LYU1401 - AndroidCopter

Supervised by Prof. LYU Rung Tsong Michael

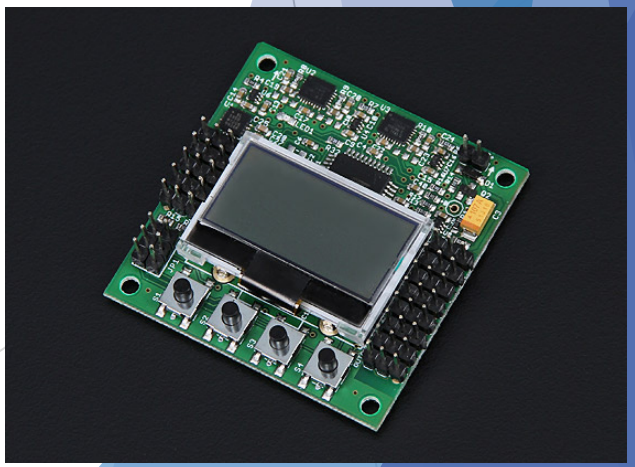
Lam Ka Ho 1155018465

Wong Chor Man 1155018466

Introduction

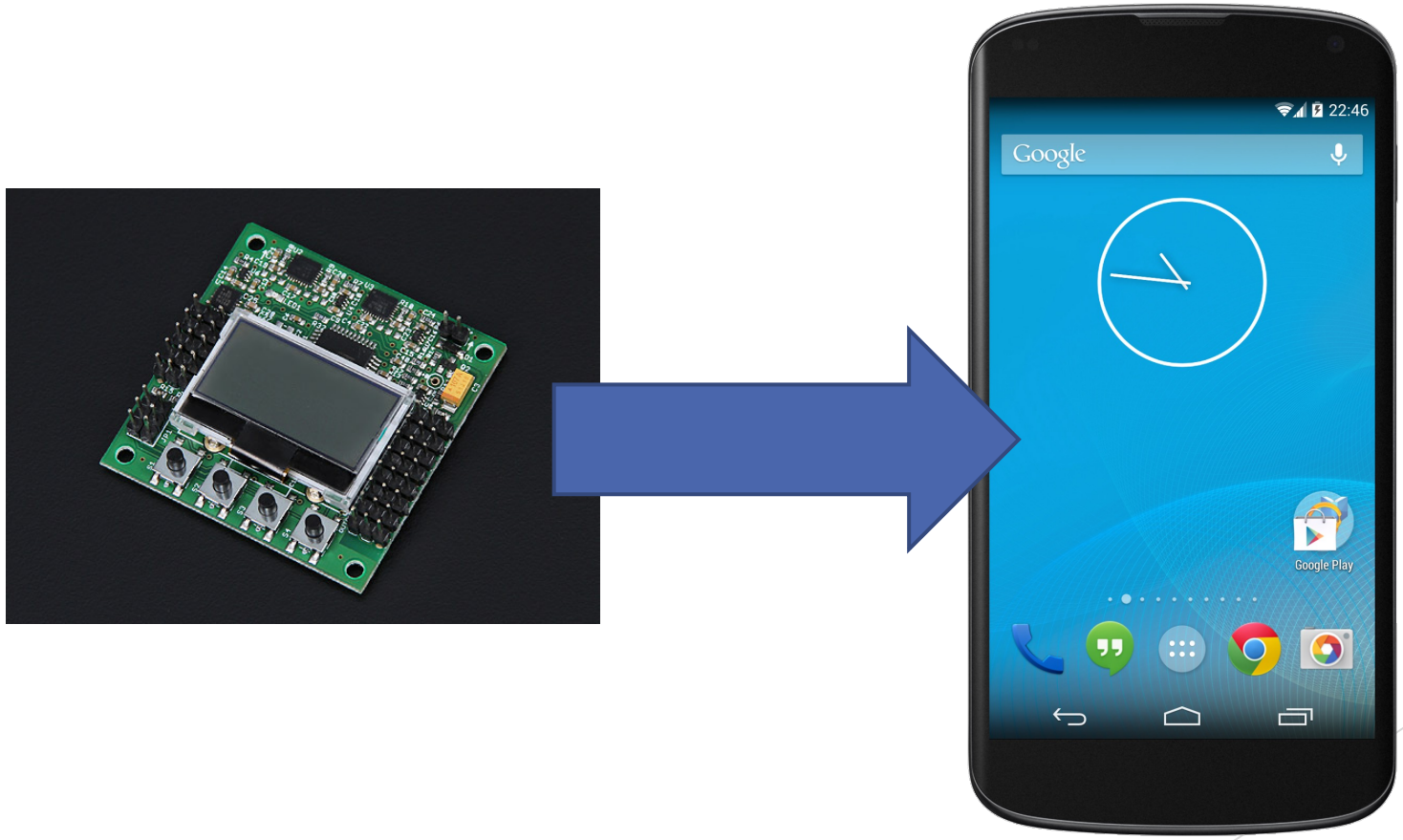
- Quadcopter is very popular:
  - ✓ Recording video,
  - ✓ Having Fun
  - ✓ Delivery Goods like Amazon Prime Air

- Most quadcopters are using flight control board
  - ✓ No extensions
  - ✓ Slow CPU Power



Objectives

- Replace a Flight Control Board with an Android Phone
- Build a Super Easy API for extension



Problems to solve

1. How Android controls the motors?
2. How a quadcopter move?
3. How an AndroidCopter balance in the air?
4. How to control the AndroidCopter?
5. How to keep it safe?
6. How is the API?

1. How Android controls the motors?

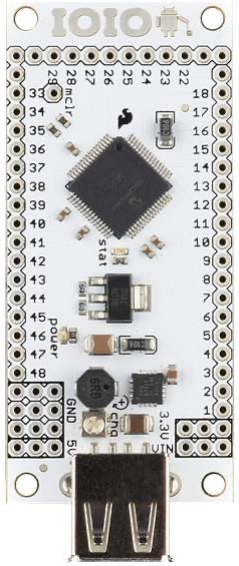
How Android controls the motors?

- The most important question, if it cannot, this project cannot be continued.

- It is possible, but need TWO things in between Android and the motors:
  - IOIO Board
  - Electronic Speed Controller (ESC)

What is IOIO Board?

- Make Android to interact with external electronic modules using signals
  - E.g. Digital IO, Pulse IO, Analog Input.
- Android connect the IOIO board with USB or Bluetooth
- Using Java code to process the I/O functions

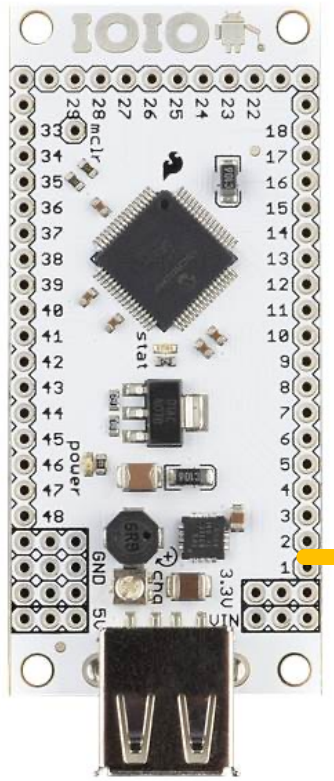


IOIO Board

IOIO Usage in the project

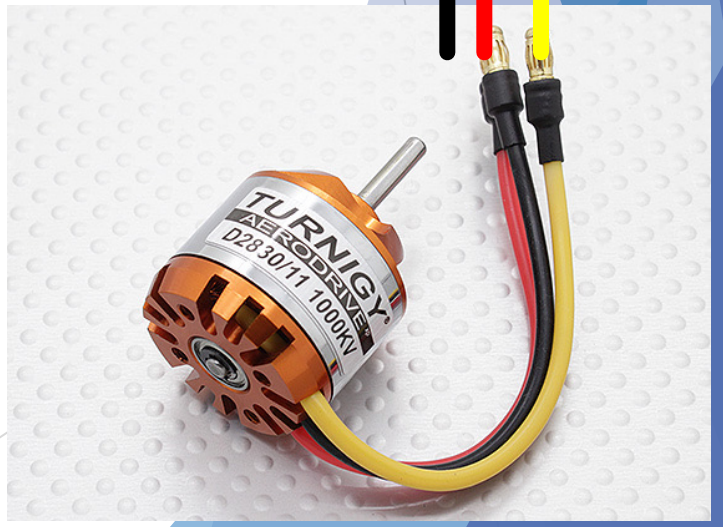
- Mainly used Pulse Output for outputting the motors signals to ESC

