**Final Project Logbook**

**“Hands-free" control of a Quadcopter**

**Student: NANDI GUO-14311495**

**Introduction:**

The popularity of quadcopters has been increasing in recent years, they have been used for business, society, first aid, etc. 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.

**Aims:**

a. This project aims to offer a human interface instead of the traditional radio controller.

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

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

d. Complete and try to upgrade the control system.

**Objectives:**

a. Choose a gesture/motion sensor and do a test through example programs of the sensor driver.

b. Interface the sensor system and to the PC and develop software to make their function combined.

c. Choose digital potentiometers/Special drive integrated circuit and design a method to drive the handheld remote-control unit.

d. Integrate gesture recognition and quad controller into a single system

e. Test gesture control working.

f. Write a thesis after the project is completed.

The basic equipment needs:

1. Quadcopter
2. Transmitter
3. microcontroller
4. gesture sensor
5. digital potentiometers or drive integrated circuit
6. PC, power supply

25/09/19

**Time plan check**

01/10/19

**Project design plan**

1. 
2. 

13/11/19

**Plan 1 advantages:**

* Don’t have to cut wires in the transmitter, this plan won’t damage the expensive transmitter. low risk.
* Easier to achieve, combined mechanical components and embedded programming.
* Easier to program.

**Plan 1 disadvantages:**

* More components needed than plan 2
* The control accuracy is lower than plan 2

**Plan 2 advantages:**

* Fewer components, simpler structure
* High accuracy, fast

**Plan 2 disadvantages:**

* Harder concept
* Easier to damage the expensive transmitter
* More complex programming and circuits construction

14/11/19

**Project Plan 1 concept and design**

* The quadcopter has confirmed

Inductrix blade small copter

Required: 4+ channel DSM2/DSMX transmitter



**Figure 1: Blade Inductrix FPV Ultra Micro Quadcopter Drone**

* The transmitter has confirmed

Dx8 transmitter

It is already able to combine with the copter

**Figure 2: DX8 8-Channel DSMX Transmitter**

Test:

Bind successfully

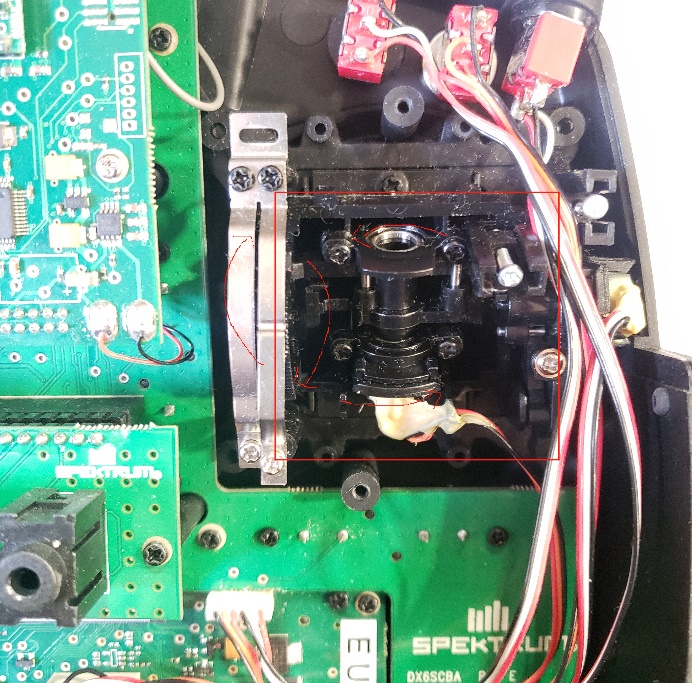
Fly control successfully 15/11/19

* Problem: the battery of the copter is almost broken, the battery can only last 2 minutes.

solution: New battery lipo 3.7v 200mh 0.74wh 01/12/19

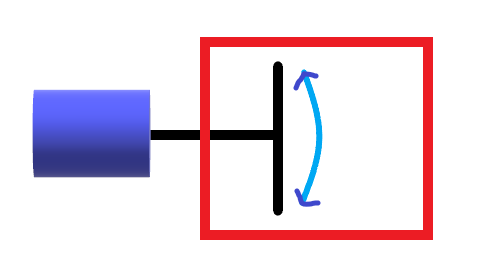
* Problem: the First transmitter can not compile with the copter (unknown reason)

