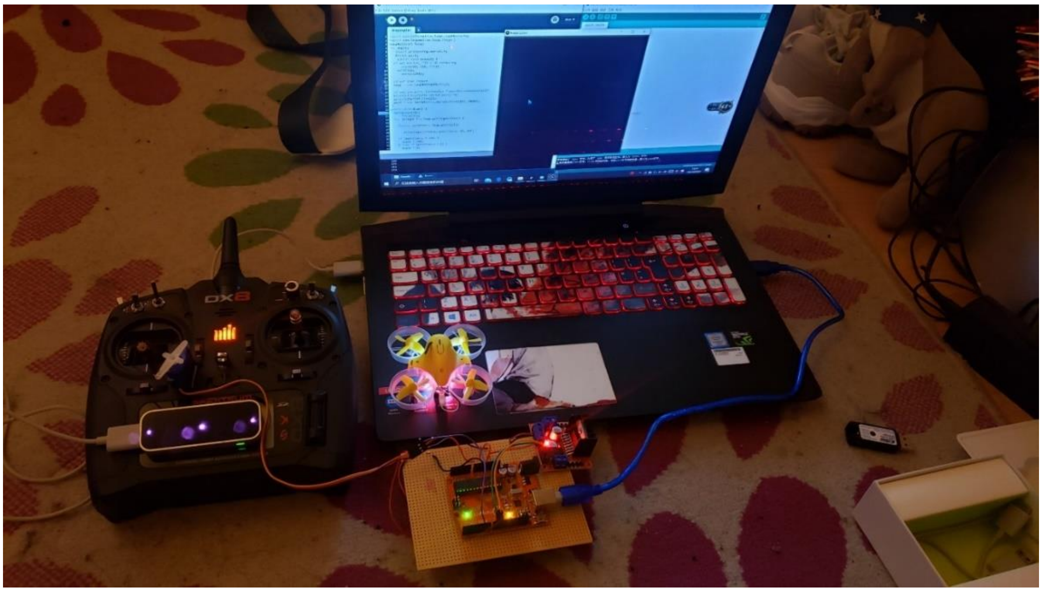


Figure 6.41: Whole system review

The leap motion throttle control was then tested 200 times in total to test its stability and accuracy.
The environment that was tested was in lightroom (100 times) and in a dark room (100 times). The results can be seen in the table below.

Experimentation results in a light environment (100 times)		
Throttle control	UP	DOWN
SUCCESS	97	100
FAILED	2	0
Success rate	97%	100%

Table 6.42: Experimentation results in a light environment (100 times)

Experimentation results in a dark environment (100 times)		
Throttle control	UP	DOWN
SUCCESS	95	97
FAILED	5	3
Success rate	95%	97%

Table 6.43: Experimentation results in a dark environment (100 times)

Figure 6.44: Video screenshot

As the table 6.42 shown, the Leap Motion recognition and the throttle control both have very high accuracy, sometimes my hand is on the top of the leap motion by accident, that’s the reason why the up/down failed.
In the dark environment, it may recognize wrong several times, but it still has high accuracy because Leap motion is using 3 cameras.

Overall, the throttle controlled by leap motion is successful, it has high accuracy and stability in both light and dark environment. The video screenshot for the Quadcopter flight can be seen in Figure 6.44.

Chapter 7: Future Work

At the moment, the several main parts (Hardware construction, software development, whole system test) of the project has been completed. However, due to the COVID-19 situation, a lot of further work cannot be achieved. There are still lots of room for further developments of this project, the further enhancements to improve the system would be illustrated in this chapter.

1. Mechanical structure between the SG90 servo motor gear and joysticks

As section 6.2 mentioned, the iron wire is used to connect the motor gear and joysticks. This method could achieve the project aim, however, the structure of this method is simple and crude, sometimes the shape of iron wire is changed by accident, then the iron wire cannot pull the joystick as usual. This method also is not conducive to make subsequent upgrades.
By contrast, method 3 (figure 6.23) has a stable performance whenever and wherever, the 3D printed mechanical structure is fixed tightly with the joystick and motor gear. The mechanical structure is also flexible, it can be pulled to any valid position by the servo motor gear. This method is conducive to
achieve the direction control of the quadcopter.
Therefore, method 3 mentioned in section 6.2 will be further developed to improve the system.

