

// 8 Channel Transmitter (No Trim) | 8 Kanal Verici (Trim Yok)

// Input pin A5

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

const uint64\_t pipeOut = 000322; // NOTE: The same as in the receiver 000322 | Alıcı kodundaki adres ile aynı olmalı

RF24 radio(9, 10); // select CE,CSN pin | CE ve CSN pinlerin seçimi

struct Signal {

byte throttle;

byte pitch;

byte roll;

byte yaw;

byte aux1;

byte aux2;

byte aux3;

byte aux4;

};

Signal data;

void ResetData()

{

data.throttle = 0;

data.pitch = 127;

data.roll = 127;

data.yaw = 127;

data.aux1 = 0; // Signal lost position | Sinyal kesildiğindeki pozisyon

data.aux2 = 0;

data.aux3 = 0;

data.aux4 = 0;

}

void setup()

{

//Configure the NRF24 module | NRF24 modül konfigürasyonu

radio.begin();

radio.openWritingPipe(pipeOut);

radio.setAutoAck(false);

radio.setDataRate(RF24\_250KBPS); // The lowest data rate value for more stable communication | Daha kararlı iletişim için en düşük veri hızı.

radio.setPALevel(RF24\_PA\_MAX); // Output power is set for maximum | Çıkış gücü maksimum için ayarlanıyor.

radio.stopListening(); // Start the radio comunication for Transmitter | Verici için sinyal iletişimini başlatır.

ResetData();

}

// Joystick center and its borders | Joystick merkez ve sınırları

int Border\_Map(int val, int lower, int middle, int upper, bool reverse)

{

val = constrain(val, lower, upper);

if ( val < middle )

val = map(val, lower, middle, 0, 128);

else

val = map(val, middle, upper, 128, 255);

return ( reverse ? 255 - val : val );

}

void loop()

{

// Control Stick Calibration for channels | Her bir kanal için kumanda Kol Kalibrasyonları

data.roll = Border\_Map( analogRead(A3), 0, 512, 1023, true ); // "true" or "false" for signal direction | "true" veya "false" sinyal yönünü belirler

data.pitch = Border\_Map( analogRead(A2), 0, 512, 1023, true );

data.throttle = Border\_Map( analogRead(A1),570, 800, 1023, false ); // For Single side ESC | Tek yönlü ESC için

// data.throttle = Border\_Map( analogRead(A1),0, 512, 1023, false ); // For Bidirectional ESC | Çift yönlü ESC için

data.yaw = Border\_Map( analogRead(A0), 0, 512, 1023, true );

data.aux1 = Border\_Map( analogRead(A4), 0, 512, 1023, true ); // "true" or "false" for change signal direction | "true" veya "false" sinyal yönünü değiştirir.