Solution: new transmitter 10/11/19

**Transmitter Internal structure**

**Figure 3: throttle control part**

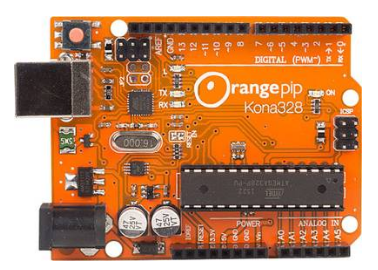
In plan 1, the red part should be added a motor to push the mechanical part to control the throttle, the motor should be controlled by a microcontroller which is connected to a gesture sensor. 15/12/19



motor

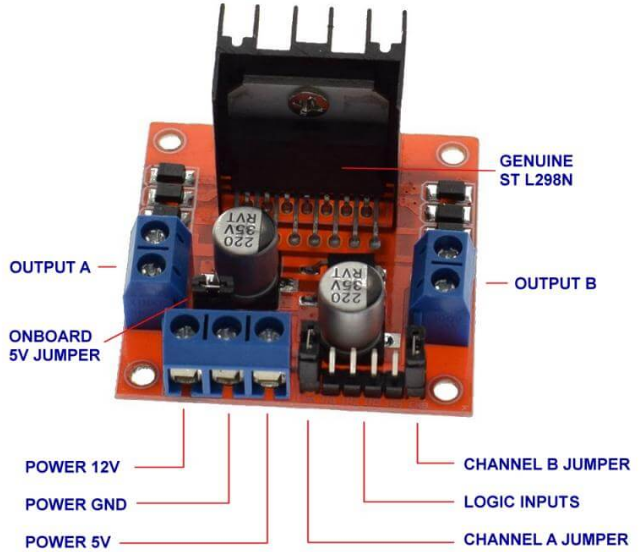
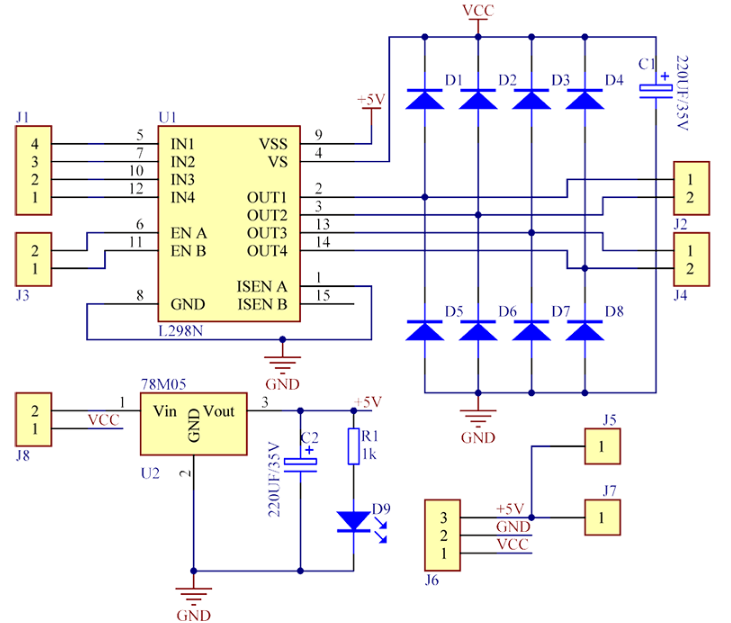
Red part

**Figure 4: Basic concept**

**Microcontroller choose:**

**Figure 5: orange pip kona328**

The Orange PIP Kona 328 development board centers around the ATmega328 microcontroller and comes with full Arduino ™ UNO compatibility. 20/12/19