- pwm.setPulseWidth(1000); // 1000 – 2000 microseconds



ESC

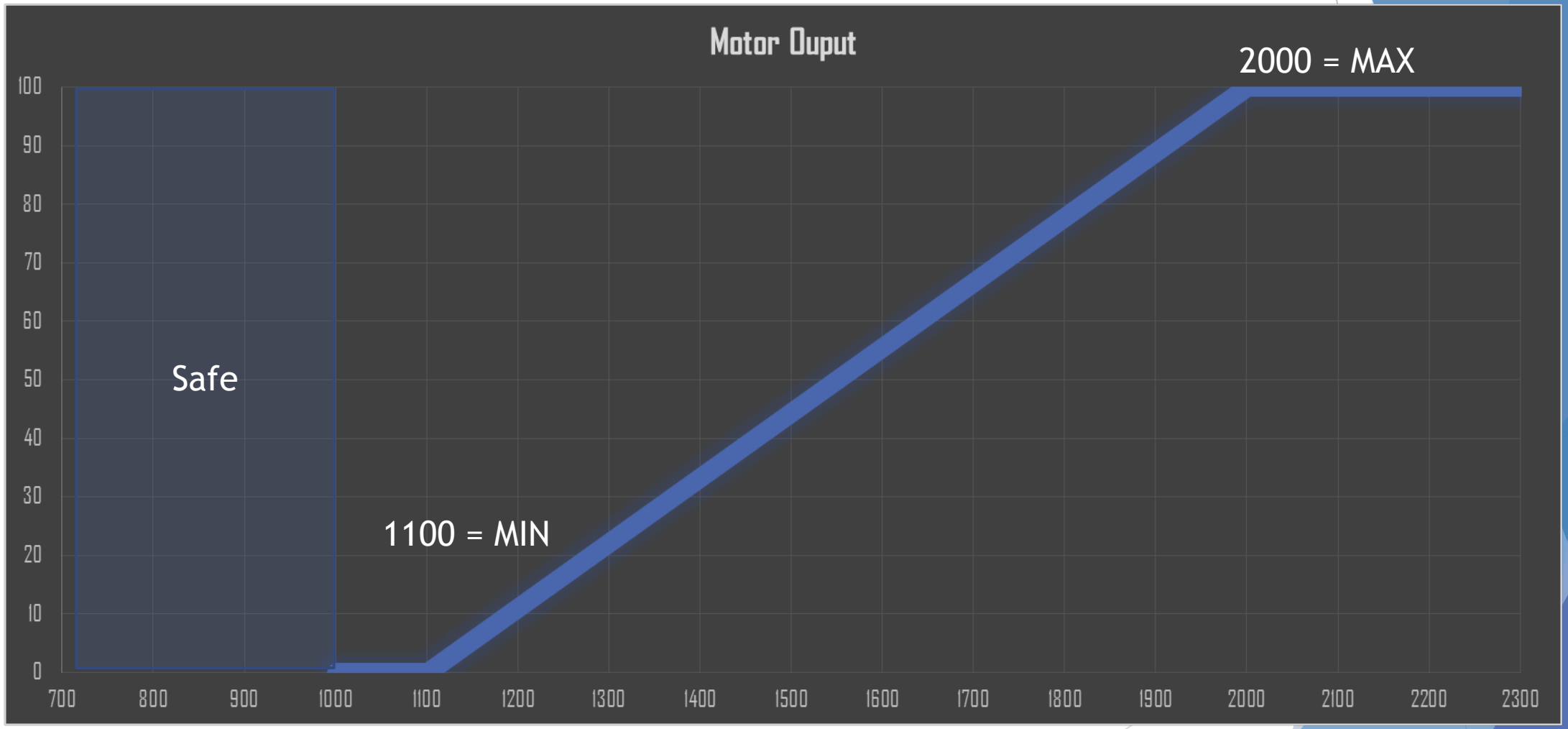
1500us
Pulse



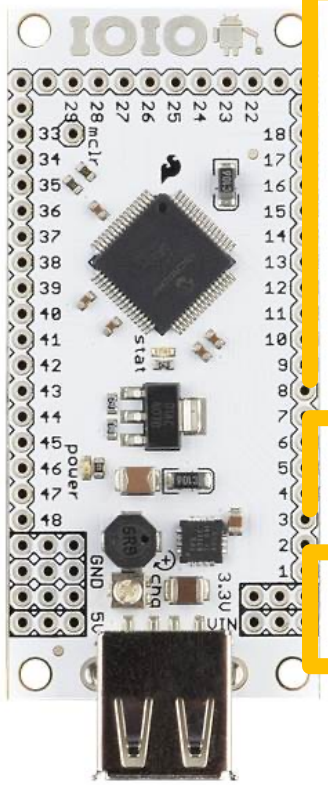
USB

```
pin1.setPulseWidth(1500);
```

Pulse Width / Motor Power



Quadcopter = 4 motors

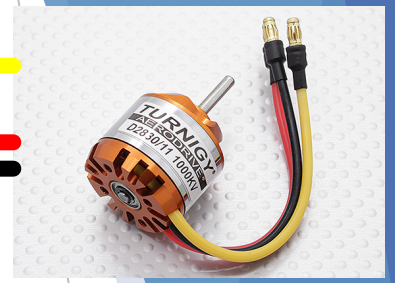


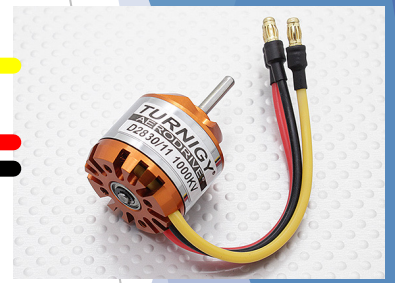
ESC

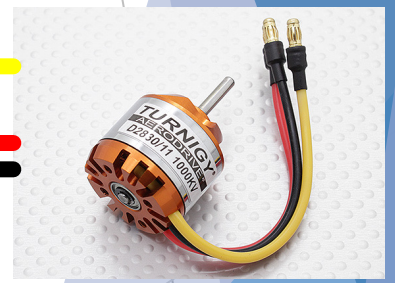
ESC

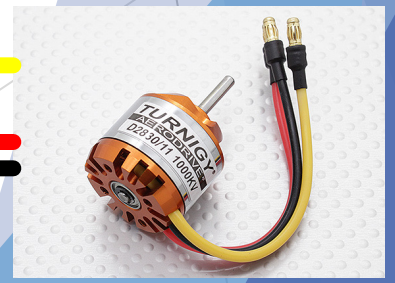
ESC

ESC



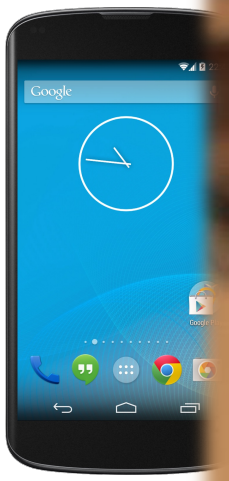






USB

Hardware



Android Ph

Electronic Sp

x4

x4

Frame

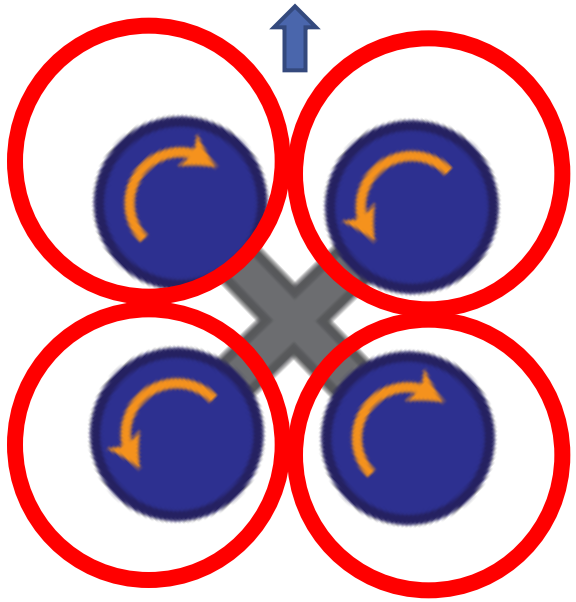
Battery

2. How a quadcopter move?

- Four types of movement:
  - **Altitude:** Up/Down
  - **Pitch:** Forward/Backward
  - **Roll:** Left/Right
  - **Yaw:** Rotate Left/Right

