

Program Robot Manual

```
#include <Wire.h>
```

```
#include <Servo.h>
```

```
Servo servoA;
```

```
const int Channel1 =26;
```

```
const int Channel2 =28;
```

```
const int Channel4 =30;
```

```
const int Channel5 =24;
```

```
int rcmin=980;
```

```
int rcmid=1480;
```

```
int rcmax=1999;
```

```
int rcmidUpper=rcmid+70; //death band
```

```
int rcmidLower=rcmid-70; //death band
```

```
int vfwd=0;
```

```
int vyaw=0;
```

```
int vside=0;
```

```
int vleft=0;
```

```
int vright=0;
```

```
int vmotor=0;
```

```
int vmotormax=255;
```

```
int vmotormin=0;
```

```
int flag=0;
```

```
int stat=0;
```

```
int val;
```

```
int ch1; // Throttle
int ch2; // Aileron
int ch4; // Elevator
int ch5;
unsigned int THR,SWA,SWB,SWC,SWD,VRA,VRB;
////////////////////////////////////
int x=0;
int row=0;
////////////////////////////////////

void setup()
{
{
  Serial.begin(9600);
  Serial.println("CLEARDATA");
  Serial.println("LABEL,TIME,moveValue,turnValue,yawValue");
}
}

pinMode(49,INPUT);
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
pinMode(6,OUTPUT);
pinMode(8,OUTPUT);
pinMode(9,OUTPUT);
pinMode(12,OUTPUT);
pinMode(13,OUTPUT);
```

```

pinMode(47,OUTPUT);
pinMode(45,OUTPUT); //relay
pinMode(Channel1, INPUT); //22
pinMode(Channel2, INPUT); //24
pinMode(Channel4, INPUT); //26
pinMode(Channel5, INPUT); //20

servoA.attach(36);
}

void RCconversionMECA()
{
Serial.print("DATA,TIME,");
Serial.println(""+String(vfwd)+ ", "+String(vside)+ ", "+String(vyaw));

ch1 = pulseIn(Channel1, HIGH, 25000); // Read the pulse width of each channel
ch2 = pulseIn(Channel2, HIGH, 25000);
ch4 = pulseIn(Channel4, HIGH, 25000);
ch5 = pulseIn(Channel5, HIGH, 25000);

if(ch2>rcmidUpper)
{vfwd=map(ch2,rcmidUpper,rcmax,vmotormin,vmotormax);
digitalWrite(47,LOW);robotmove(vfwd,vfwd,vfwd,vfwd);}

else if(ch2<rcmidLower) {vfwd=map(ch2,rcmidLower,rcmin,-vmotormin,-
vmotormax);digitalWrite(47,LOW);robotmove(vfwd,vfwd,vfwd,vfwd);}

else {vfwd=0;}

if(ch1>rcmidUpper)
{vside=map(ch1,rcmidUpper,rcmax,vmotormin,vmotormax);
digitalWrite(47,LOW);robotmove(vside,-vside,vside,-vside);}

```

```

    else if(ch1<rcmidLower) {vside=map(ch1,rcmidLower,rcmin,-vmotormin,-
vmotormax); digitalWrite(47,LOW);robotmove(vside,-vside,vside,-vside);}

    else          {vside=0;}

    if(ch4>rcmidUpper)
{vyaw=map(ch4,rcmidUpper,rcmax,vmotormin,vmotormax-100);
digitalWrite(47,LOW);robotmove(vyaw,-vyaw,-vyaw,vyaw);}

    else if(ch4<rcmidLower) {vyaw=map(ch4,rcmidLower,rcmin,-vmotormin,-
vmotormax-100); digitalWrite(47,LOW);robotmove(vyaw,-vyaw,-vyaw,vyaw);}

    else          {vyaw=0;}

    if((v fwd==0)&&(vside==0)&&(vyaw==0))
{robotmove(0,0,0,0);digitalWrite(47,HIGH);}

}

void robotmove(int mot3,int mot2,int mot1,int mot4)//robotmove(a,b,c,d)
m1,m2,m3,m4 depan kiri,depan kanan,blakang kanan,balakang kiri
{
    if(mot1>0) //FORWARD
    {
        digitalWrite(5,LOW);
        digitalWrite(6,HIGH);
        analogWrite(7,mot1);
    }
    else if(mot1<0)//BACKWARD
    {
        digitalWrite(5,HIGH);
        digitalWrite(6,LOW);
        analogWrite(7,abs(mot1));
    }
}

