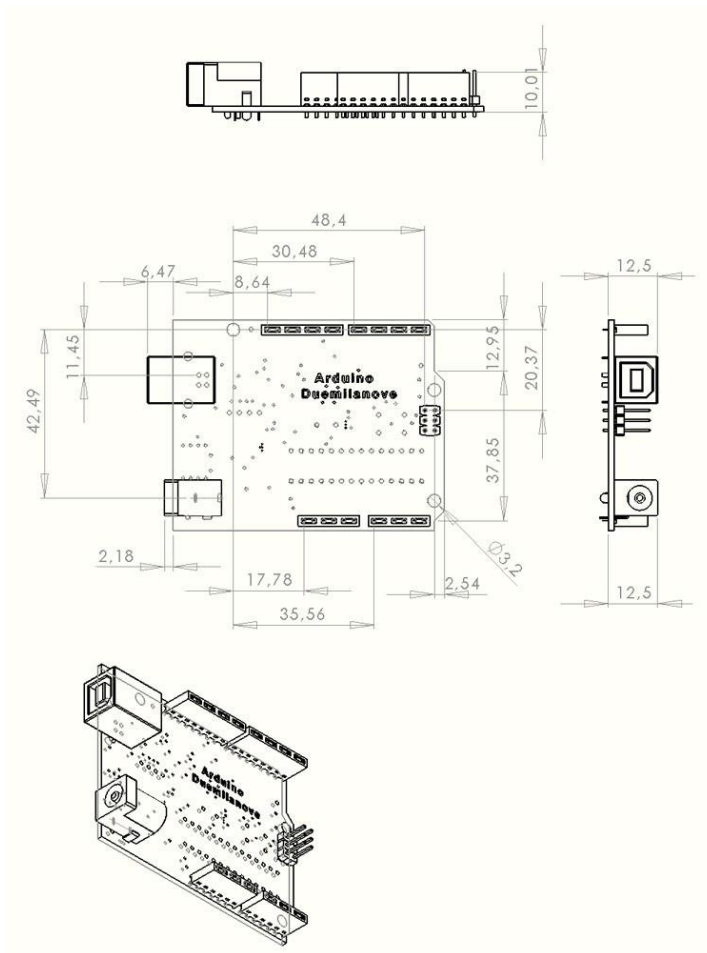
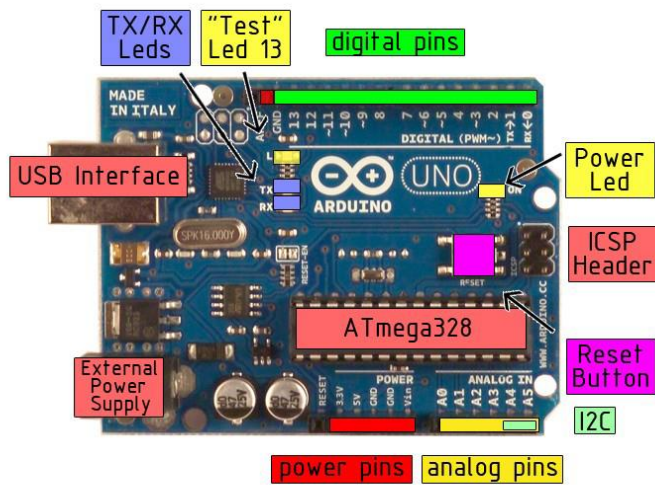
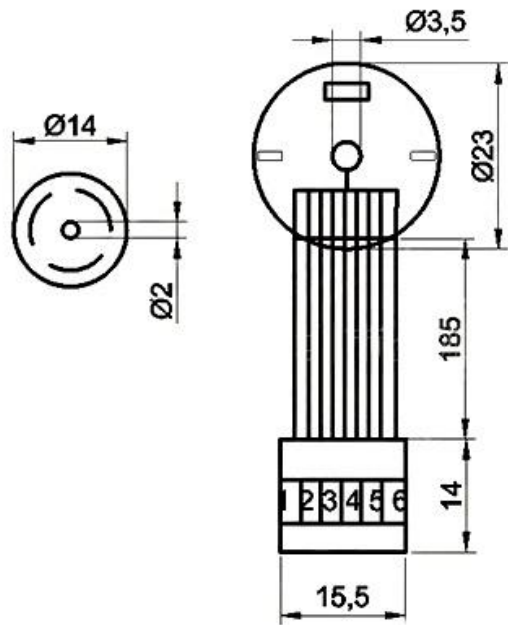


LAMPIRAN



Pin Category	Pin Name	Details
Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
Input/Output Pins	Digital Pins 0 - 13	Can be used as input or output pins.
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	AREF	To provide reference voltage for input voltage.



