

ON7WP Arduino DC rotator controller – cheap but performing.

I needed a rotator controller as I have some orphan Create and Yaesu SDX800 rotators that both work on DC.

If you google “arduino rotator” then you get tons of articles all based on the K3NG design. Seriously ??

Although it is a nice realization it is typical for amateur radio applications. Vintage like hell based on old school technologies. RELAYS, in 2019 ? Really ???

So in an afternoon I put something together minimal effort based on a cheap **Chinese Arduino**, a **Chinese LCD keypad shield** and a **Chinese L298N motor driver** module. All for less than 10 Euro !

Straight forward design: two lines on the display, the first showing the actual rotator position, the second the target position. You set the target using a 10 turn potmeter. No other parts are needed, just some wiring.

The rotator will automatically adjust speed when reaching target in three steps.

Doubling the code, the same hardware can control an azimuth/elevation rotator, you only have to shorten the display text to get all on it and add a second potmeter.

Wiring: connect 11-12-13 to the L298N controller, connect both potmeters, from rotor feedback and from set potmeter to 5V DC and the central point to resp. ADC1 and ADC2. 12V to the driver board, rotor motor to the two motor-A pins and 5V DC out to Arduino together with GND. Upload the code and all set...

Feedback appreciated Pedro.wyns@gmail.com

Arduino Code:

```
/*****
```

Description: Rotator controller for DC Create by ON7WP

```
*****/
```

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);    // select the pins used on the LCD panel
```

```
// Motor Azimuth connections
```

```

int enA = 11;

int in1 = 12;

int in2 = 13;

// Motor Elevation connections

int enB = 3;

int in3 = 2;

int in4 = 1;


int rotorfeedback;           // variable to store the value coming from the analog pin
double act;                  // variable to store the temperature value coming from the conversion
formula
double error;                // variable to store the pointing error
int potmeter;                // variable to store the value coming from the analog pin
double set;                  // variable to store the temperature value coming from the conversion
formula

void setup(){
  lcd.begin(16, 2);           // start the library


// Set all the motor control pins to outputs
pinMode(enA, OUTPUT);
pinMode(enB, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);


// Turn off motors - Initial state
digitalWrite(in1, LOW);

```

```

digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
}

void loop(){

    rotorfeedback= analogRead(1);           // read the analog in value:
    act = (double) rotorfeedback * (398/1024.0);    // temperature conversion formula

    potmeter=analogRead(2);                 // read the analog in value:
    set = (double) potmeter * (398/1024.0);        // temperature conversion formula

    // print the results to the lcd
    lcd.setCursor(0, 0);                    // set the LCD cursor position
    lcd.print("ACT: ");
    lcd.print(act-19);
    lcd.print(" deg.");
    lcd.setCursor(0, 1);                    // set the LCD cursor position
    lcd.print("SET: ");
    error=set-act;
    lcd.print(set-19);
    lcd.print(" deg.");
    if(set < act)
    {
        TurnCCW();
    }
    else {
        TurnCW();
    }
}

```

```

    }
    //speedcontrol while reaching target
    if ((abs(set-act))<10)
    {
        analogWrite(enA, 200);
        analogWrite(enB, 200);
    }
    if ((abs(set-act))<3)
    {
        analogWrite(enA, 150);
        analogWrite(enB, 150);
    }
    if ((abs(set-act))<0.3)
    {
        digitalWrite(enA, LOW);
        digitalWrite(enB, LOW);
    }
    // Stop();
}

// This function lets you control the motor to turn clockwise
void TurnCW(){
    analogWrite(enA, 255);
    analogWrite(enB, 255);
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
}

void TurnCCW(){

```

```
    analogWrite(enA, 255);  
    analogWrite(enB, 255);  
    digitalWrite(in1, HIGH);  
    digitalWrite(in2, LOW);  
    digitalWrite(in3, HIGH);  
    digitalWrite(in4, LOW);  
}  
void Stop(){  
    // Turn off motors  
    digitalWrite(in1, LOW);  
    digitalWrite(in2, LOW);  
    digitalWrite(in3, LOW);  
    digitalWrite(in4, LOW);  
}
```