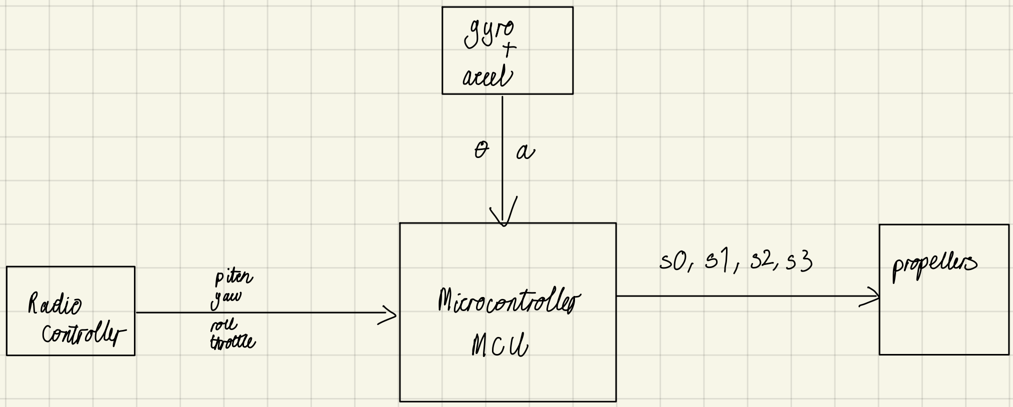
Quadcopter

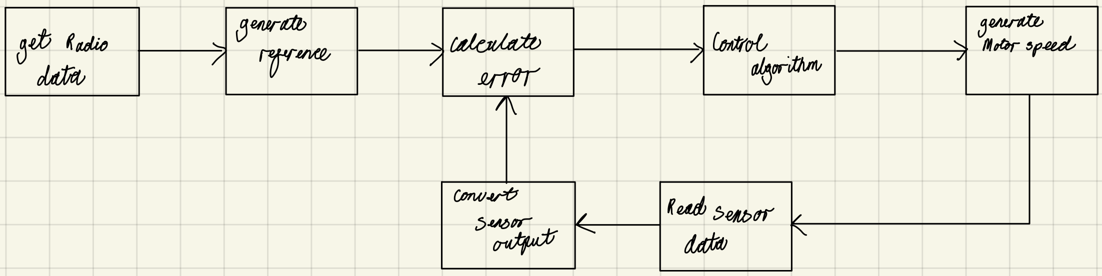
Self balancing drone

- Fly via Remote
- Hover mode

Functional Block Diagram



MCU Block Diagram



Control Algorithm

- PID
- Fuzzy Logic

List of Materials

- Power supply - rechargeable battery [ratings] ✓
  + charger
- 4 motors - propulsion [ESC - electronic speed control] 150 loading...
- PCB ✓
- D battery ✓
- wires ✓
- RC controller 80 loading...

Have

Frame ✓
GPS module
micro-controller ✓
MPU sensor ✓

Cost: $ 300

est

→ $320

360.94 - amazon cart ✓ 376.86 1 month MAX shipping
205.00 - Off garments cart - 231.65

=> 565.94 Total
608.51 (+Tax)

Birthday

- Drone [$250] 1st
- clothes [$400+] 2nd
- Celebration [$100] 3rd → $750

clothes: slippers - crocs/slides - <80
T-shirts - Off garments - 100 | minimal graphic Tee 50 → 170
fragrance - acqua Di Gio - 125