Adjust Altitude

- Copter go up or down

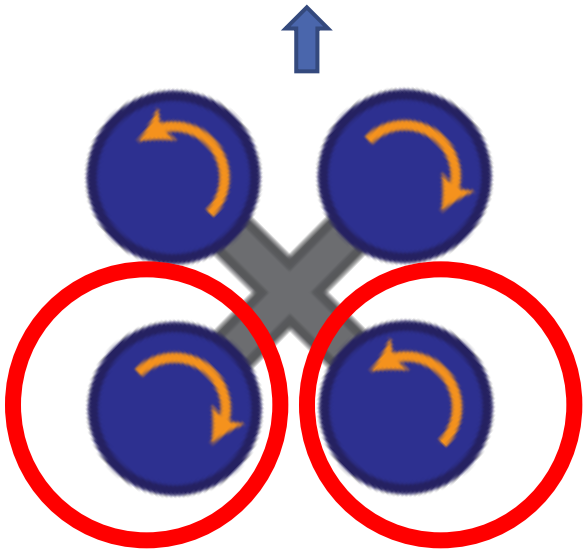


Speed up all motors

Go Up

Adjust Pitch

- Copter move forward and backward



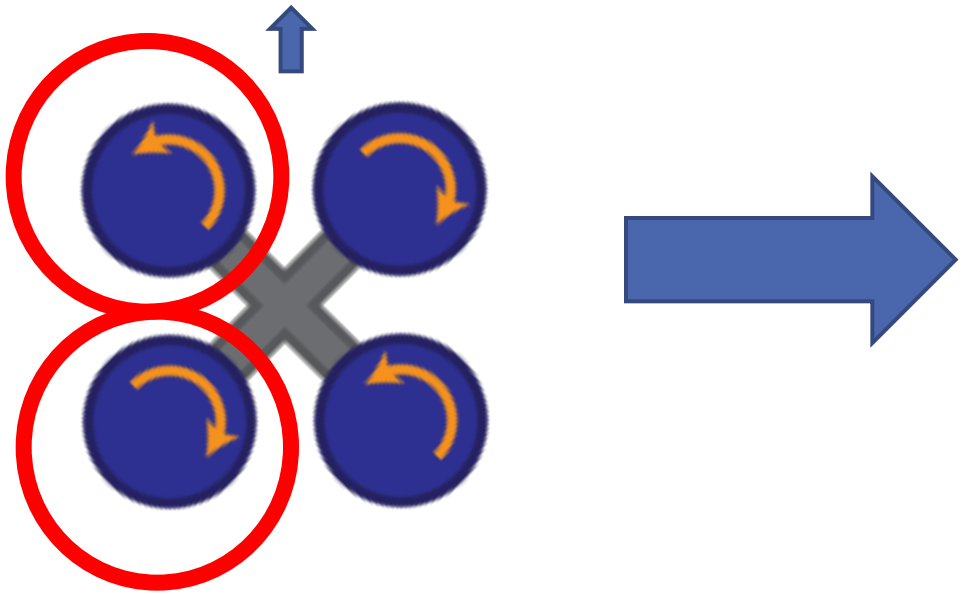
Speed up the rear set of motors
Slow down the front set of motors

Back Front

Move Forward

Adjust Roll

- Copter go to left or right



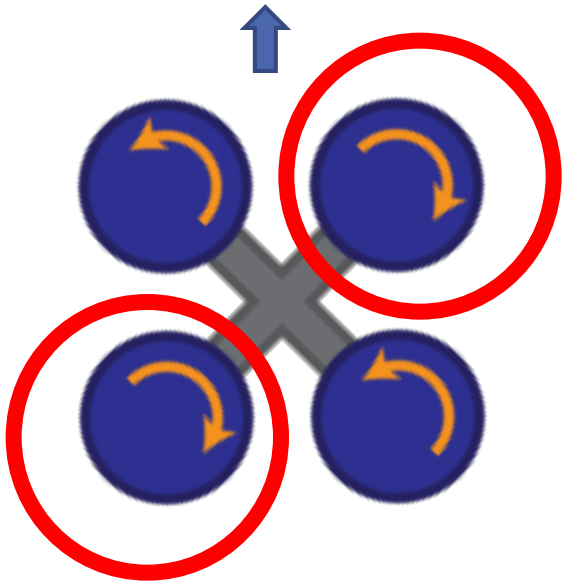
Speed up the left set of motors
Slow down the right set of motors

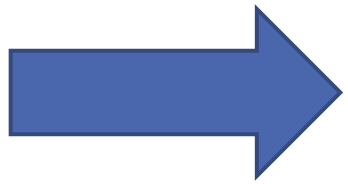
Left Right

Go Right

Adjust Yaw

- Copter turn left or right



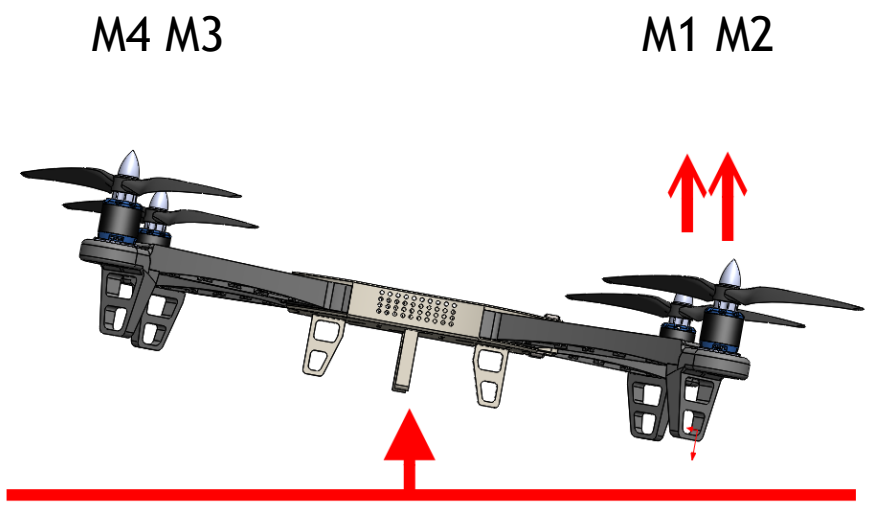


Speed up the clockwise motors
Slow down the anti-clockwise motors

Rotate Left

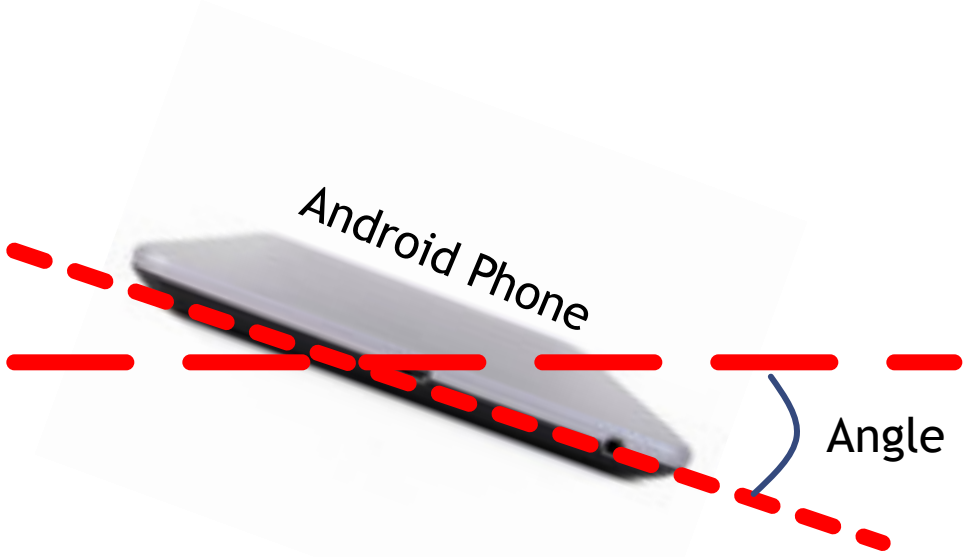
3. How an AndroidCopter balance in the air?

Stabilization Algorithm



- If we want to keep it horizontally…
- M1, M2 speed up
- How to know the current state?
- How much power should generate?

Accelerometer and Gyroscope



- For Long term, Accelerometer
  - 3-axis Orientation
  - Slow response in short term
- For Short term, Gyroscope
  - 3-axis angular speed
  - Drifting in long term
- We have to combine TWO sensors data