2. Using Leap Motion to control the flight direction of the Quadcopter

Figure 7.1: left/right direction control example

The direction control system is almost the same with the throttle control system. To implement the direction control system, just copy the throttle control system on another stripboard, and the gear of SG90 servo motor should be connected to the direction joystick on the right of the handheld controller.

Two SG90 servo motors are required, one is to push/pull the joystick up and down, another is to push/pull the joystick left and right, in this way, the Quadcopter can be controller to fly

forward/backwards and left/right. The left/right direction control example can be seen in Figure 7.1.
After the future work is done, there would be 2 stripboards and 2 Leap motion for the 2 stripboards. one leap motion is connected to the throttle control system, users could use their left hand to control the throttle of the Quadcopter, another leap motion is connected to the direction control system, users could use their right hand to control the flight direction of the Quadcopter. The iron wire used to connect the gear and the joystick would be replaced by mechanical structure mentioned in Section 6.2.

3. Software optimization and upgrade

As section 6.4 mentioned, it is recommended to use 1 or 2 fingers to control the throttle of the quadcopter. Using more fingers would lower the signal transmission speed and make the servo motor rotates unstably.
Thus, the code in Processing 3 (mentioned in section 5.4) should be optimized to identify the gestures and transmitted the signal quickly.
Once the optimization is done, the code could be upgraded to identify more complex gestures and make the quadcopter do more movements. Currently, the movements are controlled by simple gestures, for example, move the finger up and down to control the throttle. In further work, a hand movement may be able to control a complex movement, for example, turn the hand around, the quadcopter will also turn around during the flight.

Chapter 8: Conclusion

A time plan reflection would be discussed and a conclusion for the whole project would be included in this chapter.

Section 8.1: Time plan reflection

Milestones	Task name	SEP	OCT	NOV	Dec	Jan	Feb	Mar	Apr	May	Month
	Project first meeting										09--10
	Background research										09--11
	Microcontroller selection										10--11
	Gesture sensor selection										10
	Gesture sensor learn and test										10--11
	Proposal										09--10
	Necessary components selection										10--12
	Drive integrated circuit test										11--01
	Motors test										12--01
	Moderator interview										10--11
	Handheld controller modified and test										12--01
	Sensor-PC-microcontroller connection										11--01
	Christmas Break										12--02
	Hardware construction										12--02
	Communication codes development										11--03
	Hardware and software combination										02--03
	System test without Quadcopter										03--04
	System test with Quadcopter										04--05
	Throttle control successfully										5
	Further work consider										5
	Final thesis										03--05
	Easter break										03--05
	Milestones check										5
	Final thesis deadline										5
	Presentation										5

Figure 8.1: time plan

As stated by the time plan (Figure 8.1), the project progress could be summarized by the following points:

1. Important milestones planned for components selection, sensor-PC-microcontroller connection, Hardware construction, communication codes development and system test with Quadcopter are all achieved on time satisfyingly, the throttle of the Quadcopter can be controller by Leap motion properly, which is very helpful for further development.

2. All tasks are finished and several methods of performance improvement are listed beyond expectation in chapter 7. For example, all movements of Quadcopter controlled by leap motion, one complex gesture could make the Quadcopter do a complex movement.

3. Due to the COVID-19, extra time and materials are required for investigating and simulating the idea of using the mechanical structure (mentioned in section 6.2) to connect the motor gear with the throttle joystick. To ensure the following planned schedule can be finished on time, the time for developing communication codes of Leap Motion and Arduino use of 20 days during the Christmas holiday.

4. The time used for necessary components selection was more than expected, the 3 project concepts mentioned in chapter 3 took a long time. Therefore less time can be used for hardware construction compared with the time plan. The work solved by making use of the whole week in the laboratory before COVID-19 situation start.