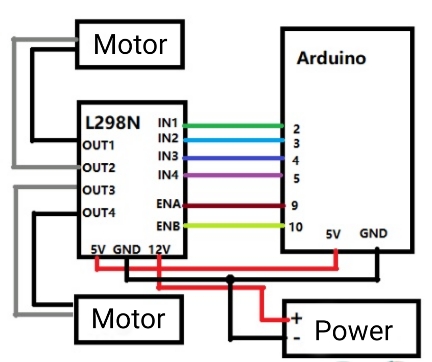
**Drive integrated circuit:**

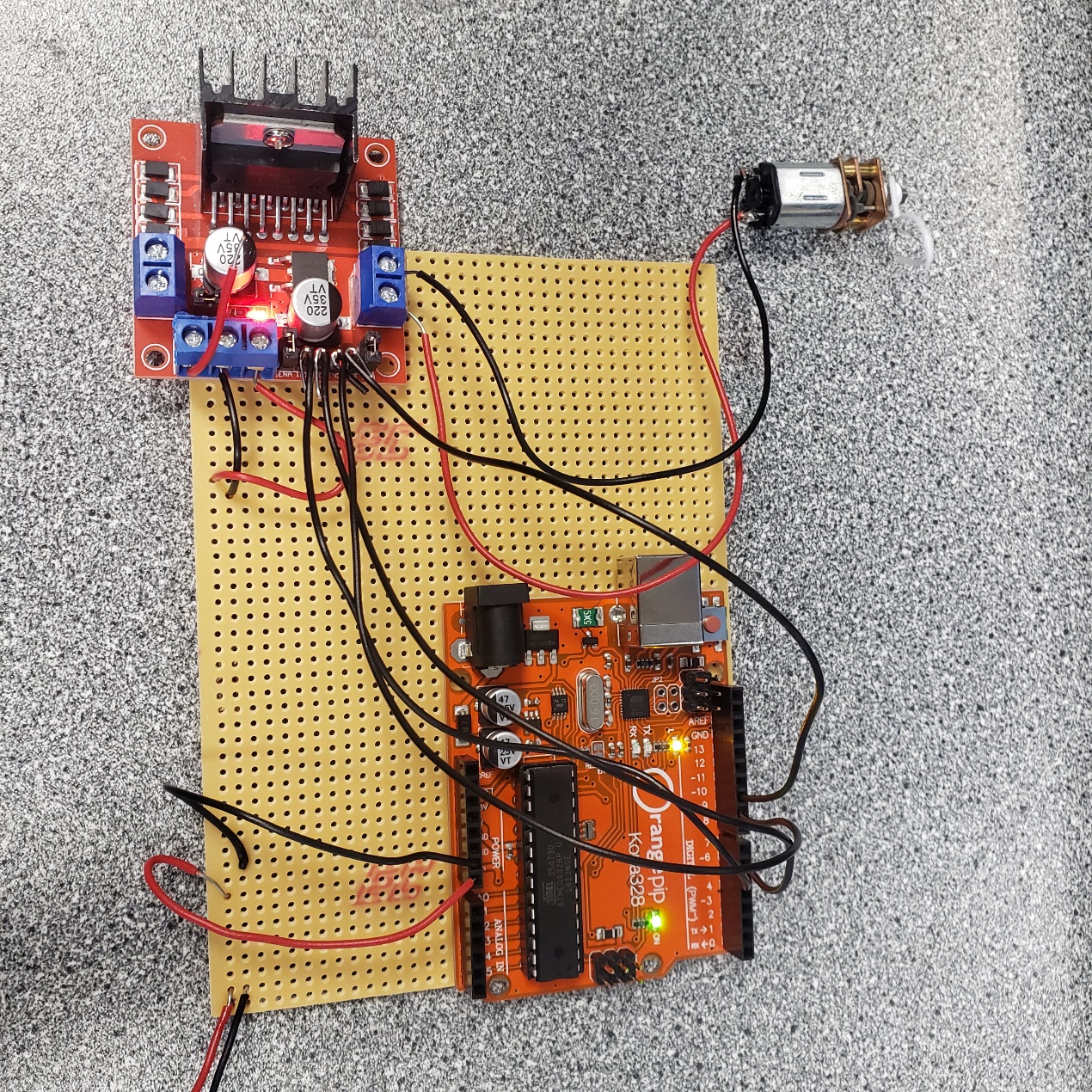
**Figure 6:** **L298n stepper motor drive board and circuit**

30/12/19

**Specifications of L298N Motor Driver**

1. The module will allow you to control the speed and direction of two DC motors.
2. It can control motors that operate between 5 to 35V and up to 2A.
3. The module has an onboard regulator which helps in giving the output of 5V.
4. 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.
5. It can also control a stepper motor.
6. 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.

**Figure 7:** **circuit of L298n with Arduino**

**Figure 8: logic of L298n**

**Figure 9: stripboard of L298n**

* Problem:

There are no I2c wires in the 8th floor, I can only use the normal wires, the I2c wires will arrive in several days. 17/01/20

Solution:

I got the wires from eBay  21/01/20

**Example codes in Arduino**

void loop() {

//forward

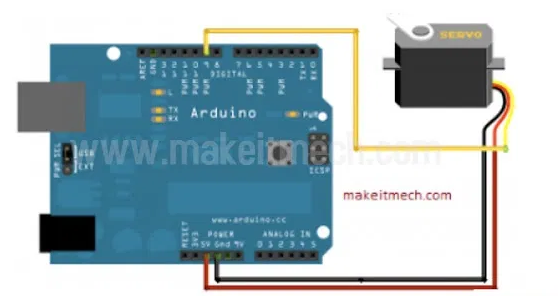
digitalWrite(input1,HIGH); //high

digitalWrite(input2,LOW); //low

delay(1000); //time

}

The motor should be fixed on the transmitter and a mechanical connecting component should be designed.