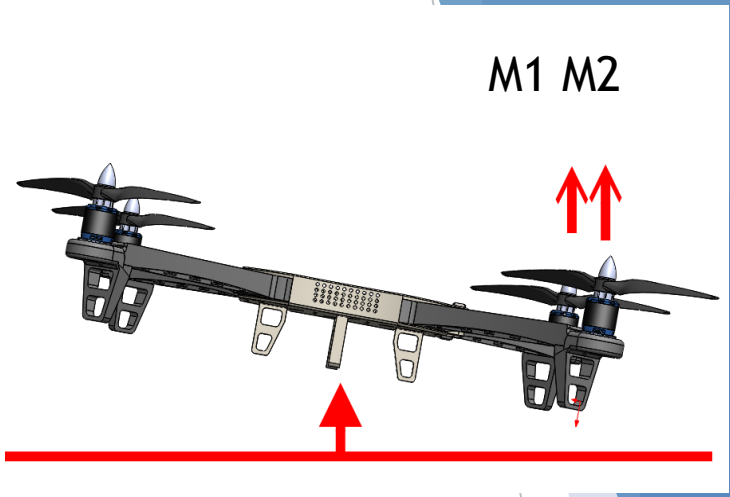
Motors Outputs

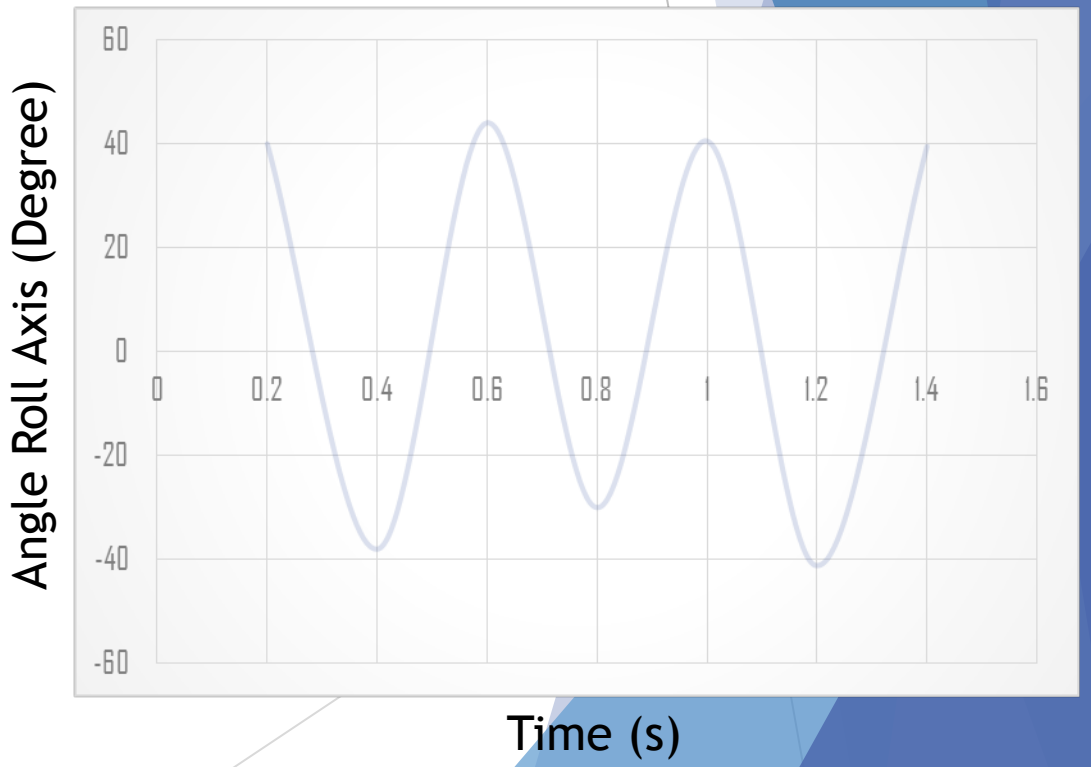
- Motors Outputs are related to the angle.

- Error = Actual Angle - Desired Angle
- Error Larger => Output Larger

- If linear equation:
- 1500 + Error * K

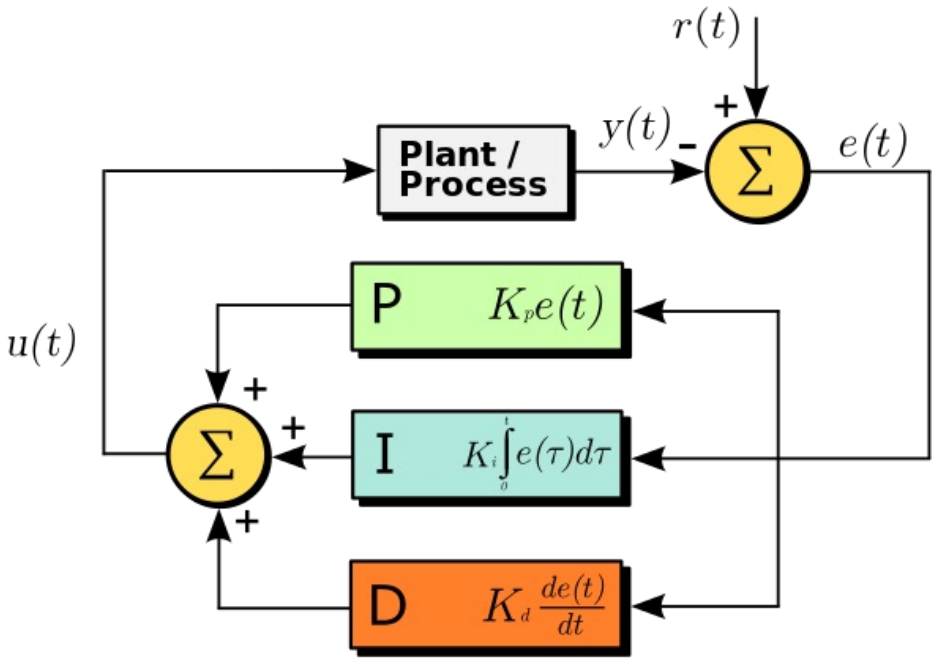
- But vibrating..





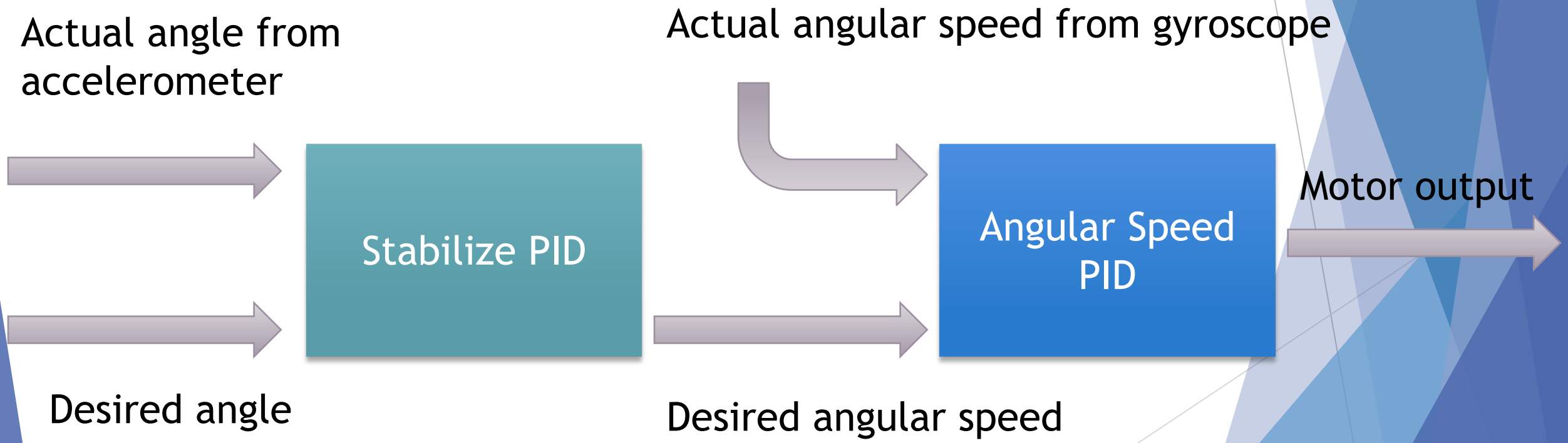
PID controller

- **Proportional-integral-derivative controller**
- Attempts to minimize the *error between a actual value and a desired value*

