LAMPIRAN

Solar_Cell_PID.ino

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

#define I2C_ADDR 0x27
#define Rs_pin 0
#define Rw_pin 1
#define En_pin 2
#define BACKLIGHT_PIN 3
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7

LiquidCrystal_I2C
lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pin);

float v;

int rain;
int l1,l2,l3,l4,l5,sensor_data,sl,sr,rl,rr;

int pidroll,pidswing;
int kp,ki,kd;

int TS;
int L_PV;
int R_PV;
int error_L;
int error_R;
int error_last_L;
int error_last_R;
int setpoint;
int LPWM, RPWM, I_L, I_R, D_L, D_R, PID_L, PID_R;

void setup()
{
  pinMode(A0,INPUT);
  pinMode(A1,INPUT);
  pinMode(A2,INPUT);
```

```
  pinMode(A3,INPUT);
  pinMode(A4,INPUT);

  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(10,OUTPUT);

  pinMode(8, INPUT_PULLUP);
  pinMode(9, INPUT_PULLUP);

  Serial.begin(9600);
  lcd.clear();
  lcd.begin (16,2);
  lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
  lcd.setBacklight(HIGH);
  lcd.home ();
  lcd.setCursor (0,0);
  lcd.print("SOLAR CELL      ");
  lcd.setCursor (0,1);
  lcd.print("   LIGHT TRACKER");
  delay(1000);
  lcd.clear();
}

void sensor()
{
  if  (digitalRead(A0)==LOW)  {l1=1;} else {l1=0;}
  if  (digitalRead(A1)==LOW)  {l2=1;} else {l2=0;}
  if  (digitalRead(A2)==LOW)  {l3=1;} else {l3=0;}
  if  (digitalRead(A3)==LOW)  {l4=1;} else {l4=0;}

    sensor_data=(l1*1)+(l2*2)+(l3*4)+(l4*8);
    sensor_data&=0b1111;

  lcd.setCursor (0,1);
  lcd.print(l1);
```

```
 lcd.setCursor (1,1);
 lcd.print(l2);
 lcd.setCursor (2,1);
 lcd.print(l3);
 lcd.setCursor (3,1);
 lcd.print(l4);
 delay(100);
 lcd.clear();
}

void lightfollowing()
{
   sensor();
   switch(sensor_data)
   {
      case 0b1010:    sl=1; sr=0; rl=0; rr=0; L_PV=1 ;  swingkiri();  break;
      case 0b0101:    sl=0; sr=1; rl=0; rr=0; L_PV=1;   swingkanan(); break;
      case 0b0011:    rl=0; rr=1; sl=0; sr=0; R_PV=1 ;  rollkanan();  break;
      case 0b1100:    rl=1; rr=0; sl=0; sr=0; R_PV=1;   rollkiri();   break;
      case 0b1111:    rl=1; rr=1; sl=1; sr=1;          berhenti();   break;
      case 0b0000:
                  if     (rl==0 && rr==1)  {rollkanan();  break; }
                  else if (rl==1 && rr==0)  {rollkiri();   break; }
                  else if (sl==0 && sr==1)  {swingkiri();  break; }
                  else if (sl==1 && sr==0)  {swingkanan(); break; }
                  else                  {berhenti();}
   }
}


void loop()
{
 if    (digitalRead(A4)==LOW)
       {
        rain=1;
        lcd.setCursor (4,1);
        lcd.print(rain);
        berhenti();
       }
 else  {
```

```
        rain=0;
        lcd.setCursor (4,1);
        lcd.print(rain);
        lightfollowing();
      }

v=(float)analogRead(A5)/40.6; //40.6
lcd.setCursor (7,0);
lcd.print("AKI=");
lcd.setCursor (11,0);
lcd.print(v,1);
lcd.setCursor (15,0);
lcd.print("V");

if    (v<12)
      {
        digitalWrite(10,HIGH);
        lcd.setCursor (7,1);
        lcd.print("TERPUTUS");
      }
else  {
        lcd.setCursor (7,1);
        lcd.print("TERHUBUNG");
        digitalWrite(10,LOW);
      }

TS=70;
kp=5;
ki=1;
kd=1;

error_L=setpoint-L_PV;
I_L= (I_L+error_L)*0.01;
D_L=(error_L-error_last_L)*100;
PID_L=kp*error_L+ki*I_L+kd*D_L;
error_last_L=error_L;

error_R=setpoint-R_PV;
I_R= (I_R+error_R)*0.01;
D_R=(error_R-error_last_R)*100;
```

```
  PID_R=kp*error_R+ki*I_R+kd*D_R;
  error_last_R=error_R;

  LPWM=TS+PID_L;
  RPWM=TS+PID_R;

  pidroll=LPWM;
  pidswing=RPWM;
}
```

motion.ino

```
voidrollkanan()
{
if  (digitalRead(8)==HIGH)
  {
digitalWrite(2, LOW);
digitalWrite(3, HIGH);
analogWrite (4, 10+pidroll);
lcd.setCursor (0,0);
lcd.print("^");
lcd.setCursor (2,0);
lcd.print(pidroll);
  }
else
  {
digitalWrite(2, LOW);
digitalWrite(3, LOW);
analogWrite (4, 0);
//  lcd.setCursor (6,1);
//  lcd.print("LR>");
  }
}

voidrollkiri()
{
if  (digitalRead(9)==HIGH)
  {
digitalWrite(2, HIGH);
digitalWrite(3, LOW);
analogWrite (4, 10+pidroll);
lcd.setCursor (0,0);
```

```
lcd.print("v");
lcd.setCursor (2,0);
lcd.print(pidroll);
  }

else
  {
digitalWrite(2, LOW);
digitalWrite(3, LOW);
analogWrite (4, 0);
//  lcd.setCursor(6,1);
//  lcd.print("LR<");
  }
}

voidswingkanan()
{
digitalWrite(5, HIGH);
digitalWrite(6, LOW);
analogWrite (7, pidswing);
lcd.setCursor (0,0);
lcd.print(">");
lcd.setCursor (2,0);
lcd.print(pidswing);
}

voidswingkiri()
{
digitalWrite(5, LOW);
digitalWrite(6, HIGH);
analogWrite (7, pidswing);
lcd.setCursor (0,0);
lcd.print("<");
lcd.setCursor (2,0);
lcd.print(pidswing);
}

voidberhenti()
{
digitalWrite(2, LOW);
```

```
digitalWrite(3, LOW);
analogWrite (4, pidswing);

digitalWrite(5, LOW);
digitalWrite(6, LOW);
analogWrite (7, pidswing);

lcd.setCursor (0,0);
lcd.print("STOP");
}
```