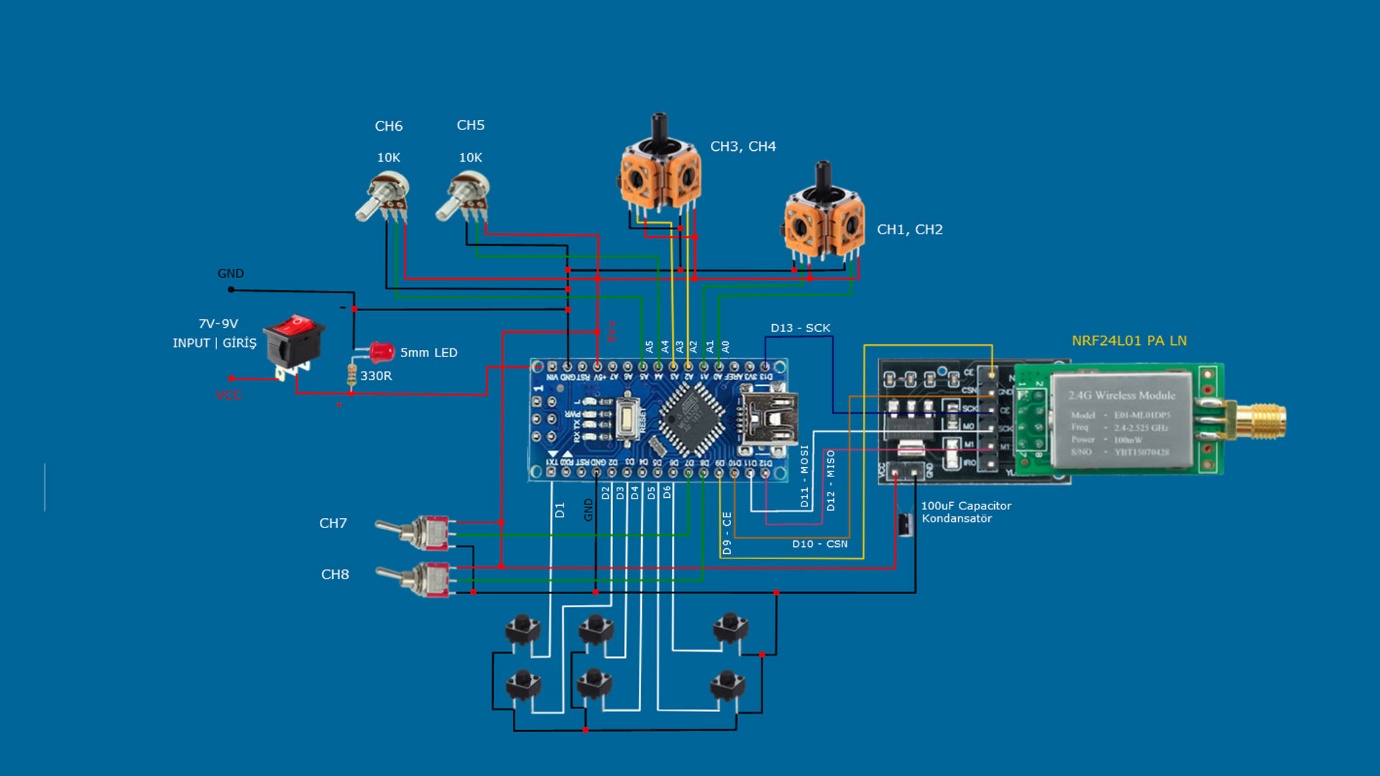
data.aux2 = Border\_Map( analogRead(A5), 0, 512, 1023, true ); // "true" or "false" for change signal direction | "true" veya "false" sinyal yönünü değiştirir.

data.aux3 = digitalRead(7);

data.aux4 = digitalRead(8);

radio.write(&data, sizeof(Signal));

}



// 8 Channel Transmitter & Trims | 8 Kanal Verici ve Trimler

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

#include <EEPROM.h>

const uint64\_t pipeOut = 000322; // NOTE: The same as in the receiver 000322 | Alıcı kodundaki adres ile aynı olmalı

RF24 radio(9, 10); // Select CE,CSN pin | CE ve CSN pinlerin seçimi

#define trimbut\_1 1 // Trim button 1 / Pin D1

#define trimbut\_2 2 // Trim button 2 / Pin D2

#define trimbut\_3 3 // Trim button 3 / Pin D3

#define trimbut\_4 4 // Trim button 4 / Pin D4

#define trimbut\_5 5 // Trim button 5 / Pin D5

#define trimbut\_6 6 // Trim button 6 / Pin D6

int tvalue1 = EEPROM.read(1) \* 4; // Reading trim values from Eprom | Trim değerlerinin Epromdan okunması

int tvalue2 = EEPROM.read(3) \* 4;

int tvalue3 = EEPROM.read(5) \* 4;

struct Signal {

byte throttle;

byte pitch;

byte roll;

byte yaw;

byte aux1;

byte aux2;

byte aux3;

byte aux4;

};

Signal data;

void ResetData()

{

data.throttle = 512; // Signal lost position | Sinyal kesildiğindeki pozisyon

data.pitch = 127;

data.roll = 127;

data.yaw = 127;

data.aux1 = 0;

data.aux2 = 0;

data.aux3 = 0;

data.aux4 = 0;

}

void setup()

{

// Configure the NRF24 module | NRF24 modül konfigürasyonu

radio.begin();

radio.openWritingPipe(pipeOut);

radio.setAutoAck(false);

radio.setDataRate(RF24\_250KBPS); // The lowest data rate value for more stable communication | Daha kararlı iletişim için en düşük veri hızı.

radio.setPALevel(RF24\_PA\_MAX); // Output power is set for maximum | Çıkış gücü maksimum için ayarlanıyor.

radio.stopListening(); // Start the radio comunication for Transmitter | Verici için sinyal iletişimini başlatır.