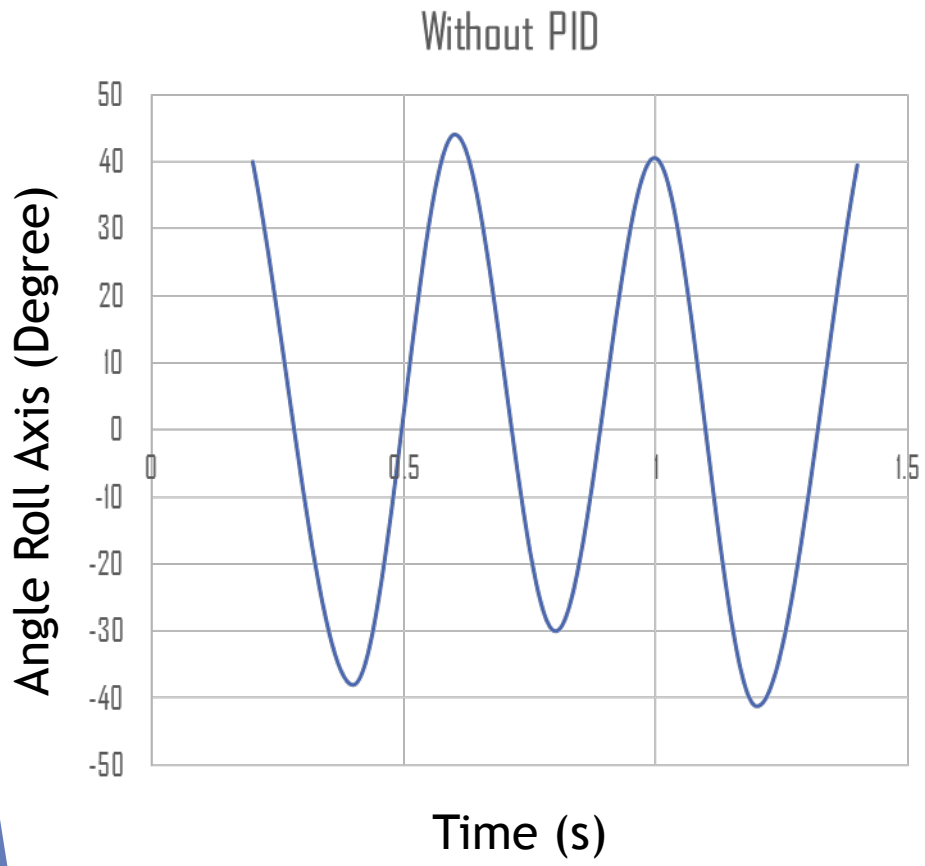


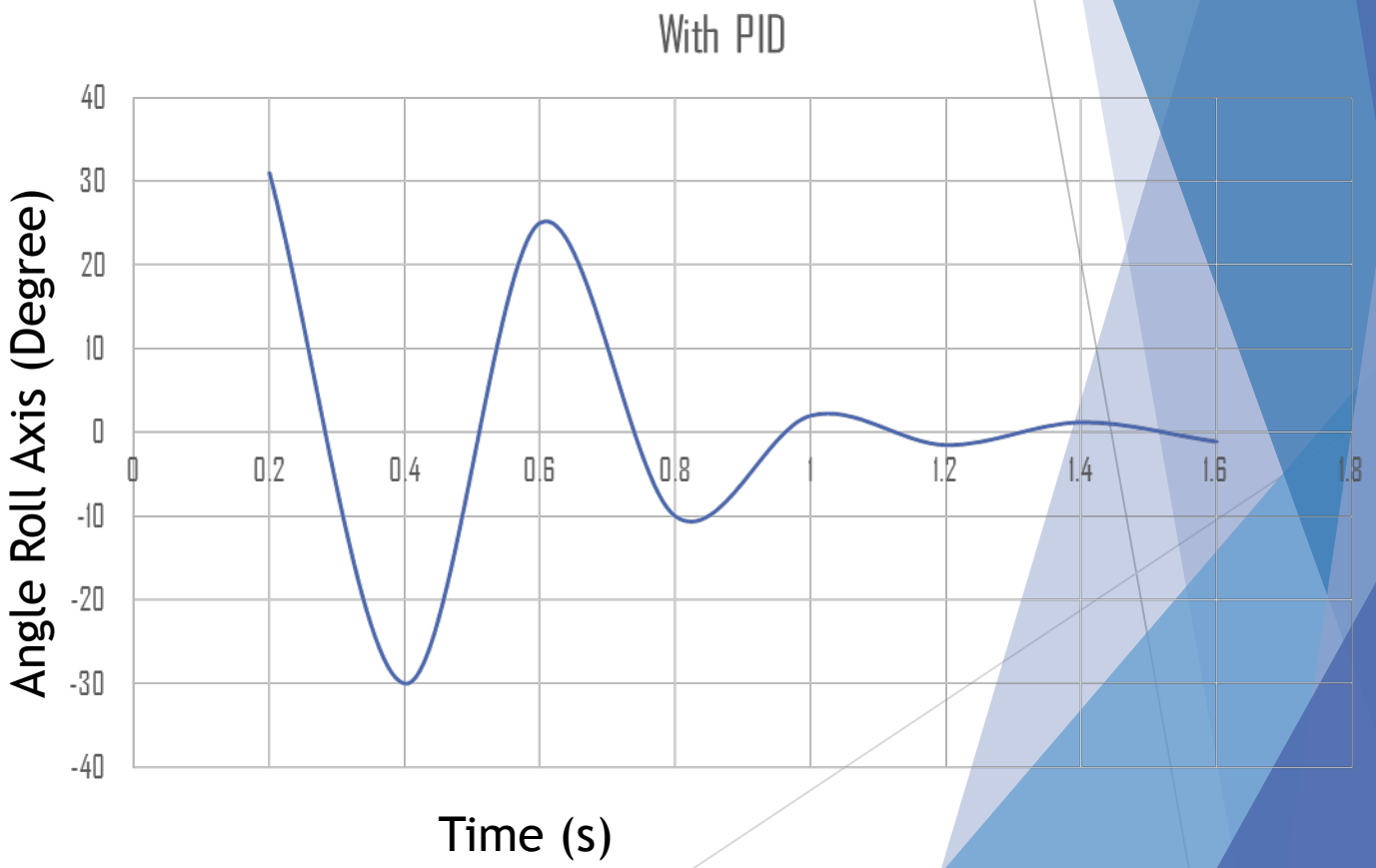
Final algorithm

PID for each Axis

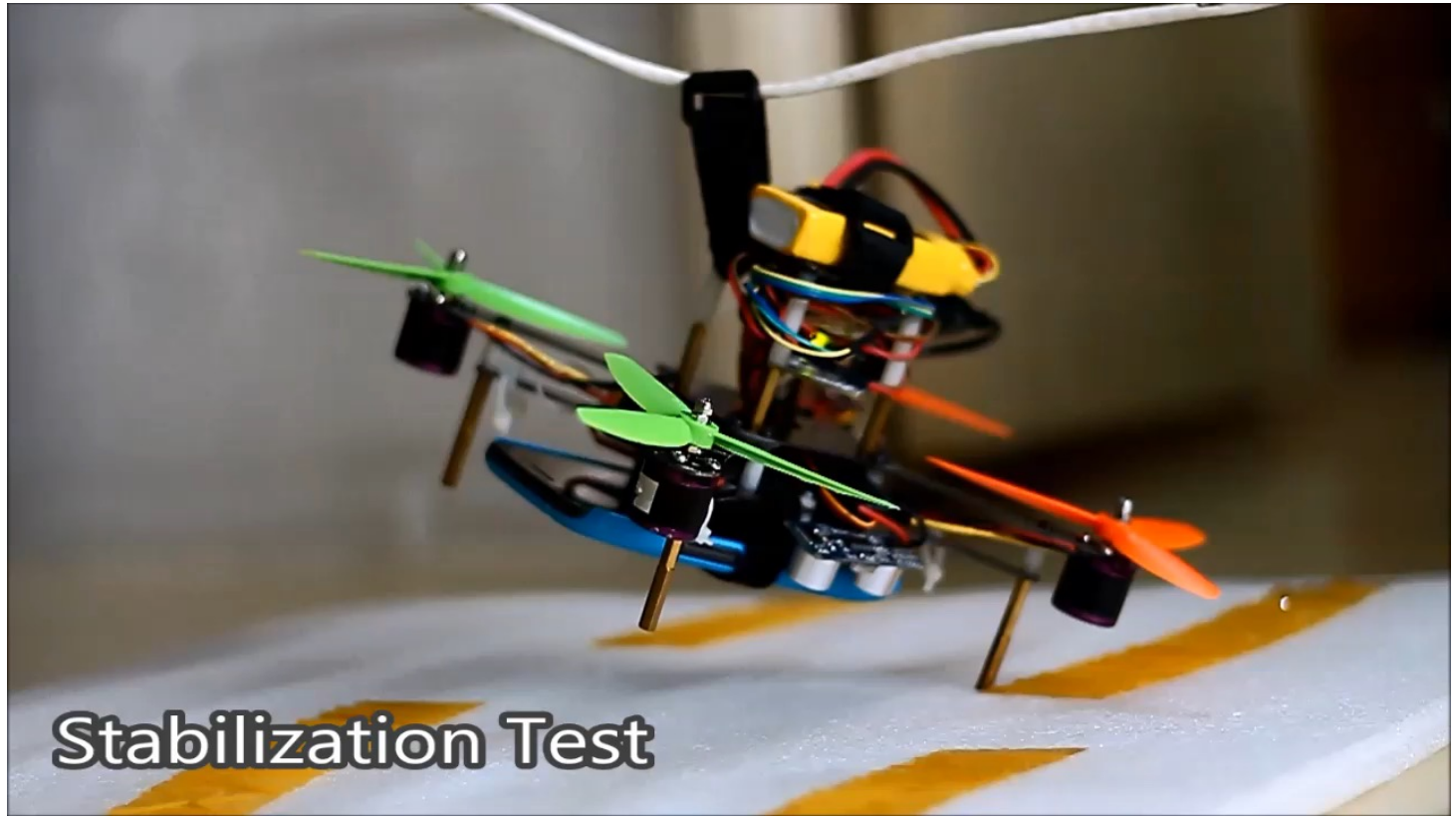


Experiments for Roll Axis



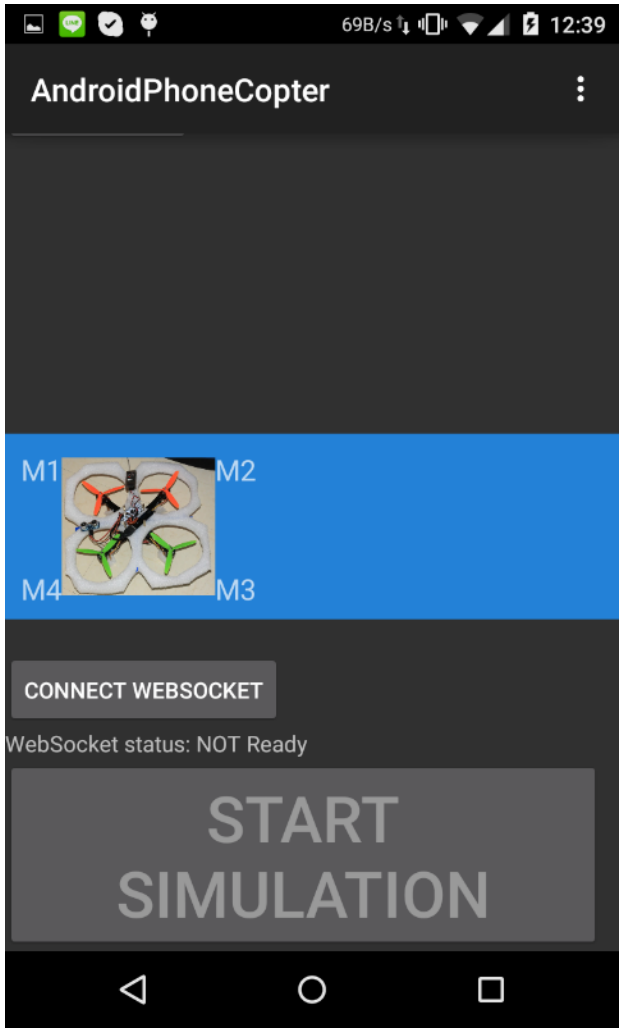


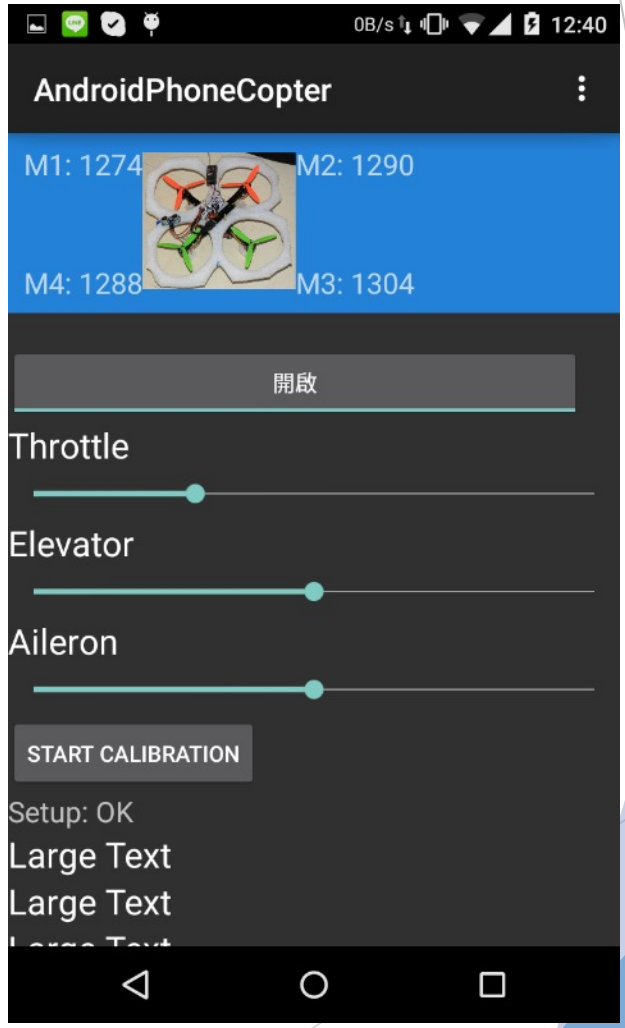
Testing and Demonstration



Android App


0B/s 12:38
AndroidPhoneCopter


69B/s 12:39
AndroidPhoneCopter
M1
M2
M4
M3
CONNECT WEBSOCKET
WebSocket status: NOT Ready
START SIMULATION


0B/s 12:40
AndroidPhoneCopter
M1: 1274
M2: 1290
M4: 1288
M3: 1304
開啟
Throttle
Elevator
Aileron
START CALIBRATION
Setup: OK
Large Text
Large Text

