DC Motor Interfacing with Arduino UNO

Overview of DC Motor



DC Motor

DC motor converts electrical energy in the form of Direct Current into mechanical energy in the form of rotational motion of the motor shaft.

The DC motor speed can be controlled by applying varying DC voltage; whereas the direction of rotation of the motor can be changed by reversing the direction of current through it.

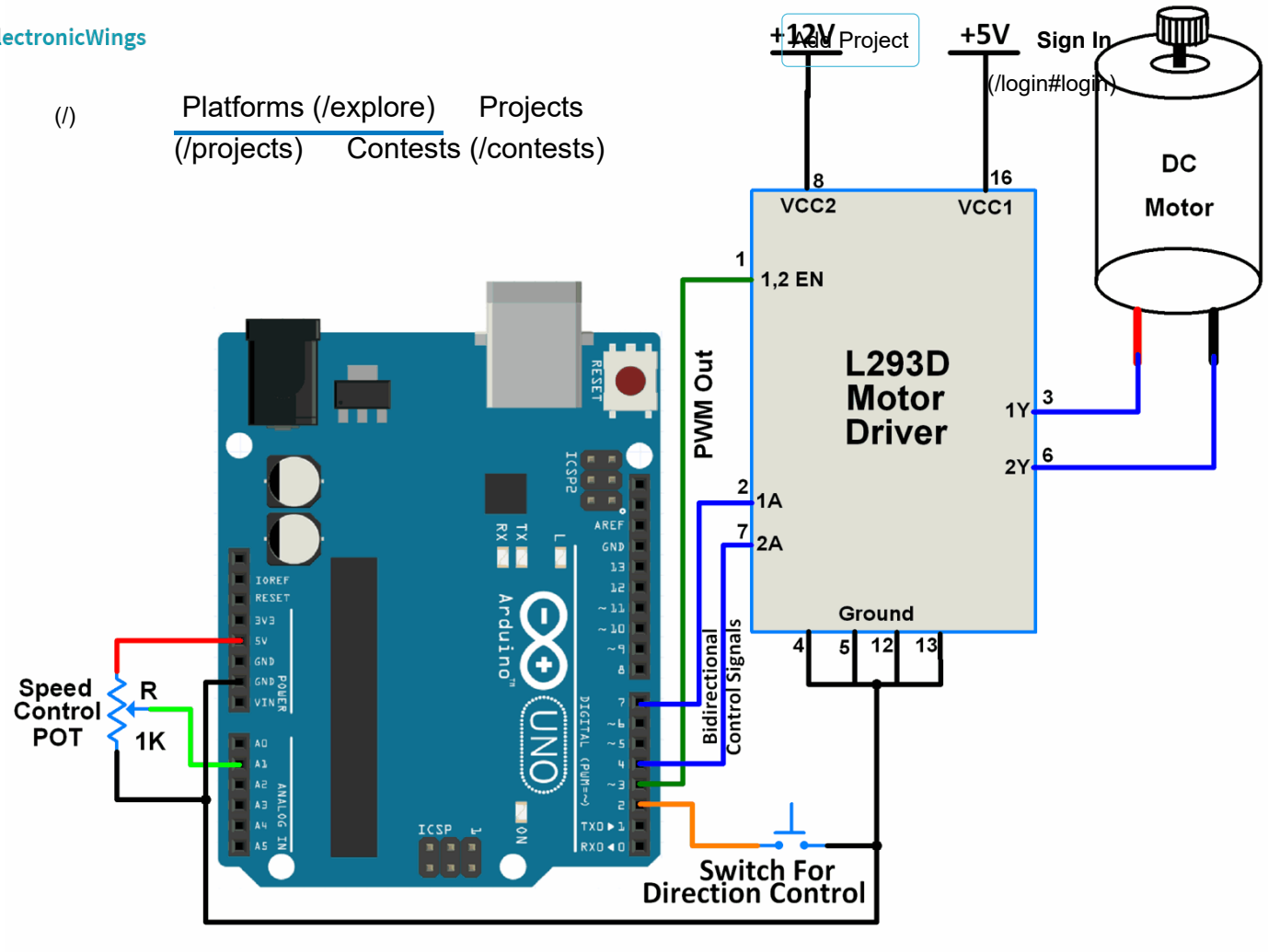
For applying varying voltage, we can make use of PWM technique.

For reversing the current, we can make use of H-Bridge circuit or motor driver ICs that employ the H-Bridge technique.

For more information about DC motors and how to use them, H-Bridge circuit configurations, and PWM technique, refer the topic DC Motors (http://www.electronicwings.com/sensors-modules/dc-motor) in the sensors and modules section.