ResetData();

pinMode(trimbut\_1, INPUT\_PULLUP);

pinMode(trimbut\_2, INPUT\_PULLUP);

pinMode(trimbut\_3, INPUT\_PULLUP);

pinMode(trimbut\_4, INPUT\_PULLUP);

pinMode(trimbut\_5, INPUT\_PULLUP);

pinMode(trimbut\_6, INPUT\_PULLUP);

tvalue1= EEPROM.read(1) \* 4;

tvalue2= EEPROM.read(3) \* 4;

tvalue3= EEPROM.read(5) \* 4;

}

// Joystick center and its borders | Joystick merkez ve sınırları

int Border\_Map(int val, int lower, int middle, int upper, bool reverse)

{

val = constrain(val, lower, upper);

if ( val < middle )

val = map(val, lower, middle, 0, 128);

else

val = map(val, middle, upper, 128, 255);

return ( reverse ? 255 - val : val );

}

void loop()

{

// Trims and Limiting trim values | Trimler ve Trim değerlerini sınırlandırma

if(digitalRead(trimbut\_1)==LOW and tvalue1 < 630) {

tvalue1=tvalue1+15;

EEPROM.write(1,tvalue1/4);

delay (130);

}

if(digitalRead(trimbut\_2)==LOW and tvalue1 > 280){

tvalue1=tvalue1-15;

EEPROM.write(1,tvalue1/4);

delay (130);

}

if(digitalRead(trimbut\_3)==LOW and tvalue2 < 630) {

tvalue2=tvalue2+15;

EEPROM.write(3,tvalue2/4);

delay (130);

}

if(digitalRead(trimbut\_4)==LOW and tvalue2 > 280){

tvalue2=tvalue2-15;

EEPROM.write(3,tvalue2/4);

delay (130);

}

if(digitalRead(trimbut\_5)==LOW and tvalue3 < 630) {

tvalue3=tvalue3+15;

EEPROM.write(5,tvalue3/4);

delay (130);

}

if(digitalRead(trimbut\_6)==LOW and tvalue3 > 280){

tvalue3=tvalue3-15;

EEPROM.write(5,tvalue3/4);

delay (130);

}

// Control Stick Calibration for channels | Her bir kanal için kumanda Kol Kalibrasyonları

data.roll = Border\_Map( analogRead(A3), 0, tvalue1, 1023, true ); // "true" or "false" for signal direction | "true" veya "false" sinyal yönünü belirler

data.pitch = Border\_Map( analogRead(A2), 0, tvalue2, 1023, true );

data.throttle = Border\_Map( analogRead(A1),570, 800, 1023, false ); // For Single side ESC | Tek yönlü ESC için

// data.throttle = Border\_Map( analogRead(A1),0, 512, 1023, false ); // For Bidirectional ESC | Çift yönlü ESC için

data.yaw = Border\_Map( analogRead(A0), 0, tvalue3, 1023, true );

data.aux1 = Border\_Map( analogRead(A4), 0, 512, 1023, true );