4. How to control the AndroidCopter?

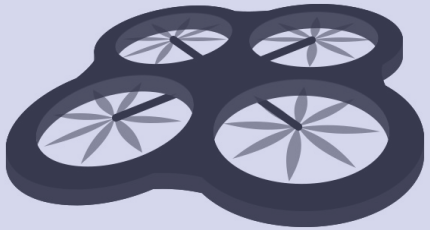
Architecture

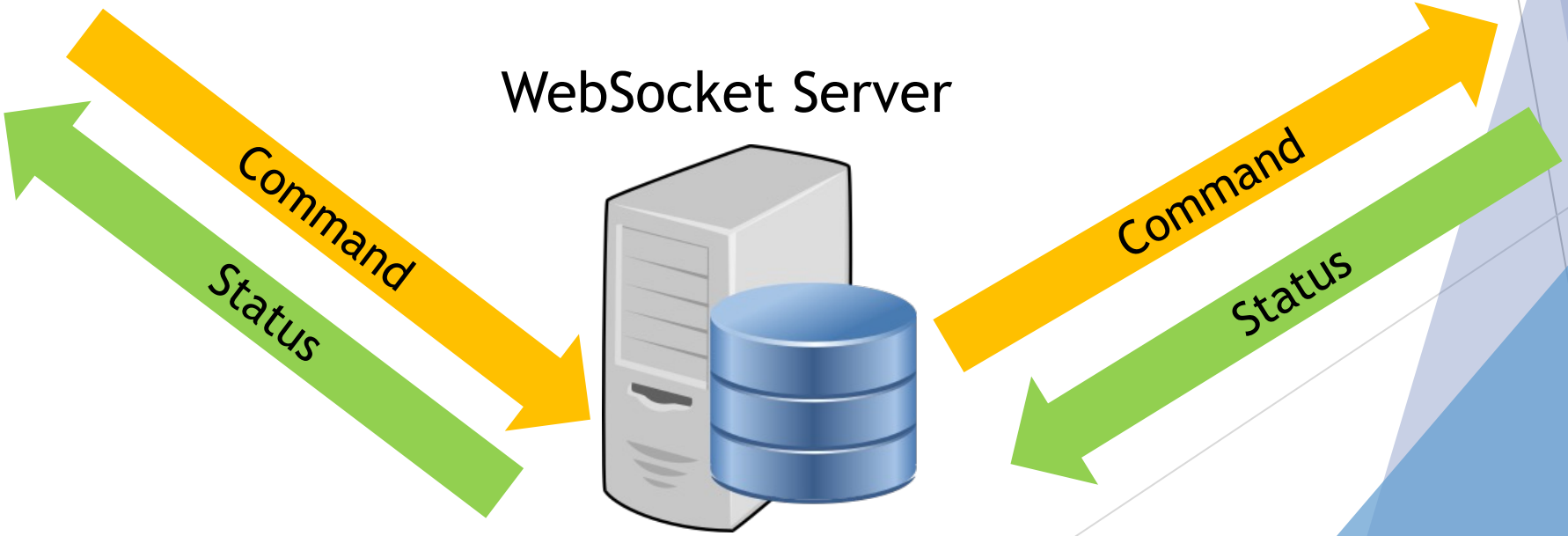
Remote Control Panel

Send Control Command
Receive the status

AndroidCopter

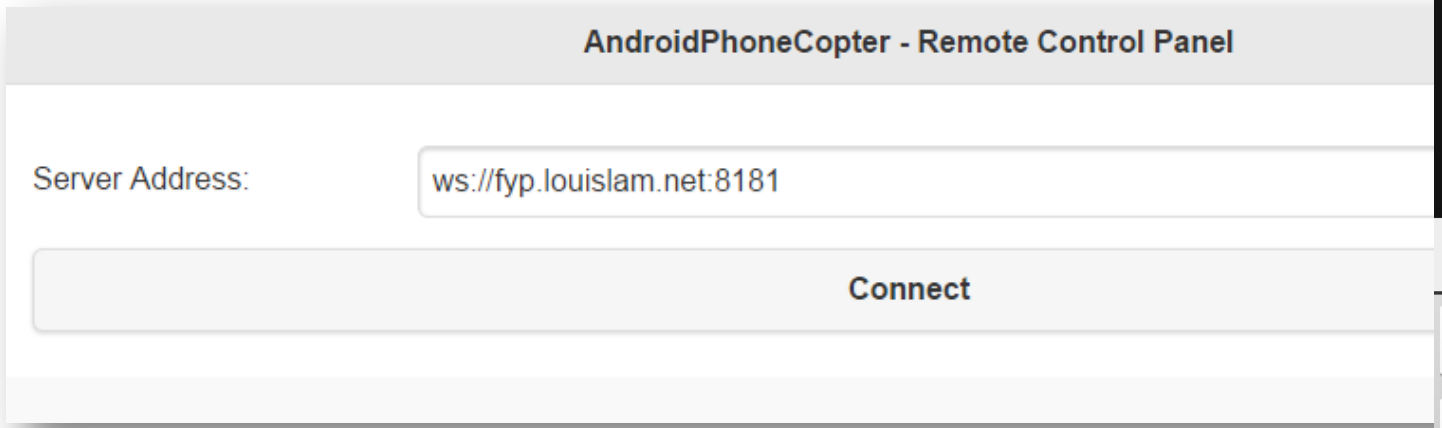
Receive Control Command
Send the status

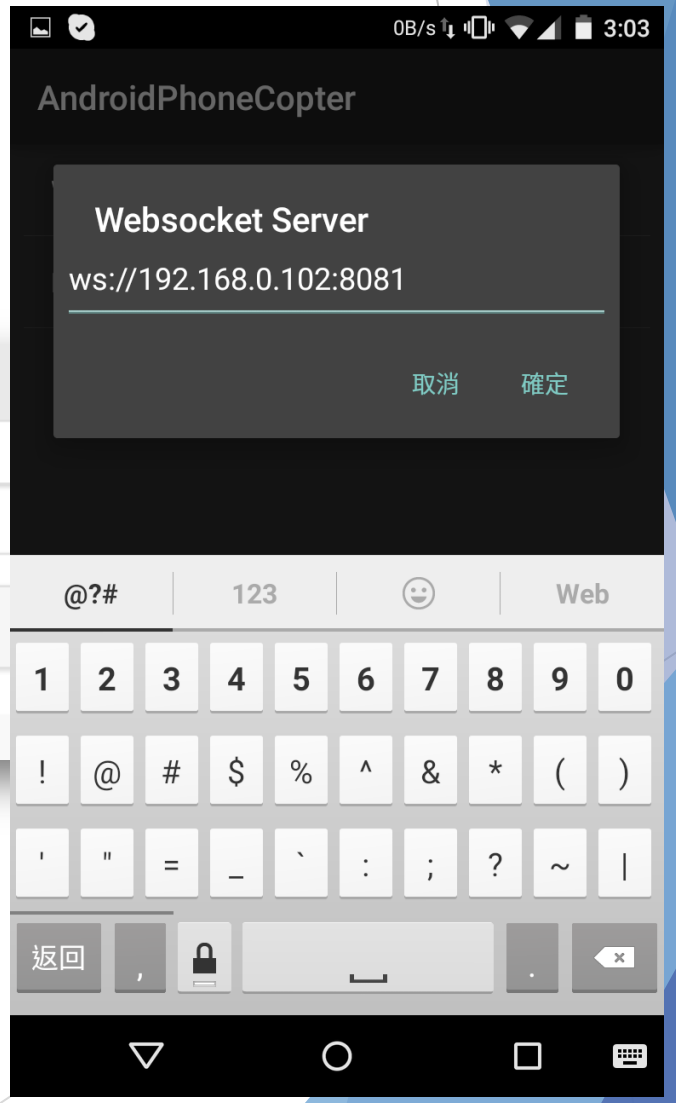




Remote Control Panel - Connect Page

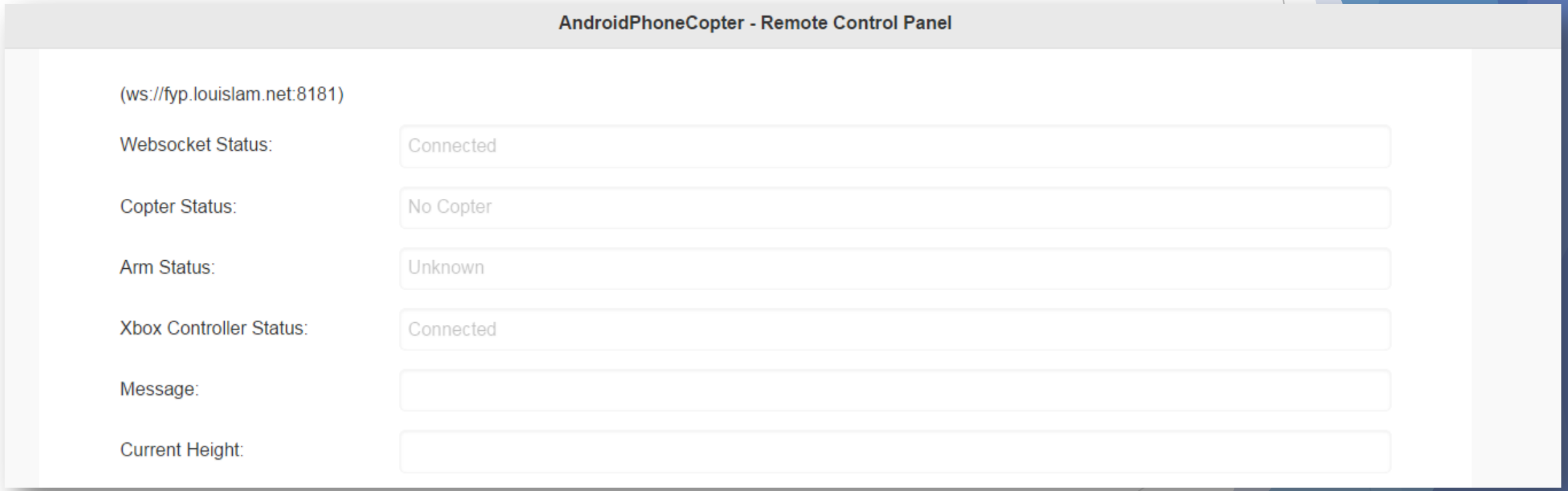
- Enter the address of the Websocket Server