Connection Diagram of DC Motor with Arduino

ElectronicWings
(/) Platforms (/explore) Projects (/projects) Contests (/contests) Add Project Sign In (/login#login)



Interfacing DC Motor with Arduino UNO

Example

Here, we are going to control the speed and rotational direction of DC motor using Arduino Uno.

Here, a potentiometer is used as a means for speed control and an input from a tactile switch is used to change the direction of the motor.

L293D motor driver IC is used for controlling the direction of the motor.

PWM wave generated on the Arduino UNO is used to provide a variable voltage to the motor through L293D. In Arduino, analogWrite function is used to generate PWM wave.

Direction and Speed Control of DC Motor Code for Arduino

ElectronicWings Add Project Sign In (/login#login)
(/) Platforms (/explore) Projects (/projects) Contests (/contests)

```
const int POT_input = A1;   /* assign ADC Channel */
bool d1 = HIGH;
bool d2 = LOW;

void setup() {
  pinMode(4, OUTPUT);  /* Motor control pin 1 */
  pinMode(7, OUTPUT);  /* Motor control pin 2 */
  pinMode(3, OUTPUT);  /* PWM pin for Speed Control */
  pinMode(2, INPUT_PULLUP);  /* Interrupt pin for direction control */
  attachInterrupt(digitalPinToInterrupt(2), motor, FALLING);    /* Interrupt on falling edge on
}

void loop() {
  int pwm_adc;
  pwm_adc = analogRead(POT_input); /* Input from Potentiometer for speed control */
  digitalWrite(4,d1);
  digitalWrite(7,d2);
  analogWrite(3, pwm_adc / 4);
}

void motor(){
  d1 = !d1;
  d2 = !d2;
  _delay_ms(200);
}
```

Video of DC Motor Speed Control Using Arduino Uno

Functions Used

1. `digitalPinToInterrupt(pin)`

- This function is used to declare the digital pin as an interrupt pin.
- Example: `digitalPinToInterrupt(2)` is used to declare digital pin 2 as an interrupt pin.
- On UNO board, only pins 2 and 3 can be configured as interrupt pins. Hence, argument to this function can only be pin 2 or pin 3.

2. `attachInterrupt(digitalPinToInterrupt(pin), ISR, mode)`

- This function is used to configure the mode of interrupt event and declare the ISR for that interrupt. The interrupt event and ISR is for the interrupt pin declared by the function digitalPinToInterrupt(pin).
- ISR in this function is the name of the ISR that will be used for this interrupt.
- mode defines when the interrupt will be triggered. There are four modes available to choose from :
  - `LOW` : trigger the interrupt whenever the pin is low.
  - `CHANGE` : trigger the interrupt whenever the pin changes value.
  - `RISING` : trigger when the pin goes from low to high.
  - `FALLING` : trigger when the pin goes from high to low.
- Example, `attachInterrupt(digitalPinToInterrupt(2), motor, FALLING)` configures digital pin 2 as an interrupt pin with ISR named motor and which generates interrupt for every falling edge event on pin 2.

3. `analogWrite(pin,value)`

- (//)This function is used for generating PWM on PWM digital pins(pins 3,5,6,9,10,11 for Arduino UNO).
- value can be any number between 0 to 255. 0 being 0% duty cycle and 255 being 100% duty cycle.

Powered By MOUSER ELECTRONICS (https://www.mouser.in?utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

Components Used

Arduino UNO Arduino UNO	**X 1**	(https://www.mouser.in/ProductDetail/Arduino/A000073?qs=8PMfw1Pw72VfrrCu0Mm0mA%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0) Datasheet (/components/arduino-uno/1/datasheet)
Arduino Nano Arduino Nano	**X 1**	(https://www.mouser.com/ProductDetail/Arduino/ABX00033?qs=PzGy0jfpSMueXyfBfl2XSA%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0) Datasheet (/components/arduino-nano/1/datasheet)

Components Used

L293D Driver
L293D Driver
X 1
(https://www.mouser.com/ProductDetail/STMicroelectronics/L293D?qs=gr8Zi5OG3MgMJ1ICDzLQbg%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)
Datasheet (/components/electronicwings/l293d-driver/1/datasheet)

Breadboard
Breadboard
X 1
(https://www.mouser.com/ProductDetail/BusBoard-Prototype-Systems/BB830?qs=VEImQw3kOauhPeTwYxNUaA%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)
Datasheet (/components/breadboard/1/datasheet)

Powered By (https://www.mouser.in?utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

Components Used

Mini Push Button Switch - 5-6mm Mini Push Button Switch - 5-6mm	**X 1**	(https://www.mouser.in/ProductDetail/Adafruit/4183?qs=PzGy0jfpSMvkd%252B6tk9f0Pw%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

Supporting Files

Source Code

- DC Motor Interfacing With Arduino UNO Download | 1034

Comments

Please login/signup to comment and reply.