Connectors & Integration

battery connector
Extension cord to work desk
✓ >4 AA batteries

* Barometric pressure sensor
  - height data
* Magnetometer - compass

Project Notes

- Hardware configuration & device compatibility

To config: - MPU 6050
- GPS module
- RC controller

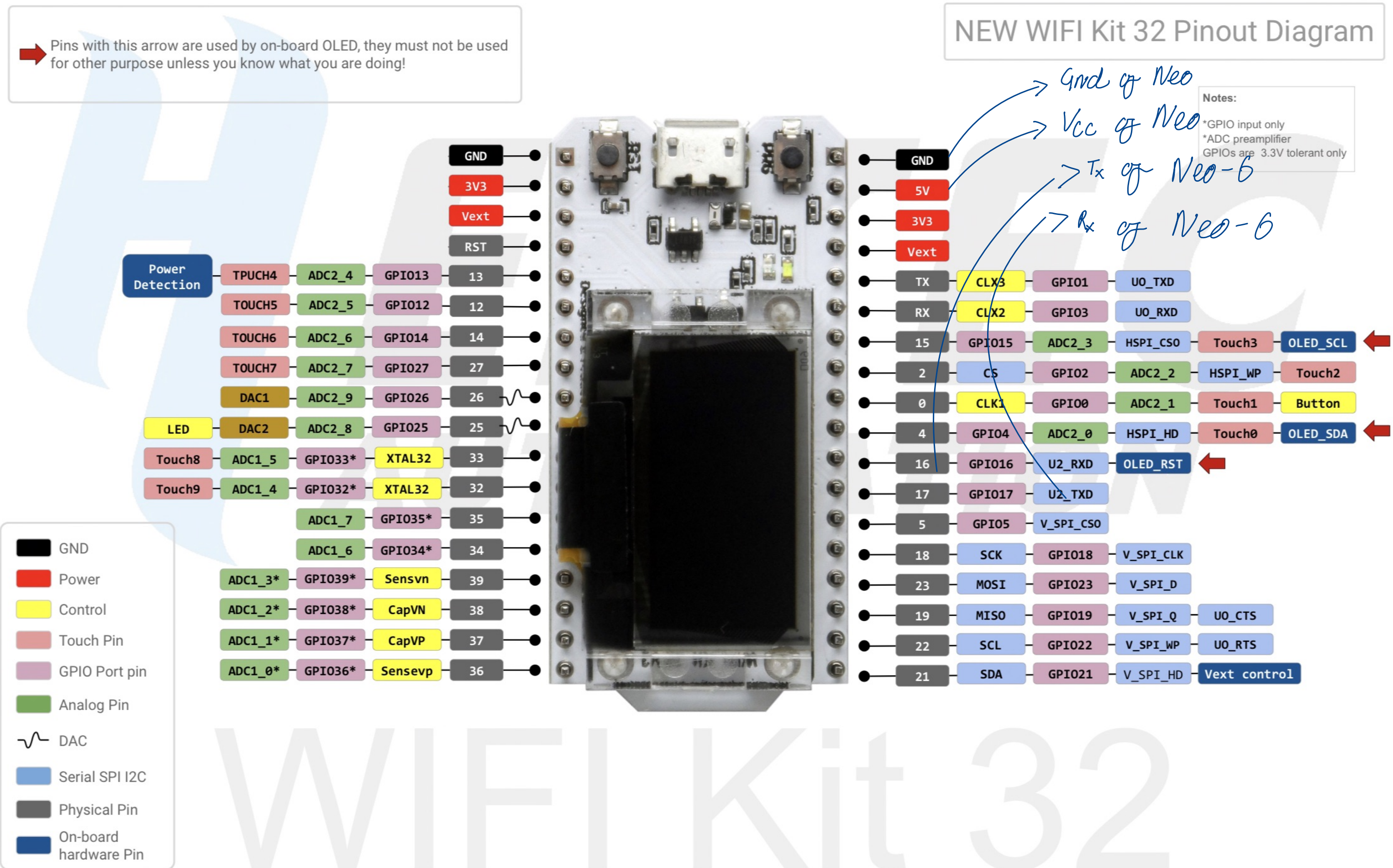
Issues

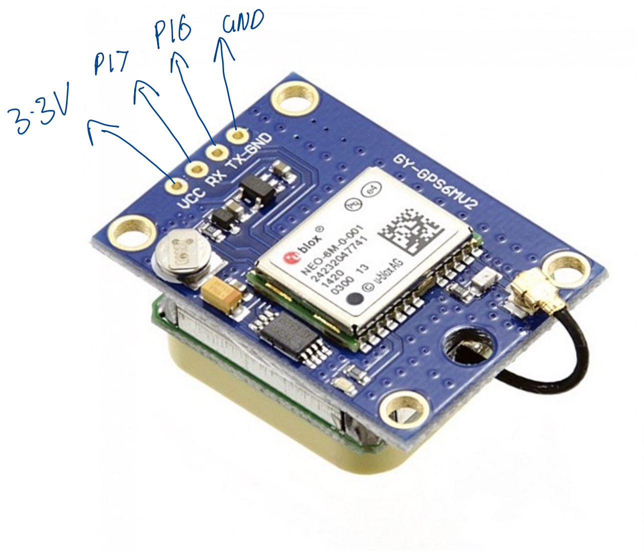
- When Tx / Rx pin from GPS module connected to Rx / Tx of ESP-32, serial communication to PC hindered

Config central Logic unit [CLU];
Stage 1 - configure CLU interface
- GPS module
1 - MPU 6050 module - write self test program
- RC interface - connector from Rx unit to microcontroller | iBus |

Resolution

- Talk to T-vax
- If no solution, talk to Paul Negedu. ✓
- Reddit, ✓



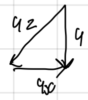


NEO-6M GPS Module

Sensor fusion

angles

$$roll = \arctan\left(\frac{-a_x}{a_z}\right)$$

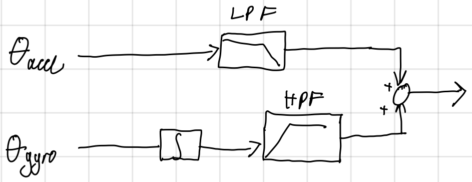


gyroscope - accumulated data error

accelerometer - high data noise

digital filters

sensor fusion



PID for speed and balance

D > P > I

Sensors - Barometric

* Orientation

MPU-6050 features

GYRO → user programmable range: ± 250, 500, 1000, 2000 (°/sec)
16 bit ADC