- Address format is ws://hostname:port





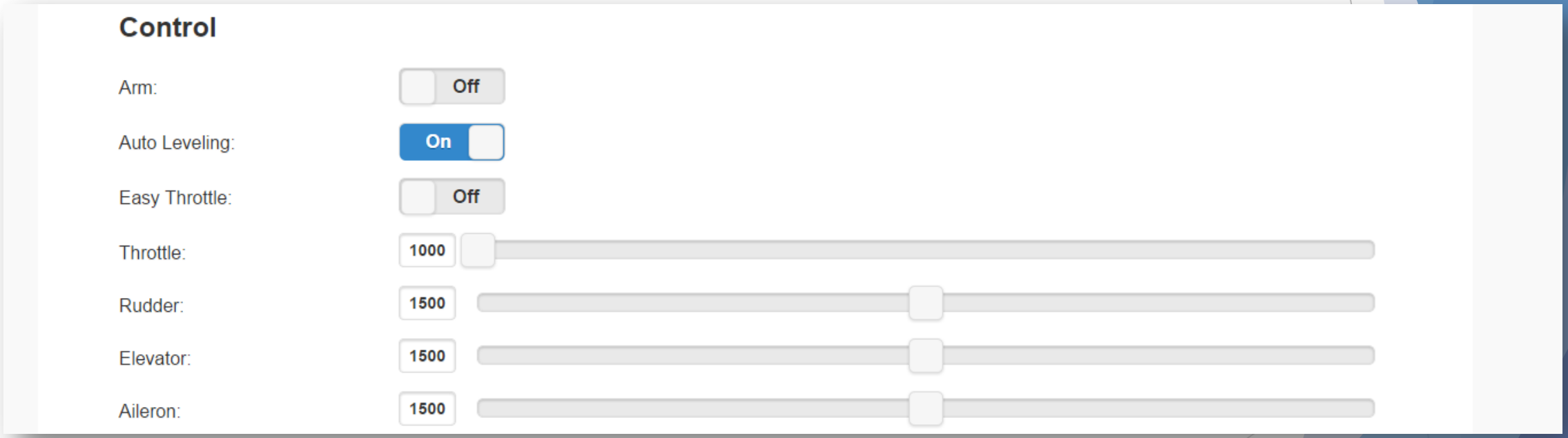
Remote Control Panel - Control Page

- Provides important information



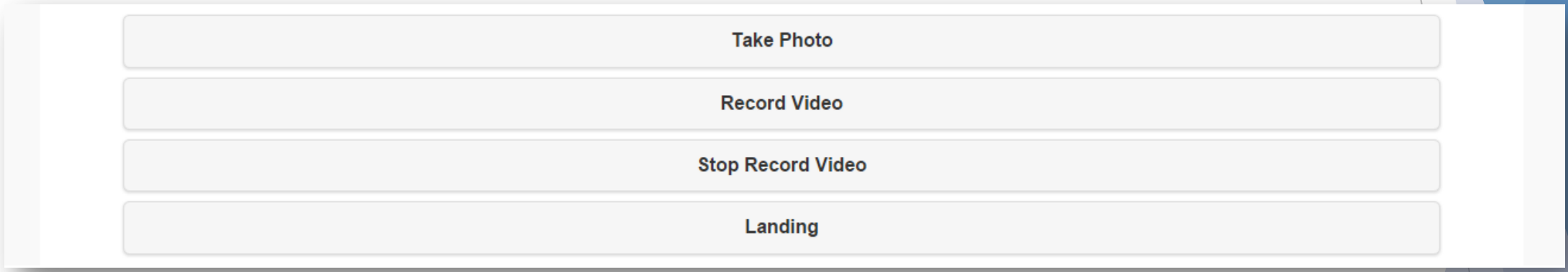
Remote Control Panel - Control Page

- Control the copter's motion
- Control using joystick



Remote Control Panel - Control Page

- Triggers Functions like Photo taking and video recording



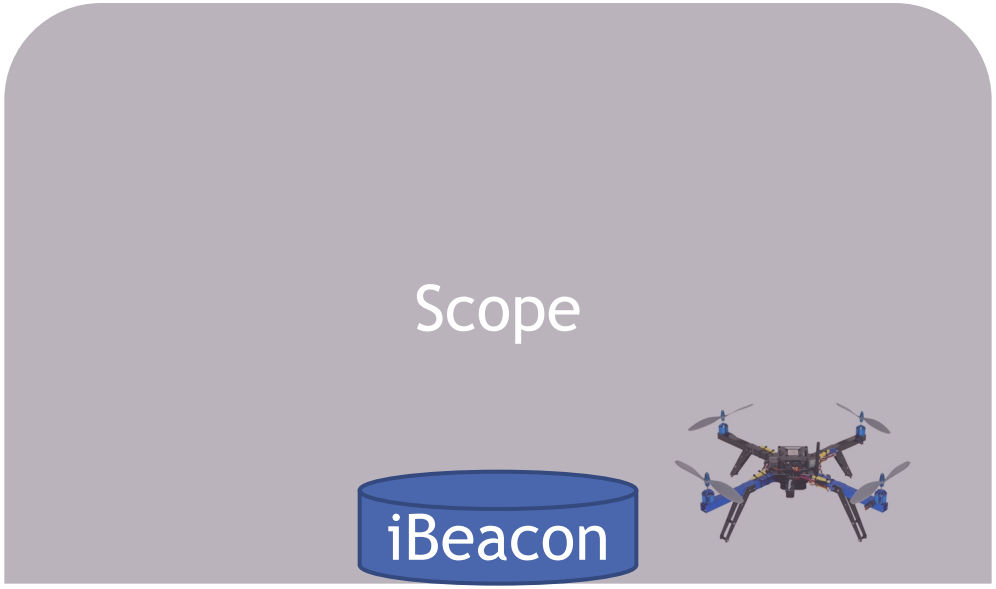
5. How to keep it safe?