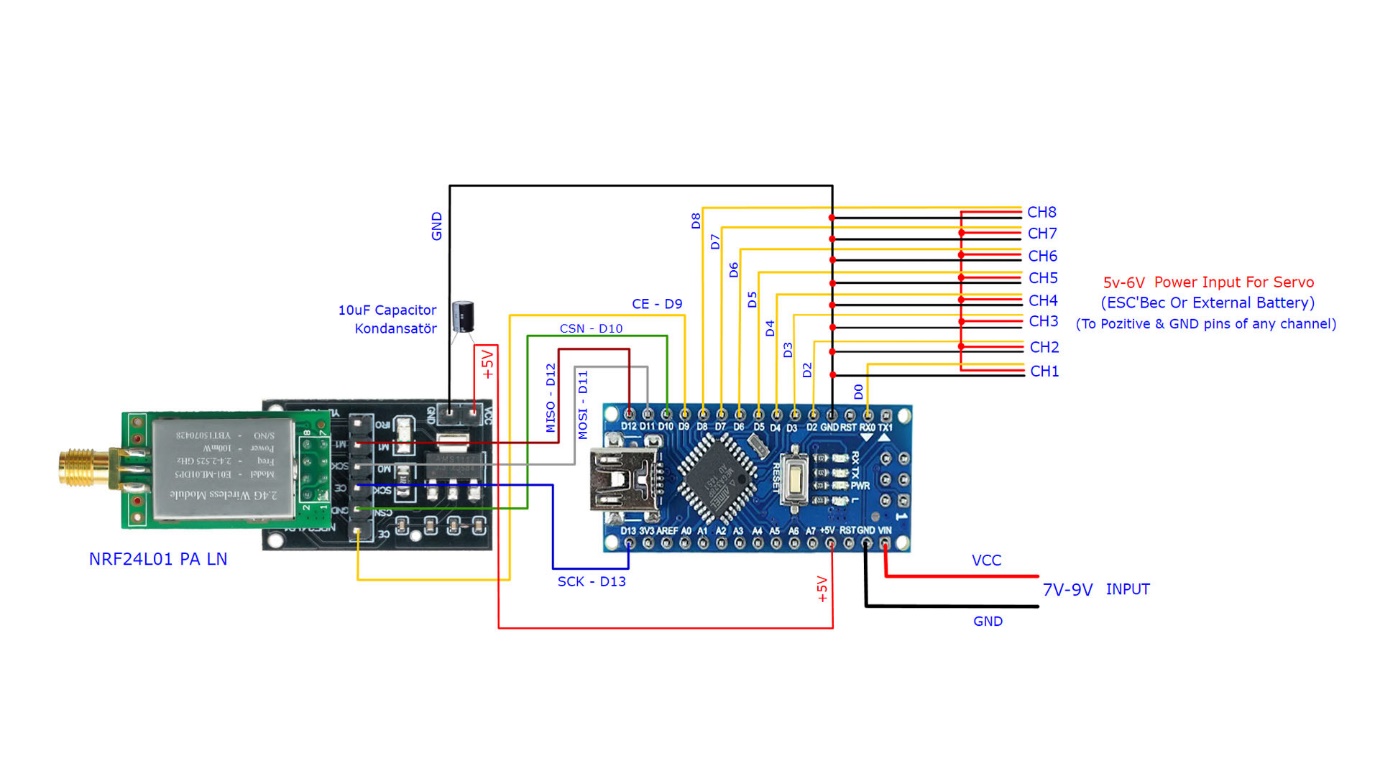
data.aux2 = Border\_Map( analogRead(A5), 0, 512, 1023, true );

data.aux3 = digitalRead(7);

data.aux4 = digitalRead(8);

radio.write(&data, sizeof(Signal));

}



// 8 Channel Receiver | 8 Kanal Alıcı

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

#include <Servo.h>

int ch\_width\_1 = 0;

int ch\_width\_2 = 0;

int ch\_width\_3 = 0;

int ch\_width\_4 = 0;

int ch\_width\_5 = 0;

int ch\_width\_6 = 0;

int ch\_width\_7 = 0;

int ch\_width\_8 = 0;

Servo ch1;

Servo ch2;

Servo ch3;

Servo ch4;

Servo ch5;

Servo ch6;

Servo ch7;

Servo ch8;

struct Signal {

byte throttle;

byte pitch;

byte roll;

byte yaw;

byte aux1;

byte aux2;

byte aux3;

byte aux4;

};

Signal data;

const uint64\_t pipeIn = 000322;

RF24 radio(9, 10);

void ResetData()

{

data.throttle = 0;

data.roll = 127;

data.pitch = 127;

data.yaw = 127;

data.aux1 = 0; // Define the inicial value of each data input. | Veri girişlerinin başlangıç değerleri

data.aux2 = 0;

data.aux3 = 0;

data.aux4 = 0;

}

void setup()

{

// Set the pins for each PWM signal | Her bir PWM sinyal için pinler belirleniyor.

ch1.attach(0);

ch2.attach(2);

ch3.attach(3);

ch4.attach(4);

ch5.attach(5);

ch6.attach(6);

ch7.attach(7);

ch8.attach(8);

ResetData(); // Configure the NRF24 module | NRF24 Modül konfigürasyonu

radio.begin();

radio.openReadingPipe(1,pipeIn);

radio.setAutoAck(false);

radio.setDataRate(RF24\_250KBPS); // The lowest data rate value for more stable communication | Daha kararlı iletişim için en düşük veri hızı.

radio.setPALevel(RF24\_PA\_MAX); // Output power is set for maximum | Çıkış gücü maksimum için ayarlanıyor.

radio.startListening(); // Start the radio comunication for receiver | Alıcı için sinyal iletişimini başlatır.