Digitally programmable LPF

ACCEL → programmable range: ± 2g, 4g, 8g & 16g
16 bit ADC
Tap detection
user programmable interrupts - set up for FIFO overflow!
High-G interrupt ??

+ features → XDA + XCL for additional I^2C bus reading (e.g magnetometer)

write self test program - June 28th

* pull-up resistor for SDA / SCL pins

* Change from factory trim -14% to +14% [range] - else product is damaged !!!

Neo-6 features

50 channel - 6 positioning engine

Time-to-first-fix (TTFF) → time for GPS module to acquire satellite info and calculate a position solution (fix)

TTFF - < 1 sec.

Frequency - 1,575.42 MHz

Control algorithm

Inputs: Gyro data - x, y, z
Accel data - x, y, z
Gps data - ???
ESC info - Speed (maybe V, rpm)

pinmode

serial

Outputs: ESC control voltages

Flight Control System (FCS)

Functions: Calculate altitude
" rotation position
" speed & acceleration
" direction
Control drone position, speed & direction [main]
Calculate gps location
Communicate statistics to pilot

ESP Pin config

Tx of GPS - 16
Rx of GPS - 17

SCL - 22
SDA - 21

Roll - 27
Pitch - 14
Throttle - 12
Yaw - 13

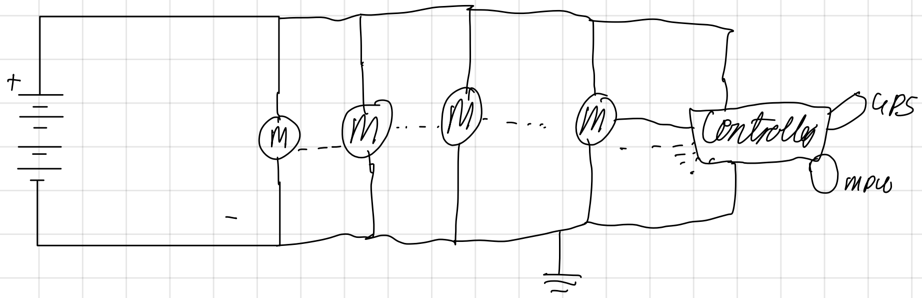
ESC0 - 36
ESC1 - 37
ESC2 - 38
ESC3 - 39

Hardware Configuration

Needed:

Battery XT90 connector / cut off connector

How to connect 1 battery to 4 motors - parallel



- ESC connectors: what to where

  blue connectors to motor, don't matter how
  White analog input to ESP
  Black ground

Sorted
- ESP to Battery

Connected to controller

MPU (4) - SCL, SDA, 3 V, GND
GPS (4) - Tx, Rx, 5V, GND
Remote receiver (4) - Yaw, pitch, roll, Throttle
ESC signal (4) - speed control 1, 2, 3, 4

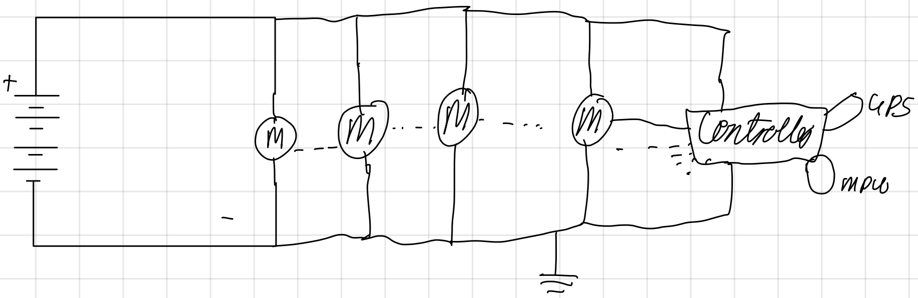
To Find: How to connect Remote receiver to ESP → RC jumpers to ESP pins