Parameter:

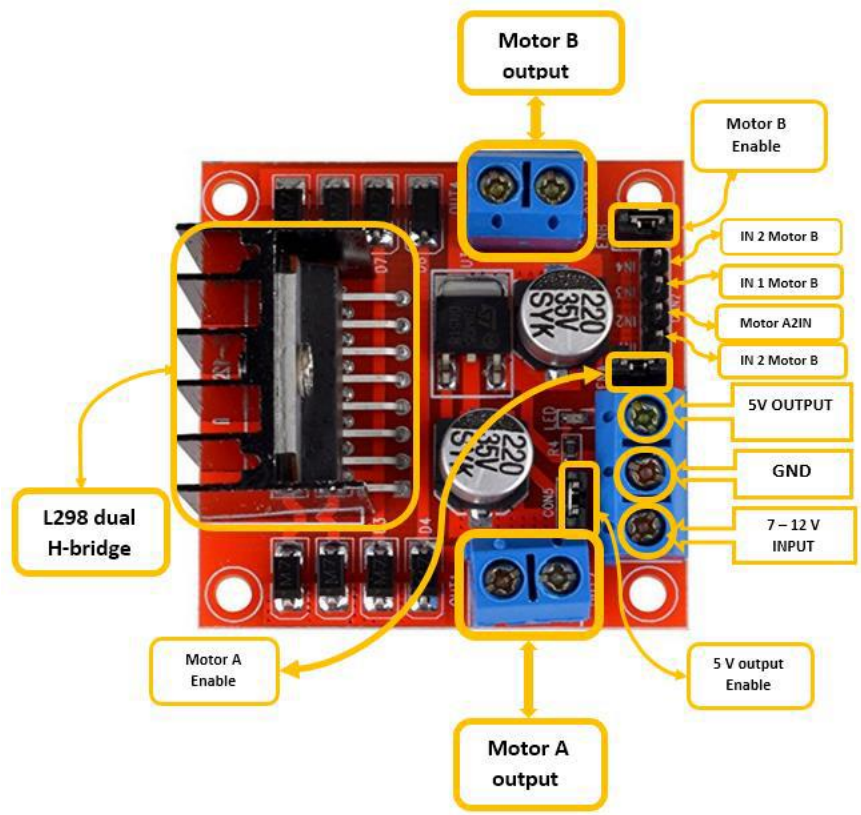
Operating voltage: between 6 and 24 V DC
 Nominal voltage: 12 V DC

Connection of the encoder:

- Red: Motor power supply +
- Black: Coding power supply -
- Yellow: Signal feedback
- Green: Signal feedback
- Blue: Coding power supply +
- White: Motor power supply -



- M1 Motor-
- GND Encoder-
- Encoder A Phase
- Encoder B Phase
- V3V Encoder+
- M1 Motor+



Kode program keseluruhan

```
#include <LiquidCrystal.h> //masukan library lcd
#include <PID_v1.h> //masukan library PID
#define encodPinA1 2 //encoder A di pin 2
#define encodPinB1 8 //encoder B di pin 8
#define M1 9 //pin 9 keluaran pwm ke modul driver L298N
#define M2 10 //pin 10 keluaran pwm ke modul driver L298N
#define PERIOD 1000 //memberi variabel period bernilai 1 detik

int button1 = 3; //pushbutton 1 di pin 3
int button2 = 13; //pushbutton 2 di pin 13

int stateSetpoint = HIGH;
int stateButton;
int previous = LOW;
long times = 0;
long debounce = 200;

int buttonPushCounterA = 0, buttonPushCounterB = 0 ; // counter for the number of
button presses

int buttonState1 = 0, buttonState2 = 0; // current state of the button
int lastButtonState1 = 0, lastButtonState2 = 0;
int setpoint_a = 1;
int setDerajat, inputDerajat, pot;
double input, output, setpoint, error = 0; //konfigurasi input, setpoint dan output dalam
tipe data double
volatile long encoderPos, encoderlast;
unsigned long int waktu;
const long lcdPing = 100;
unsigned long now = 0;
unsigned long lastMessage = 0;
```

```

const int rs = 11, en = 12, d4 = 4, d5 = 5, d6 = 6, d7 = 7; //inisialisasi pin Lcd

//memasukan parameter PID

double kp = 4.41; //masukan nilai KP tipe data berupa pecahan ganda
double Ti = 1.24; //masukan nilai Ti tipe data berupa pecahan ganda
double Td = 0.31; //masukan nilai Td tipe data berupa pecahan ganda
double ki = kp/Ti; //perhitungan nilai KI tipe data berupa pecahan ganda
double kd = kp*Td; //perhitungan nilai KD tipe data berupa pecahan ganda

PID myPID(&input, &output, &setpoint, kp, ki, kd, DIRECT); //sintaks PID kontroler
dengan variabel myPID

LiquidCrystal lcd(rs,en,d4,d5,d6,d7); //sintaks penampil lcd dengan variabel lcd

void setup(){
  lcd.begin(16,2);
  pinMode(button1,INPUT);
  pinMode(button2,INPUT);
  digitalWrite(button1,HIGH);
  digitalWrite(button2,HIGH);
  pinMode(encodPinA1, INPUT_PULLUP);
  pinMode(encodPinB1, INPUT_PULLUP);
  attachInterrupt(0, encoder, FALLING); //perbarui posisi enkoder
  myPID.SetMode(AUTOMATIC); //menseset PID secara otomatis
  myPID.SetSampleTime(1); //sampling time 1ms
  myPID.SetOutputLimits(-255, 255); //menseset min dan maks pwm
  Serial.begin(115200);
  lastMessage = millis();
  penalaan();
  encoderPos = encoderlast; //encoder = nilai encoder sebelumnya }