Section 8.2: Conclusion

Overall, all the objectives and tasks are achieved successfully. The throttle of the Quadcopter can be controlled by the Leap Motion gesture sensor properly. Based on the analysis in Chapter 6 and 7, with the aim of enhancing the system performance and upgrade the control system,

it is necessary to 3D print the mechanical structure between the SG90 motor gear and the joystick of the handheld controller, the mechanical structure is convenient for subsequent upgrades.

The other Leap motion and control system board is required to achieve the direction control of the Quadcopter.

Communication codes between the Leap motion and Arduino also could be optimised and upgraded to increase the response speed of the gesture recognition.

Due to the COVID-19 situation, lots of components cannot be delivered and the work which required laboratory cannot be finished, however, all important parts of the project have been completed successfully on time.
With sufficient conditions, the further work mentioned in Chapter 7 could be achieved easily.

Appendix: Supervisor Project Description

“Hands-free" control of a Quadcopter

A quadcopter is a multi-rotor vehicle that has, as its name suggests, four rotors. The popularity of the vehicle has increased in recent years and is used extensively by hobbyists and research institutions interested in unmanned aerial vehicles. For this project, the vehicle is to be supplied as a kit of parts, which is to be built and its operation demonstrated by initially flying the vehicle in the traditional manner via the use of a handheld radio controller. The vehicle supplied uses an Arduino based flight controller whose features and capabilities extend beyond those of traditional hand-held radio and vehicles using this controller are generically known as Arducopters [1]. The focus of this project is to take advantage of the enhanced capabilities of the flight controller and develop systems to communicate with the flight controller to control all or part of the vehicle's flight. These systems are to be PC based, avoiding the requirement for the traditional manual/hand-held controller. The ultimate aim is to design and develop a system that allows the user to control all or part of the vehicle's flight via an Xbox Kinect device[2]. The Kinect is a motion-sensing device intended to allow users to interact with games running on an Xbox 360. Microsoft has released a software development kit (SDK) that allows the functionality of the Kinect device to be accessed when connected to a PC and allows developers to build PC based applications using common e.g. C++ or Visual Basic. Using the Kinect the user’s gestures (or speech) can be translated into commands that are in turn transmitted to the vehicle, via a PC, without the user having to physically touch a controller.

Tasks: 1. Assemble the quadcopter vehicle, detailing the steps necessary to provide a vehicle that is ready to fly. 2. Investigate flight capabilities of the Arduino autopilot and demonstrate via a simple figure of 8 flight paths at fixed altitude. 3. Investigate methods of transmitting control signals from a PC to the vehicle. 4. Design a system capable of controlling the flight of the vehicle; this may be as simple as controlling lift off, to set altitude, prior to flying a predetermined flight path. 5. Acquire an understanding of the MS Xbox Kinect device and its SDK and develop an application to track simple body/limb movements. 6. Design and develop an interface that allows control of the quadcopter via the Kinect device; this again may be as simple as controlling lift off, to set altitude, prior to flying a predetermined flight path.

Communication codes on Arduino

```
// Throttle control
#include <Servo.h>
Servo myServo1;
/*int handPos;
int angle;*/
void setup() {

  myServo1.attach(6);
  myServo1.write(0);
  Serial.begin(9600);
//this can be changed

}
void loop() {
  byte angle;

  if (Serial.available()) {
    // Read the angle data from Processing
    angle = Serial.read();
    Serial.println(angle);
    // If fingers displayed on window, excute the data
    myServo1.write(angle);
```

```
// may use in direction control
#include <Servo.h>

Servo myServo;
Servo myServo2;
void setup() {

  myServo.attach(6);
  myServo2.attach(5);
  myServo.write(0);
  myServo2.write(0);
  Serial.begin(9600);
}

void loop() {
  byte angle;
  byte angle2;

  if (Serial.available()) {
    // Read angle from Processing
    angle = Serial.read();
    Serial.println(angle);
    // If fingers in window, read servo angle
    myServo.write(angle);
  }
    if (Serial.available()) {
```

```
    // Read angle from Processing
    angle2 = Serial.read();
    Serial.println(angle2);
    // If fingers in window, read servo angle
    myServo2.write(angle2);
  }
}
```