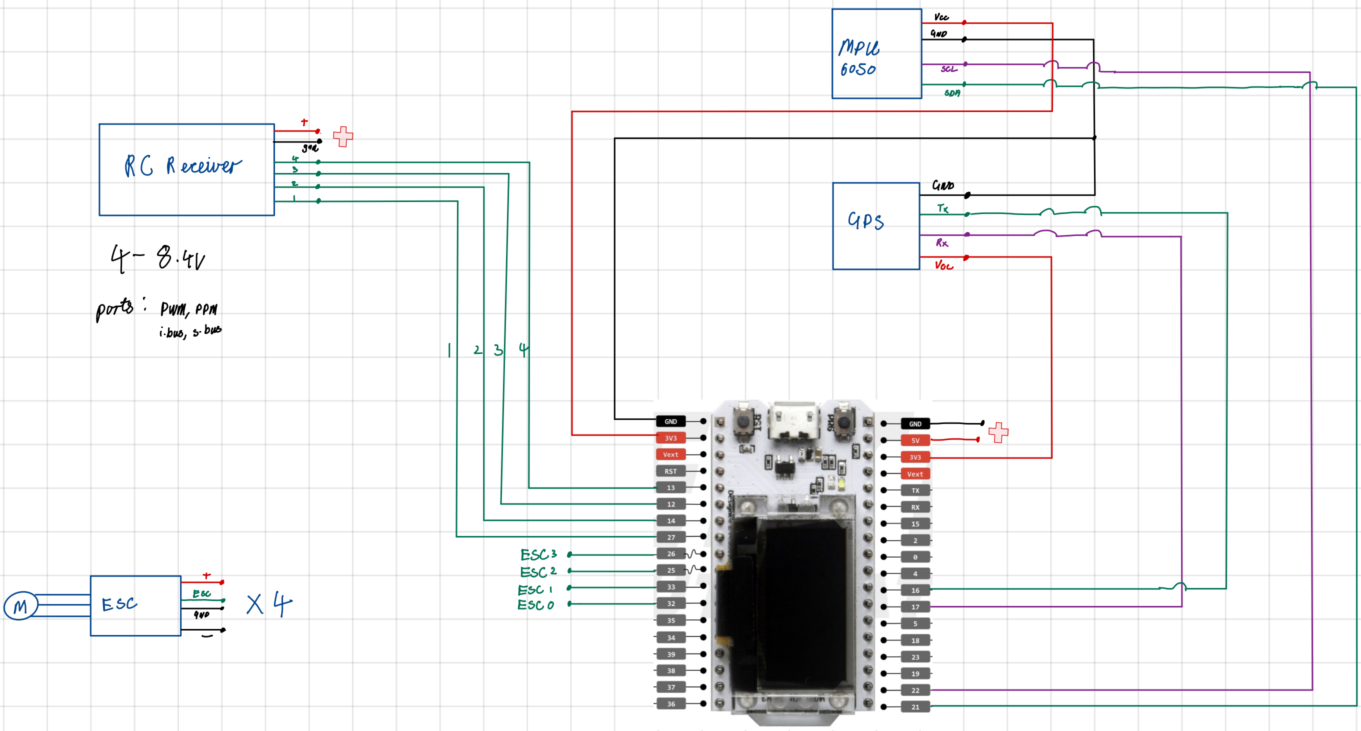
White - ESP
Black - GND
Red - 5V

How to wire an ESC to ESP ✓
" " Battery ✓
" " Motor ✓

complete

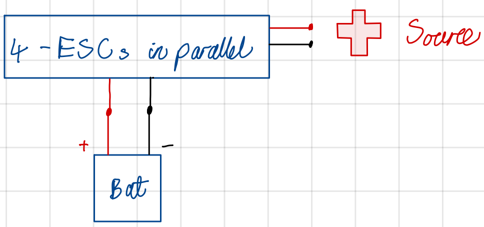
Drone Electrical Schematics





4- 8.4V

ports : PWM, PPM
i-bus, s-bus



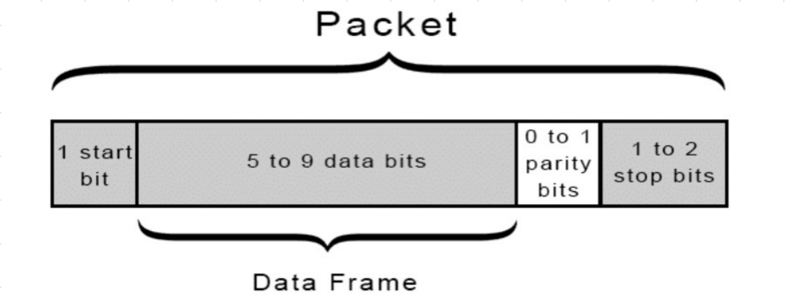
11.1V to 5V

ESP servo available on - 2, 4, 5, 12-19, 21-23, 25-27, 32-33

Connection to ESP

1 MPU - 6050 done

2 Neo 6 GPS done → UART protocol library
Registers to read
decode the Rx data



stored in UART buffer

data transmitted in frames

* UART baud rate

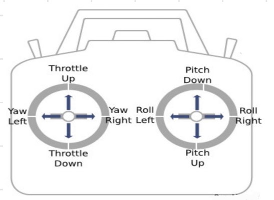
LSB transmitted first

* Serial Library - for UART comms → A function

Software serial

done

3 Radio Receiver - 4 channels
- Yaw - spin L ↔ Ch 4
- Roll - side to side (longitudinal) R ↔ Ch 1
- Pitch - forward / backward R ↕ Ch 2
- Throttle - z-axis L ↕ Ch 3