```

```

//program pembacaan encoder
void encoder() {          // pulse and direction, direct port reading to save cycles
    if (PINB & 0b00000001) encoderPos++; // if(digitalRead(encodPinB1)==HIGH) count
    ++;
    else          encoderPos--; // if(digitalRead(encodPinB1)==LOW) count --;
}

//program keluaran ke motor berupa PWM
void pwmout(int out) {    // to H-Bridge board
    if (out > 0) { //motor berputar CW
        analogWrite(M1, abs(out)); //nilai pwm di M1 bernilai mutlak |M1|
        analogWrite(M2, 0);
    }
    else { //motor berputar CCW
        analogWrite(M1, 0);
        analogWrite(M2, abs(out)); //nilai pwm di M1 bernilai mutlak |M2|
    }
}

//program untuk merekam data ke excell
void daq(){
    waktu = millis();
    Serial.print("DATA,TIME,");
    Serial.print(waktu);
    Serial.print(",");
    Serial.println(input);
}

//program lcd untuk menampilkan data
void tampilan(){

```



```

lcd.setCursor(0,0);
lcd.print(setpoint);
lcd.setCursor(0,1);
lcd.print(input);
lcd.setCursor(7,0);
lcd.print(setDerajat);
lcd.setCursor(12,0);
lcd.print("Deg");
lcd.setCursor(7,1);
lcd.print(inputDerajat);
lcd.setCursor(12,1);
lcd.print("Deg");
}
//program lcd untuk menampilkan parameter PID
void penalaan(){
  lcd.setCursor(4,0);
  lcd.print("PARAMETER");
  lcd.setCursor(6,1);
  lcd.print("PID");
  delay(2000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(kp);
  lcd.setCursor(0,1);
  lcd.print(ki);
  delay(1500);
  lcd.clear(); }

```

```

void loop() {
  buttonState1 = digitalRead(button1);
  if (buttonState1 != lastButtonState1) {
    if (buttonState1 == HIGH) { //program untuk menghitung jumlah PB A ditekan
      buttonPushCounterA++;
    }
  }
  lastButtonState1 = buttonState1;
  buttonState2 = digitalRead(button2);
  if (buttonState2 != lastButtonState2) {
    if (buttonState2 == HIGH) {
      buttonPushCounterB++;
    }
  }
  lastButtonState2 = buttonState2;
  if (buttonPushCounterA == 1){
    pot = analogRead(A0); //modify to fit motor and encoder characteristics, potmeter
    connected to A0

    //setpoint berupa sinyal analog
    setpoint = map(pot, 0, 1023, 0, 217); //mensetting putaran potensio dari 0 - 90 derajat
  }
  else if (buttonPushCounterA == 2 ){
    setpoint = 870/4; //menset PB A ditekan 2 kali, nilai setpoint 217
  }
  else if (buttonPushCounterA == 3){
    setpoint = 870/2; //menset PB A ditekan 3 kali, nilai setpoint 435
  }
}

```

```

else if (buttonPushCounterA == 5){
    buttonPushCounterA = 0; //menset PB A ditekan 5 kali counter PB A kembali ke awal
}
setDerajat=map(setpoint ,0 ,870 ,0 ,360 );
input = encoderPos; //feedback berupa nilai encoder
inputDerajat = map (input ,0 , 870, 0,360 );
myPID.Compute(); //PID kontroler
pwmout(output); // drive L298N H-Bridge module
now = millis();
if(now - lastMessage > lcdPing){ //program untuk dapat menampilkan data ke lcd
    daq(); //memanggil fungsi daq
    tampilan(); //memanggil fungsi tampilan
    lastMessage = now;
}
encoderlast = input; //nilai encoder last sama dengan nilai feedback
}

```