Communication codes on Processing 3 IDE (Leap motion)

```
// Throttle control
   int angle;
   import processing.serial.*;
   Serial port;
    public void setup() {
  // set window size
      size(720, 720, P3D); noFill();
      stroke(255);

  // set the leapmotion object
  leap = new LeapMotionP5(this); // a library for the project

  // set port and the baud rate.
  println("Available serial ports:");
  println(Serial.list());
  port = new Serial(this,Serial.list()[0], 9600);
 }
public void draw() {
  background(0);
      fill(255);
  for (Finger f : leap.getFingerList()) {
    PVector position = leap.getTip(f);
    //PVector velocity = leap.getVelocity(f);
       ellipse(position.x, position.y, 10, 10);
//for the finger recognition in window
    if (position.y > 720) {
      angle = 180;
    } else if (position.y < 0) {
      angle = 0;
    } else {
      angle = int((position.y) / 4);
}
//port communication
    port.write(angle);
    println(angle);
  }
}

public void stop() {
  leap.stop();
 }
```

```
//may use in direction control
import com.onformative.leap.LeapMotionP5;
import com.leapmotion.leap.Finger;
LeapMotionP5 leap;
int angle;
int angle2;
   import processing.serial.*;
   Serial port;
    public void setup() {
  // set window, P3D = 3D rendering
     size(720, 720, P3D);
    noFill();
     stroke(255);

  // set LEAP object
  leap = new LeapMotionP5(this);

  // set com port. Currently: "/dev/tty.usbmodemfd121"
  println("Available serial ports:");
  println(Serial.list());
  port = new Serial(this,Serial.list()[0], 9600);

 }
public void draw() {
  background(0);
     fill(255);
  for (Finger f : leap.getFingerList()) {

    PVector position = leap.getTip(f);

       ellipse(position.x, position.y, 10, 10);

    if (position.y > 720)
    {
      angle = 180;
    }
    else if (position.y < 0)
    {
      angle = 0;
    }
    else
    {
      angle = int((position.y) / 4);
    }
    port.write(angle);
    println(angle);

     if (position.x > 720)
    {
```

```
      angle2 = 180;
    }
    else if (position.x < 0)
    {
      angle2 = 0;
    }
    else
    {
      angle2 = int((position.x) / 4);
    }
    port.write(angle2);
    println(angle2);
  }


}

public void stop() {
  leap.stop();
 }
```

References

[1] ScienceDirect, Gesture Recognition [online] https://www.sciencedirect.com/topics/computer-science/gesture-recognition, access at (02/03/20)

[2] ENGINNERS GARAGE, Gesture Recognition Technology [online] https://www.engineersgarage.com/article_page/gesture-recognition-technology/, access at (04/03/20)

[3] What is Gesture Recognition Technology and How Does It Work? [online] http://www.nelson-miller.com/gesture-recognition-technology-work/ , access at (08/03/20)

[4] Lhongzhong, Leap Motion used experience [online], (01/25/2013), https://www.ifanr.com/240002, access at (15/10/2019)

[5] Vladimir I. Pavlovic , Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. [online] https://www.cs.rutgers.edu/~vladimir/pub/pavlovic97pami.pdf, access at (08/03/20)

[6] Official website, Azure Kinect DK, [online] https://azure.microsoft.com/zh-cn/services/kinect-dk/ , access at (20/10/2019)

[7] Official website, Intel's 3D camera, [online] https://www.intel.co.uk/content/www/uk/en/architecture-and-technology/realsense-overview.html , access at (20/10/2019).

[8] Official website, Leap Motion, [online] https://www.leapmotion.com/zh-hans/, access at (20/10/2019).