}

unsigned long lastRecvTime = 0;

void recvData()

{

while ( radio.available() ) {

radio.read(&data, sizeof(Signal));

lastRecvTime = millis(); // Receive the data | Data alınıyor

}

}

void loop()

{

recvData();

unsigned long now = millis();

if ( now - lastRecvTime > 1000 ) {

ResetData(); // Signal lost.. Reset data | Sinyal kayıpsa data resetleniyor

}

ch\_width\_1 = map(data.roll, 0, 255, 1000, 2000);

ch\_width\_2 = map(data.pitch, 0, 255, 1000, 2000);

ch\_width\_3 = map(data.throttle, 0, 255, 1000, 2000);

ch\_width\_4 = map(data.yaw, 0, 255, 1000, 2000);

ch\_width\_5 = map(data.aux1, 0, 255, 1000, 2000);

ch\_width\_6 = map(data.aux2, 0, 255, 1000, 2000);

ch\_width\_7 = map(data.aux3, 0, 1, 1000, 2000);

ch\_width\_8 = map(data.aux4, 0, 1, 1000, 2000);

ch1.writeMicroseconds(ch\_width\_1); // Write the PWM signal | PWM sinyaller çıkışlara gönderiliyor

ch2.writeMicroseconds(ch\_width\_2);

ch3.writeMicroseconds(ch\_width\_3);

ch4.writeMicroseconds(ch\_width\_4);

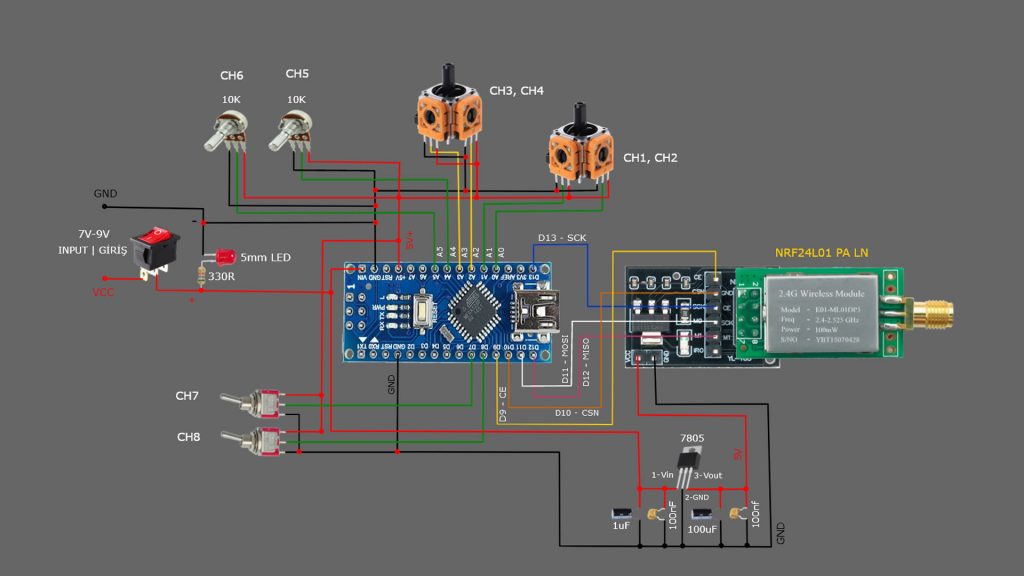
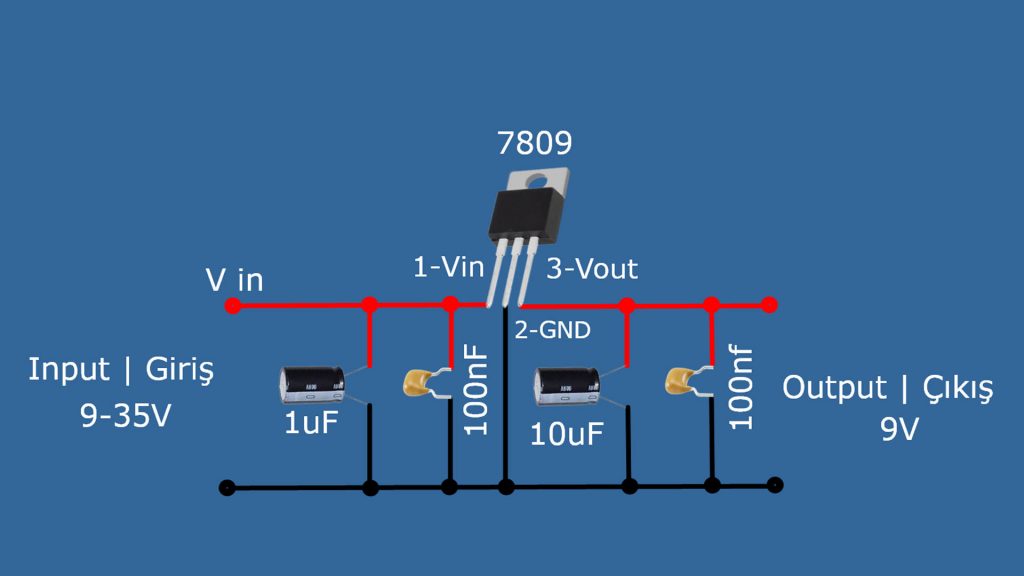
ch5.writeMicroseconds(ch\_width\_5);