```

```
else //BRAKE
{
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  analogWrite(7,0); //kiri
}

if(mot2>0) //FORWARD
{
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  analogWrite(4,mot2); //kanan
}
else if(mot2<0)//BACKWARD
{
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);
  analogWrite(4,abs(mot2)); //kanan
}
else //BRAKE
{
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  analogWrite(4,0); //kanan
}

if(mot3>0) //FORWARD
{
```

```
digitalWrite(8,HIGH);
digitalWrite(9,LOW);
analogWrite(10,mot3);
}
else if(mot3<0)//BACKWARD
{
digitalWrite(8,LOW);
digitalWrite(9,HIGH);
analogWrite(10,abs(mot3));
}
else //BRAKE
{
digitalWrite(8,LOW);
digitalWrite(9,LOW);
analogWrite(10,0);
}

if(mot4>0) //FORWARD
{
digitalWrite(12,LOW);
digitalWrite(13,HIGH);
analogWrite(11,mot4);
}
else if(mot4<0)//BACKWARD
{
digitalWrite(12,HIGH);
digitalWrite(13,LOW);
analogWrite(11,abs(mot4));
```

```
    }
    else //BRAKE
    {
        digitalWrite(12,LOW);
        digitalWrite(13,LOW);
        analogWrite(11,0);
    }
}
void loop()
{
    // {
    // Serial.print("DATA,TIME,");
    // Serial.print(String(vfwd));
    // Serial.print(String(vside));
    // Serial.println(String(vyaw));
    // delay(100);
    // }

    val = digitalRead(ch5);
    val = map(ch5, 980, 1999, 0, 180);
    servoA.write(val);

    RCconversionMECA();
//}
row++;
x++;
if (row>1000)
{
    row=0;
```

```
Serial.println("ROW,SET,2");  
}  
//delay(10);  
  
}
```

Program Test *Receiver*

```
#define LOWER_STOP_RANGE_MOVE -20  
#define UPPER_STOP_RANGE_MOVE 20  
#define LOWER_STOP_RANGE_TURN -20  
#define UPPER_STOP_RANGE_TURN 20
```

```
#include <Servo.h>
```

```
Servo servoA;  
const int Channel1 = 6;  
const int Channel2 = 7;  
const int Channel4 = 8;  
const int Channel5 = 9;
```



```
boolean stop_state = true;

// MODE2
int ch1; // Throttle
int ch2; // Aileron
int ch4; // Elevator
int ch5; // Rudder

int yawValue;
int moveValue;
int turnValue;
int val;

void setup(){
  servoA.attach(36);
  pinMode(Channel1, INPUT); //6
  pinMode(Channel2, INPUT); //7
  pinMode(Channel4, INPUT); //8
  pinMode(Channel5, INPUT); //9
  Serial.begin(9600);
} // void setup()

void loop() {
  // put your main code here, to run repeatedly:
  ch1 = pulseIn(Channel1, HIGH, 25000); // Read the pulse width of each channel
  ch2 = pulseIn(Channel2, HIGH, 25000);
  ch4 = pulseIn(Channel4, HIGH, 25000);
  ch5 = pulseIn(Channel5, HIGH, 25000);
```

```
moveValue = map(ch2, 980, 1999, -255, 255); //center over zero
```

```
moveValue = constrain(moveValue, -255, 255);
```

```
turnValue = map(ch1, 980, 1999, -255, 255);
```

```
turnValue = constrain(turnValue, -255, 255);
```

```
yawValue = map(ch4, 980, 1999, -255, 255);
```

```
yawValue = constrain(yawValue, -255, 255);
```

```
val = digitalRead(ch5);
```

```
val = map(ch5, 980, 1999, 0, 180);
```

```
servoA.write(val);
```

```
Serial.println("moveValue: "+String(moveValue)+ ", turnValue:  
"+String(turnValue)+ ", yawValue: "+String(yawValue)+ ", servoValue:  
"+String(val));
```

```
if (moveValue>LOWER_STOP_RANGE_MOVE &&  
moveValue<UPPER_STOP_RANGE_MOVE &&  
turnValue>LOWER_STOP_RANGE_TURN &&  
turnValue<UPPER_STOP_RANGE_TURN){
```

```
    if(stop_state == false){
```

```
        stop_state = true;
```

```
        Serial.println("Stop");
```

```
    }
```

```
}
```

```
//GO FORWARD & BACKWARD
```

```
    else if(turnValue>LOWER_STOP_RANGE_TURN &&
turnValue<UPPER_STOP_RANGE_TURN){
        if(moveValue>UPPER_STOP_RANGE_MOVE){
            stop_state = false;
            Serial.println("Go Forward "+String(moveValue));
        }
        else if(moveValue<LOWER_STOP_RANGE_MOVE){
            stop_state = false;
            Serial.println("Go Backward "+String(moveValue));
        }
    }
    //TURN RIGHT & LEFT
    else if(moveValue>LOWER_STOP_RANGE_MOVE &&
moveValue<UPPER_STOP_RANGE_MOVE){
        if(turnValue>UPPER_STOP_RANGE_TURN){
            stop_state = false;
            Serial.println("Turn Right "+String(turnValue));
        }
        else if(turnValue<LOWER_STOP_RANGE_TURN){
            stop_state = false;
            Serial.println("Turn Left "+String(turnValue));
        }
    }
    delay(200);
}
```