[9] Divya Joshi, Drone technology uses and applications for commercial, industrial and military drones in 2020 and the future, [online] https://www.businessinsider.com/drone-technology-uses-applications?r=US&IR=T access at (22/10/2019)

[10] PX4 Dev Team, Flying 101, [online] https://docs.px4.io/v1.8.2/en/flying/basic_flying.htmlV , access at (20/10/2019).

[11] Which one is better Kinect, SoftKinectic or Leap Motion? [online] https://www.quora.com/Which-one-is-better-Kinect-SoftKinectic-or-Leap-Motion, access at (04/04/2020).

[12] Official website, Leap Motion, [online] https://www.leapmotion.com/zh-hans/, access at (20/10/2019).

[13] Leap Motion Overview [online] https://blog.csdn.net/weixin_33853794/article/details/86047996, (05/04/2020).

[14] Blade Inductrix FPV Ultra Micro Quadcopter Drone with Monitor and SAFE Technology - BLH8500G [online] https://www.wirelessmadness.com/blade-inductrix-fpv-ultra-micro-quadcopter-drone-with-monitor-and-safe-technology-blh8500?language=en&currency=GBP&gclid=CjwKCAjw7-P1BRA2EiwAXoPWAyQkp_6y1Q02exYQcFR9_mvclmR0EfIctF_0eJmyGWvodTF_5HI4_xoCzksQAvD_BwE, access at (05/02/2020)

[15] SPMR8000EU ： DX8 TRANSMITTER ONLY MODE 2 EU [ONLINE] http://vi.raptor.ebaydesc.com/ws/eBayISAPI.dll?ViewItemDescV4&item=153888366763&category=40703&pm=1&ds=0&t=1589044968126 , access at (05/02/2020)

[16] SparkFun I2C DAC Breakout - MCP4725 [online] https://www.sparkfun.com/products/12918, access at (05/02/2020)

[17] Advantages of Digital Potentiometer | disadvantages of Digital Potentiometer [online] https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Digital-Potentiometer-over-Analog-Potentiometer.html , access at (05/02/2020)

[18] Micro Metal Gearmotor (Extended back shaft) – 50:1 [online] https://shop.pimoroni.com/products/micro-metal-gearmotor-extended-back-shaft?variant=3073681025 , access at (16/02/2020)

[19] SERVO MOTOR SG90 DATA SHEET [online] http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf , access at (16/02/2020)

[20] Dejan, Arduino DC Motor Control Tutorial – L298N | PWM | H-Bridge [online] https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/ , access at (16/02/2020)

[21] 2478 KEYSTONE [online] https://www.tme.eu/gb/details/keys2478/batteries-holders/keystone/2478/?brutto=1&gclid=CjwKCAjw7-P1BRA2EiwAXoPWA5_LaYf5nkCuzmBK9a2_mDOILdR_mt1x3omYvNkOP2HaTDAG8DulnxoCEhEQAvD_BwE , access at (16/02/2020)

[22] the_ice_man, The LeapCopter! (Leap Motion to Control Quad Copter), (2014), [online] https://www.instructables.com/id/The-LeapCopter-Leap-motion-to-control-Quad-copter-/, access at (16/01/2020)

[23] Nick Koumaris, Using the SG90 Servo Motor With an Arduino [online] https://www.electronics-lab.com/project/using-sg90-servo-motor-arduino/, access at (14/04/2020)

[24] Interface L298N DC Motor Driver Module with Arduino [online] https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/, access at (22/02/2020)

[25] Nick Koumaris, Using the SG90 Servo Motor With an Arduino [online] https://www.electronics-lab.com/project/using-sg90-servo-motor-arduino/, access at (28/02/2020)

[26] Information Systems Services [online] https://www.dcu.ie/iss/software/Processing-P3.shtml, access at (25/04/2020)

[27] Real-Time Hand Gesture Recognition Using Finger Segmentation [online] https://www.hindawi.com/journals/tswj/2014/267872/ access at (02/05/2020)