The motor can also be SG90, this type will not need L298n circuit, I haven’t tried it, maybe later.

**Figure 10: SG90 servo**

Servo is curtained as the main part of throttle control.

28/01/20

Within using a motor, the rotation angle of the motor can not be curtained thus the accurate throttle of the transmitter can not be curtained, the fly control would be unstable.

L298n is still used to provide a steady 5v power supply to microcontroller and servo because the battery pack is providing a 6v voltage, L298n has a 12v input pin and a 5v output pin.

An SG90 servo can be controlled accurately, the rotation angle could change slightly with the motion of fingers.

The logical communication

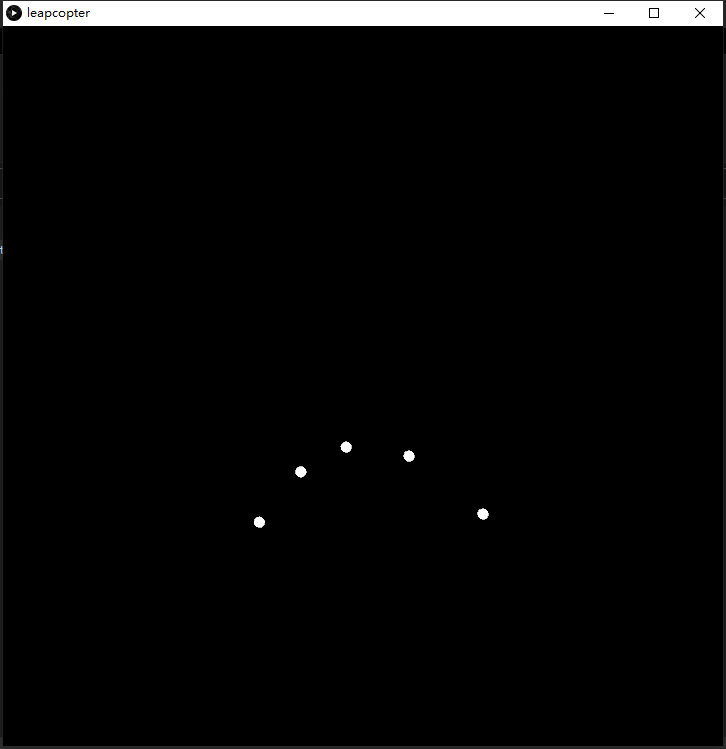
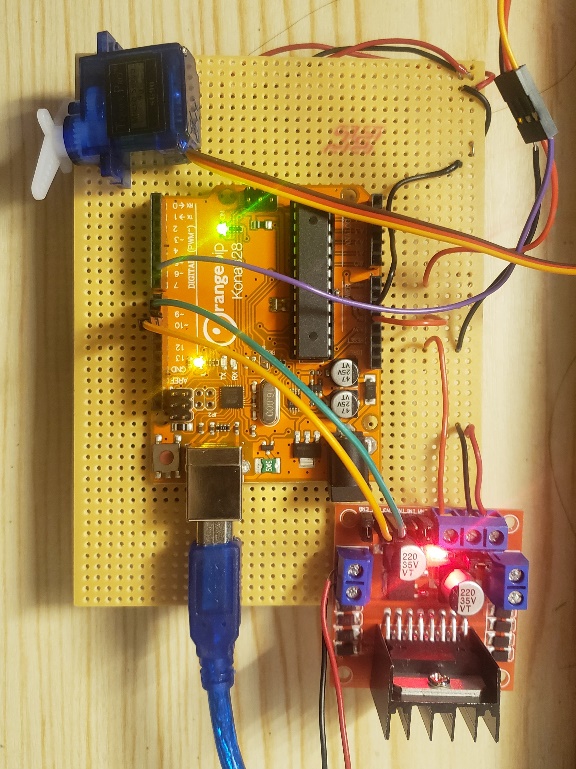


Attach servo with pin # 6 to control it with Hand gesture recognition technology. VCC is connected with 5v and black wire of motor connected with Ground. After uploading the codes it turns first 0 degrees (this could make sure the throttle won’t be high) then tracks gestures on the y-axis and changed its angles as per gestures changes.