4 ESC + Motor - done

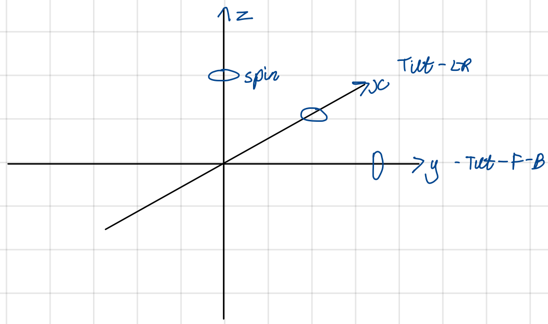
- Numbered side up - B - Y - R
ESC Motor

ESP servo available on - 2, 4, 5, 12-19, 21-23, 25-27, 32-33

$\boxed{\text{250 grams}}$

Sensor Fusion

- Angular velocity : G_x, G_y, G_z
- Linear acceleration : A_x, A_y, A_z



Kalman Filter

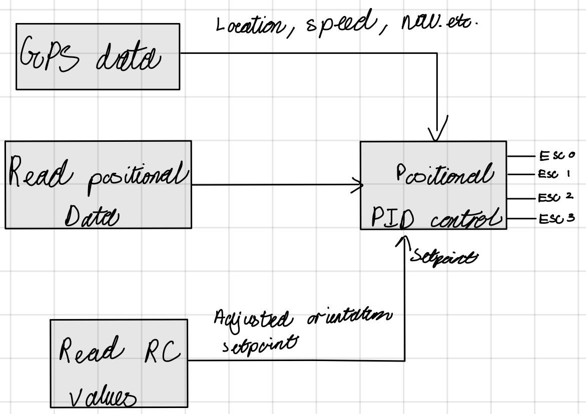
Outputs : precise angle - θ_x, θ_y, θ_z

Algorithm Flow Chart

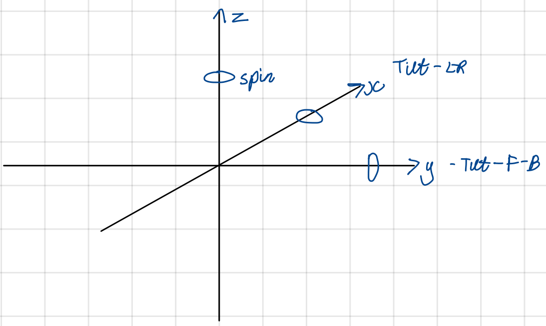
Motor + ESC operations

→ Motor start-up [within setup() func.]

→ Motor operations [RC control] - manual / MCS mode
Auto mode { [steady control] - hover
[Directed control] - GPS
} [called in Main() func.]

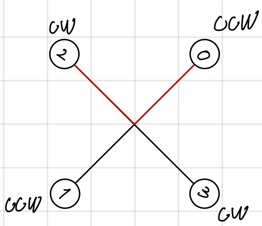


Modes



Acceleration : Drone's Positional
Rotation : Drone's Orientation

$$\theta_x = -m_0 + m_1 + m_2 - m_3$$
$$\theta_y = m_0 - m_1 + m_2 - m_3$$
$$\theta_z = -m_0 - m_1 + m_2 + m_3$$



pitch - ↑↓ : 2-0 ↓↑ : 1-3
roll - ↑↓ : 2-1 ↓↑ : 0-3
yaw - ↑↓ : 2-3 ↓↑ : 0-1
thrust - ↑↓ : 0-1-2-3

$$m_0 = Th - Y + P - R$$
$$m_2 = Th + Y + P + R$$
$$m_1 = Th - Y - P + R$$
$$m_3 = Th + Y - P - R$$