Arming and Disarming Features

- Arming the motors when no people is nearby
- Disarming the motors when it is safe to get close to

Ranging Limitation using iBeacon

- Get the approximate distance from the iBeacon
- Will not fly away easily in the open area



Out of Scope/iBeacon off=>
Auto Landing

6. How is the AndroidCopter API?

AndroidCopter API

- Encapsulating all flight control logic into a object
- Other developers can develop their applications easily

- (PS. Full API are in the report)

	Send a command or message with specific value to the Remote Control Panel.
void	send(JSONObject json) Send commands or messages in JSON format.
void	reloadSettings() If some values are changed, this method should be invoked to apply changes.
void	enableAutoLeveling(boolean va) This method can be invoked to enable or disable Auto Leveling function of the AndroidCopter. An accelerometer is required in order to use this feature.
void	autoLanding() The AndroidCopter will go back to the ground after this method is called. Once this method is invoked, other commands cannot interrupt it until it lands on the ground. A sonic sensor and a barometer are required.

Example Usage

Fly to 100cm and Take a Photo

```
FlightControl flightControl = new IOIOFlightControl(this);
AndroidCopter copter = AndroidCopter.getInstance(this, flightControl);

copter.setLooper(new AndroidCopterLooper () {
    public void loop() {
        if (copter.getHeight() < 100) {    // if height < 100
            copter.setHeight(100);
        } else {                           // if height >=100cm
            copter.takePhoto();
            copter.autoLanding();
            copter.endLooper();
        }
    }
});
```

Open Source - MultiWii in C++

```
// ******** Main Loop *********
void loop () {
  static uint8_t rcDelayCommand; // this indicates the number of time (multiple of RC measurement a
  static uint8_t rcSticks;       // this hold sticks position for command combos
  uint8_t axis,i;
  int16_t error,errorAngle;
  int16_t delta;
  int16_t PTerm = 0,ITerm = 0,DTerm, PTermACC, ITermACC;
  static int16_t lastGyro[2] = {0,0};
  static int16_t errorAngleI[2] = {0,0};
  #if PID_CONTROLLER == 1
  static int32_t errorGyroI_YAW;
  static int16_t delta1[2],delta2[2];
  static int16_t errorGyroI[2] = {0,0};
  #elif PID_CONTROLLER == 2
  static int16_t delta1[3],delta2[3];
  static int32_t errorGyroI[3] = {0,0,0};
  static int16_t lastError[3] = {0,0,0};
  int16_t deltaSum;
  int16_t AngleRateTmp, RateError;
  #endif
  static uint16_t rcTime  = 0;
  static int16_t initialThrottleHold;
  int16_t rc;
  int32_t prop = 0;

  #if defined(SERIAL_RX)
    if (spekFrameFlags == 0x01) readSerial_RX();
  #endif
  #if defined(OPENLRSv2MULTI)
    Read_OpenLRS_RC();
  #endif

  #if defined(SERIAL_RX)
  if ((spekFrameDone == 0x01) || ((int16_t)(currentTime-rcTime) >0 )) {
    spekFrameDone = 0x00;
  #else
  if ((int16_t)(currentTime-rcTime) >0 ) { // 50Hz
  #endif
    rcTime = currentTime + 20000;
    computeRC();
    // Failsafe routine - added by MIS
    #if defined(FAILSAFE)
      if ( failsafeCnt > (5*FAILSAFE_DELAY) && f.ARMED) {                  // Stabilize, and set Th
        for(i=0; i<3; i++) rcData[i] = MIDRC;                               // after specified guar
        rcData[THROTTLE] = conf.failsafe_throttle;
        if (failsafeCnt > 5*(FAILSAFE_DELAY+FAILSAFE_OFF_DELAY)) {          // Turn OFF motors afte
          go_disarm();     // This will prevent the copter to automatically rearm if failsafe shuts
```

No API, not easy to follow

Difficulty of the project

- Testing Environment

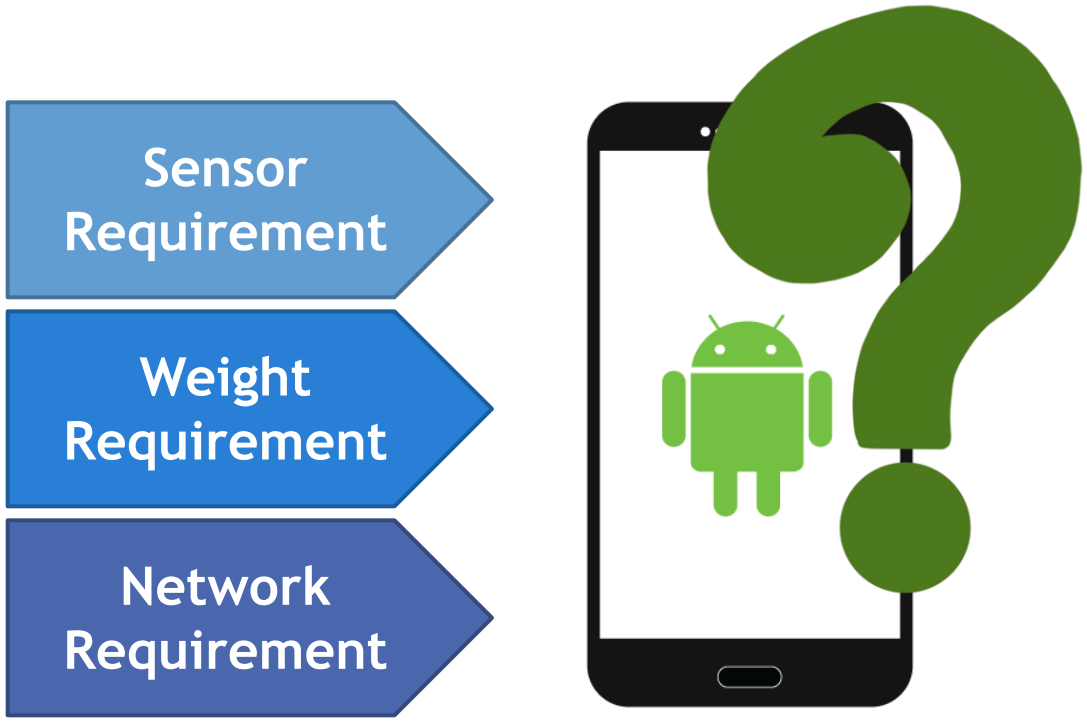
Huge noise

Risky

Large and open area

Current Limitations of AndroidCopter

- Hardware Dependencies



Current Limitations of AndroidCopter

- Short Flying Time

Limited capacity of power source

Weight of Android phone

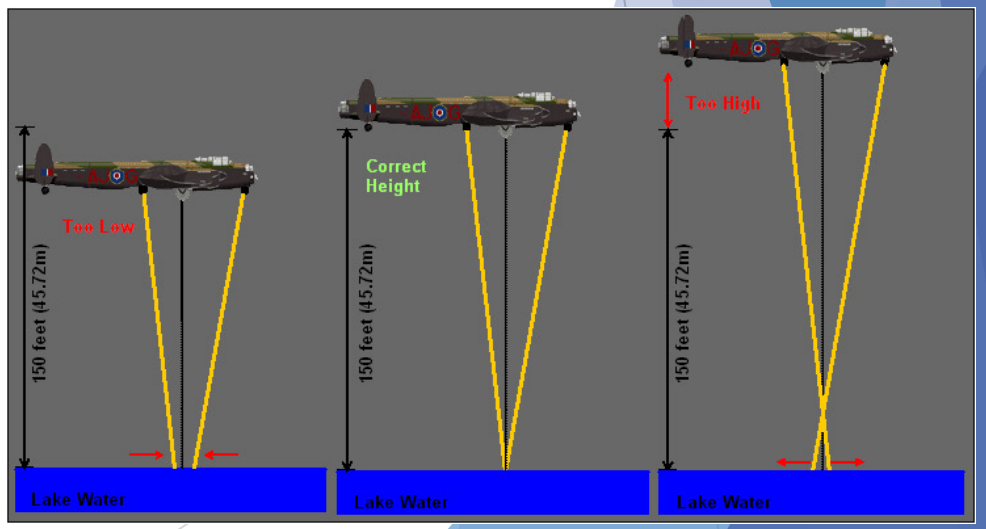
Conclusion

- We have built the first version of AndroidCopter
  - Android App (Java + IOIO)
  - Remote Control Panel (WebSocket + PHP)

- Using Android phone as a flight controller is possible.

Future works in next term

- Several Directions:
  1. Auto Pilot
  2. Optical Flow Algorithm - greatly improve the stabilization.
  3. Indoor Navigation using iBeacon
  4. Selfie



Episode


所有應用程式
新增應用程式
篩選器
應用程式名稱
價格
目前安裝次數 / 安裝總次數
平均評分 /
總評分次數: #
免費
—
—
AndroidCopter (WIP/Preview) 1.11
免費
52
239
5.00 / 2

Demonstration

Because it is too dangerous to fly in a small room, we just show you how to control the AndroidCopter.

The End

Thank you very much!