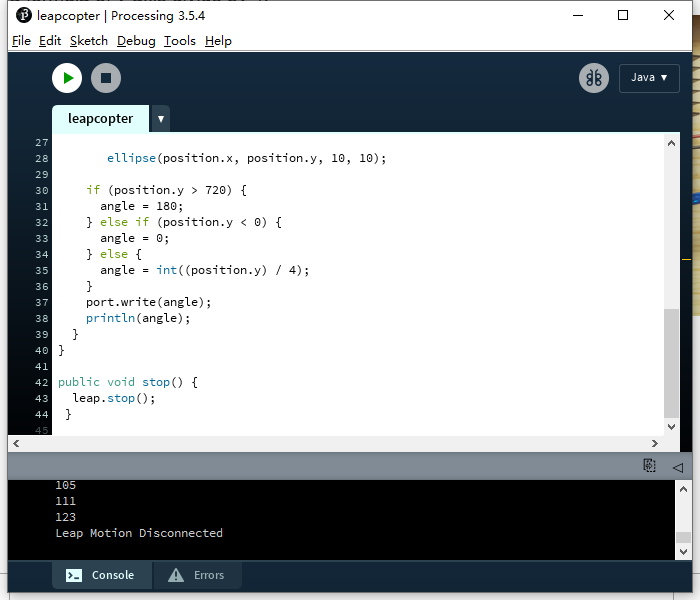
For coding part

Making a class of leap and making an integer of name angle which is used further for storing the angles of the servo motor and send it to Arduino. 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 the zero the angle turns to 0 degrees. In between the windows, the angle has a formula of the y-axis divide by 4.

**Figure 11: processing 3 window and servo circuit**  15/02/20

**Figure 12: example codes in processing 3**

In the if loop, y is the main role to identify the value of the height of the hand to make sure the angle of the servo.

18/02/20

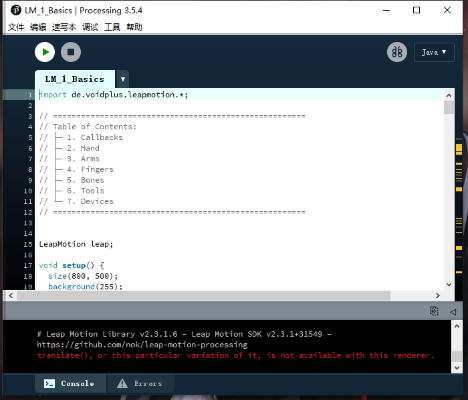
**Gesture Sensor**

Leap-motion

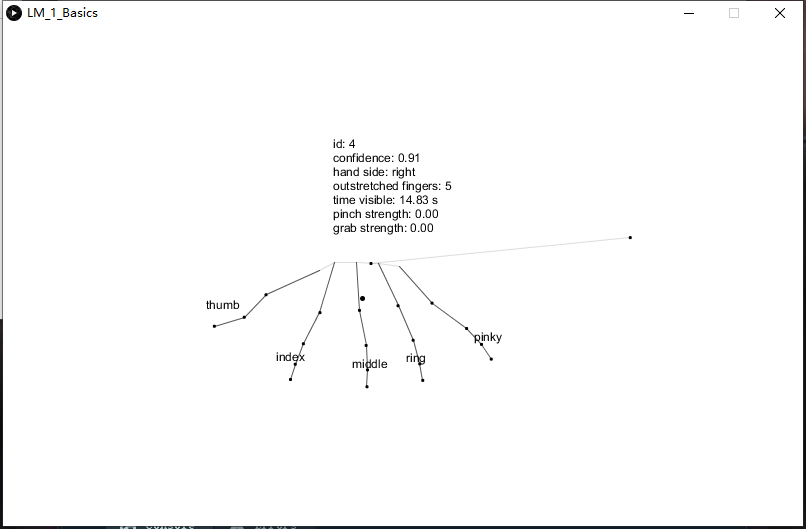
**Figure 12: leap-motion and leap-motion control panel**

The drive is downloaded, the leap-motion is identified.