— ①

RC control

Lower / Leftwards : 999
Middle : 1499 / 1500
Upper / Rightwards : 2000

GPS data

main output : - longitude
- lattitude

ESC

starts running @ 88 i $^{-50}$ = 38

1040 + 35 → 1075   30

1075   5

start − 10
1040 + 30 + 5 =   ← 10   stop − 10

11.1 V

Min speed − 9 + 10

2200 KV → 2200 rpm per 1V

Orientation & Position Controller

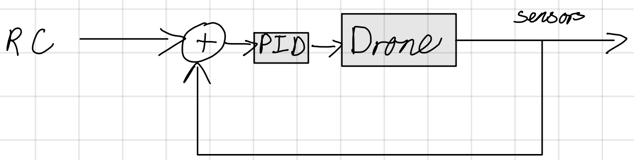
Inputs: Angular velocity : G_x, G_y, G_z } $\theta_x, \theta_y, \theta_z$] – balancing
Linear acceleration : A_x, A_y, A_z
Throttle : Th
Yaw : Y
Pitch : P] – Adjust behavior /setpoint
Roll : R

$\Theta = [\theta_x, \theta_y, \theta_z]$

$RC = [Th, P, R, Y]$

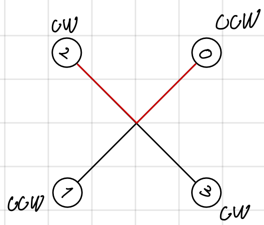
Outputs: Motor Speed : m_0, m_1, m_2, m_3 || $M = [m_0, m_1, m_2, m_3]$

Controller: PID or Fuzzy-PID



Weight : 1,179.34g

Controllers:
- Fuzzy control / Fuzzy-PID
- PID



pitch – ↑↓: 2–0 ↓↑: 1–3
roll – ↑↓: 2–1 ↓↑: 0–3
yaw – ↑↓: 2–3 ↓↑: 0–1
thrust – ↑↓: 0–1–2–3

$$m_0 = Th - Y + P - R$$
$$m_2 = Th + Y + P + R$$
$$m_1 = Th - Y - P + R$$
$$m_3 = Th + Y - P - R$$

Motor Mixing

Relation between Θ and M

$$\theta_x = -m_0 + m_1 + m_2 - m_3$$
$$\theta_y = m_0 - m_1 + m_2 - m_3$$
$$\theta_z = -m_0 - m_1 + m_2 + m_3$$

Model for Quadcopter

Input : motor speed : 4

Output : Gyro + accelerometer : 6

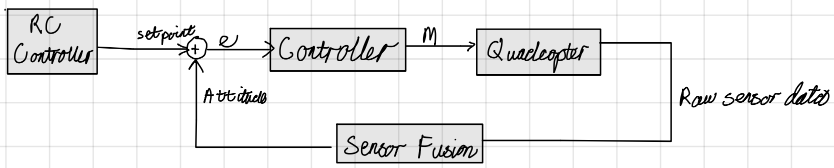
Sensor Fusion

Complementry filter for gyro and accelero

Kalman Filter

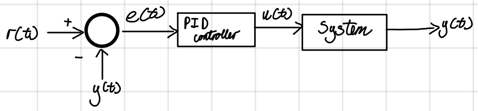
Speed

$M = [m0, m1, m2, m3]$



My design : Controller [PID or Fuzzy PID]
Sensor fusion
Set point adjustment [RC]

PID Control Algoritm



$$e(t) = r(t) - y(t)$$

$$u(t) = K_p e(t) + K_i \int e(t)\,dt + K_D \frac{de}{dt}$$

y(t) - sensor output, θ

r(t) - reference position, 0 (zero)

