### **1. Dedicated Gimbal Controller (Simpler Setup)**

#### Components:

* Brushless motor
* Magnetic encoder (e.g., AS5048A)
* Gimbal controller (e.g., SimpleBGC)
* IMU (e.g., MPU6050)

#### Connections:

1. Motor to Controller:
   * The brushless motor has 3 wires (phases) connected to the motor output terminals on the gimbal controller.
   * These are typically labeled as M1, M2, etc., for each axis.
2. Encoder to Controller:
   * Connect the encoder to the controller's encoder input port:
     + Power (VCC): Connect to the controller's 3.3V or 5V supply.
     + Ground (GND): Connect to the controller's ground.
     + SPI or I2C lines: Use the controller’s interface to connect:
       - MISO, MOSI, SCK, CS for SPI
       - SDA, SCL for I2C.
3. IMU to Controller:
   * The IMU (Inertial Measurement Unit) connects via I2C to provide motion feedback.
4. Power Supply:
   * Provide a suitable voltage (often 2S–3S LiPo battery) to the gimbal controller.

### **2. Custom Setup with STM32**

#### Components:

* STM32 Microcontroller
* Magnetic encoder (e.g., AS5048A)
* Motor driver IC (e.g., DRV8313 or L6234)
* Brushless motor
* IMU (e.g., MPU6050)

#### Connections:

1. Brushless Motor to Motor Driver:
   * Connect the motor’s three-phase wires to the motor driver's outputs (U, V, W).
   * Power the motor driver with an appropriate voltage (e.g., 7.4V–12V for most gimbal motors).
2. Motor Driver to STM32:
   * The STM32 generates PWM signals to control the motor driver:
     + PWM1, PWM2, PWM3: Send these signals to the motor driver for the three phases.
     + Enable: If required, connect an enable signal from STM32 to the motor driver.
3. Encoder to STM32:
   * Use the encoder for position feedback:
     + Power (VCC): Connect to STM32’s 3.3V or 5V pin.
     + Ground (GND): Connect to STM32’s GND.
     + SPI/I2C:
       - For SPI: Connect MISO, MOSI, SCK, CS to STM32 SPI pins.
       - For I2C: Connect SDA and SCL to STM32 I2C pins.
4. IMU to STM32:
   * Connect the MPU6050 to STM32 via I2C:
     + SDA and SCL to the corresponding STM32 I2C pins.
5. Power Supply:
   * Provide power to the motor driver and STM32 (via a regulated 5V supply or battery).
6. Software Integration:
   * Write firmware for the STM32 to:
     + Read encoder data for position feedback.
     + Process IMU data for motion feedback.
     + Generate PWM signals to control the motor driver.

### **Wiring Diagram (Custom STM32 Setup)**

plaintext

Copy code

[Brushless Motor]

| | |

| | | --> (U, V, W) --> [Motor Driver (DRV8313/L6234)]

| | |

[Encoder (AS5048A)] --> (SPI/I2C) --> [STM32]

| |

[IMU (MPU6050)] ----------- (I2C) -------|

|

[Power Supply] ------------ (VCC, GND)

### **Example Usage**

1. Power up the system.
2. The encoder sends position feedback to the controller or STM32.
3. The IMU provides motion data to stabilize the gimbal.
4. The controller processes the feedback and adjusts the motor signals to maintain stabilization.

### **Driver Board controller**

PWM Signal to control motors

### **Pin Definitions**

1. FC Roll / FC Pitch
   * FC stands for Flight Controller.
   * These pins are used for communication between the gimbal controller and a flight controller in drone setups.
   * Purpose:
     + Allows the flight controller to command the gimbal to adjust its roll and pitch angles dynamically.
     + Used in scenarios where the gimbal orientation must be synchronized with the drone's movement.
   * Type of Signal:
     + PWM or analog signals, where the signal value corresponds to a desired angle.
2. RX Roll / RX Pitch
   * RX stands for Receiver.
   * These pins are used for direct control of the gimbal axes (roll and pitch) from an RC receiver.
   * Purpose:
     + Allows the user to manually control the roll and pitch angles via an RC transmitter (e.g., to point the camera in a specific direction).
   * Type of Signal:
     + PWM signals from the RC receiver, where the duty cycle corresponds to the desired angle.