Problem: it can’t run any example software on the drive

Solution: Go to the directory, Find a File "LeapSvc64", click to install it as a 64-bit version. 13/11/19

**Figure 13: “processing 3”, the IDE to compile codes for the leap-motion.**

  **Figure 14:** **The Gesture recognition program**

03/01/20

**Communications between Arduino and Processing 3**

* This can be regarded as the communication between leap-motion and Arduino board

1. already have the codes for processing3 and Arduino, I haven’t compiled yet.
2. Once the codes compile successful, I should be able to use the leap-motion to control the throttle of the copter.

10/02/20

#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);

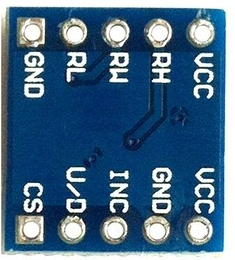
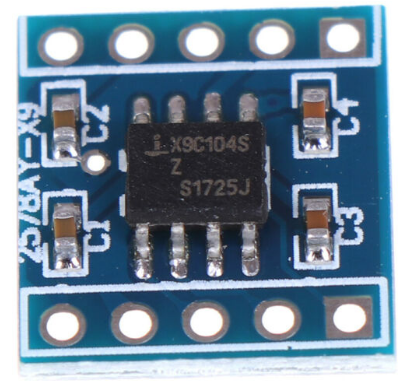
}

}

//codes for arduino

//SERVO

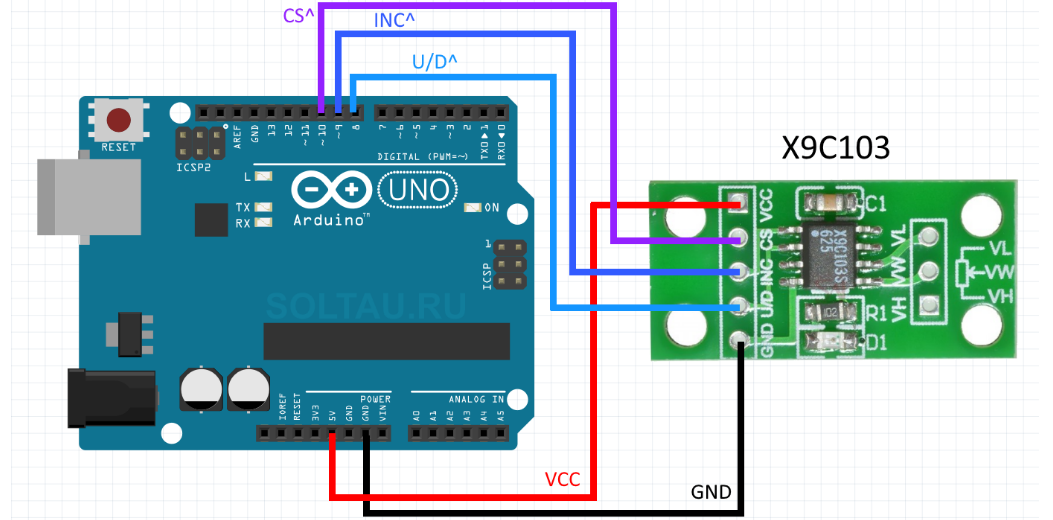
**Project Plan 2 concept and design**

****

**Figure 14:**

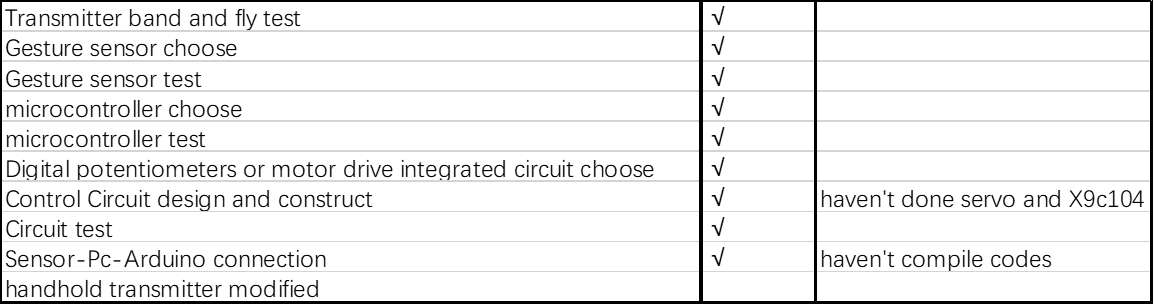
**Digital potentiometer**

**X9C104**



**Figure 15: example circuit of the digital potentiometer**

The communication concept is almost the same with Plan1, I haven't written the codes for this plan yet. 05/02/20

**Task progress check** 10/02/20