r(t) - reference

y(t) - sensor output

e(t) - error signal

u(t) - input to system

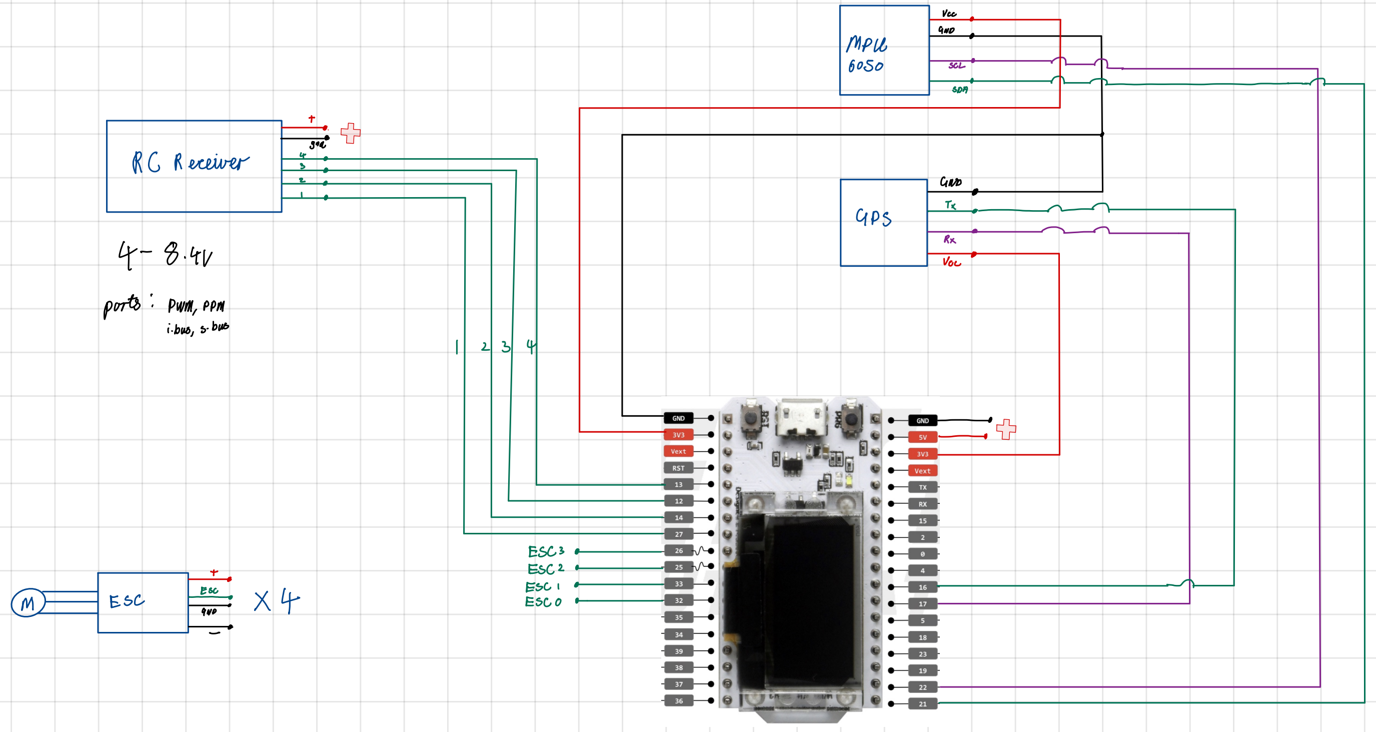
Software

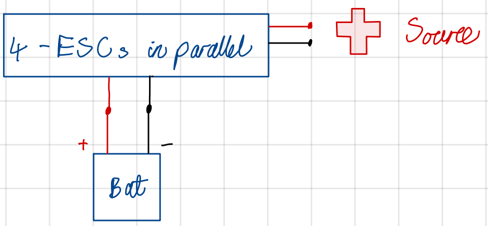
1 file : main.c / main.ino

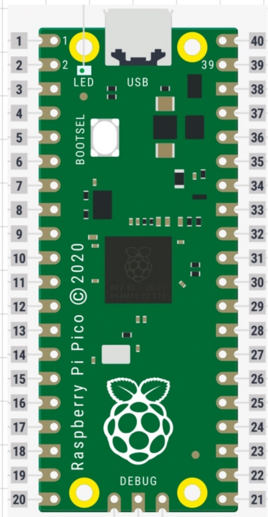
Setup () {
- MPU 650
- Motors
- RC
- Reset variables
}

Main () {
- Read sensor
- Calculate error
- Control PID
}

ISR () {
- Read RC
- Calculate M
}







32 / GP27
31 / GP 26