ch6.writeMicroseconds(ch\_width\_6);

ch7.writeMicroseconds(ch\_width\_7);

ch8.writeMicroseconds(ch\_width\_8);

}



***Required materials for regulators:***

1uF electrolytic capacitors : <https://s.click.aliexpress.com/e/_DDVm071>  
10uF electrolytic capacitors: <https://s.click.aliexpress.com/e/_Ddfia2R>  
100nF ceramic capacitor: <https://s.click.aliexpress.com/e/_Dd496aB>  
7805 5V Regulator IC: <https://s.click.aliexpress.com/e/_DDvYt5v>  
7809 9V Regulator IC: <https://s.click.aliexpress.com/e/_DlXmEG7>

80W soldering iron: <https://s.click.aliexpress.com/e/_DD75tWX>  
Digital Multimetre AC DC A830L : <https://s.click.aliexpress.com/e/_DmjzgOJ>  
Profesyonel Digital Multimetre Tester: <https://s.click.aliexpress.com/e/_DmtkrmJ>

***Necessary materials:***  
2 x  PS4 Analogue Joystick : <https://s.click.aliexpress.com/e/_DBSn2AB>  
2 x Toggle switch: <https://s.click.aliexpress.com/e/_DCd5Pzh>  
2 x 100mW NRF24L01 PA LN (E01-ML01DP5): <https://s.click.aliexpress.com/e/_DmEoWQf>  
2 x Arduino Nano: <https://s.click.aliexpress.com/e/_DlhwLS3>  
2 x NRF24 wireless 3.3v adaptor: <https://s.click.aliexpress.com/e/_DlmATh1>  
2 x 100uF electrolytic capacitors : <https://s.click.aliexpress.com/e/_DBJpcn1>  
4 x Female Header Pin (15 pins): <https://s.click.aliexpress.com/e/_DDqmgbh>  
2 x 10K potantiometer: <https://s.click.aliexpress.com/e/_DCV1V11>  
Universal PCB board: <https://s.click.aliexpress.com/e/_DF8xNJ9>  
AA Pil Kutusu (6xAA Cells) : <https://s.click.aliexpress.com/e/_DFMnduf>  
veya 18650 pil kutusu : <https://s.click.aliexpress.com/e/_DERuJnl>  
5mm LED: <https://s.click.aliexpress.com/e/_DB6EXhh>  
330R Direnç:  <https://s.click.aliexpress.com/e/_DFkljF9>  
Rocker buton (10x15mm): <https://s.click.aliexpress.com/e/_DDvMlXd>  
Tact buton (6x6x7): <https://s.click.aliexpress.com/e/_Dm2yUBt>  
Screws: <https://s.click.aliexpress.com/e/_DEvN4J9>