```
// AEK-MOT-2DC40Y1 H-Bridge Connection PINs
#define Motor_driver_enA 5  // PWM, front1
#define Motor_driver_enB 6  // PWM, front2
#define Motor_driver_enC 9  // PWM, back1
#define Motor_driver_enD 10  // PWM, back2
#define Motor_driver_in1 2  // Dir Motor A
#define Motor_driver_in2 3  // Dir Motor A
#define Motor_driver_in3 4  // Dir Motor B
#define Motor_driver_in4 7  // Dir Motor B
#define Motor_driver_in5 8  // Dir Motor C
#define Motor_driver_in6 11  // Dir Motor C
#define Motor_driver_in7 12  // Dir Motor D
#define Motor_driver_in8 13  // Dir Motor D

// Interpret Serial Messages
bool is_right_wheel_cmd = false;
bool is_left_wheel_cmd = false;
bool is_right_wheel_forward = true;
bool is_left_wheel_forward = true;
char value[] = "00.00";
uint8_t value_idx = 0;
bool is_cmd_complete = false;

// Setpoint - Desired
double right_wheel_cmd_vel = 0.0;     // rad/s
double left_wheel_cmd_vel = 0.0;     // rad/s
```

```
void setup() {
 // Init L298N H-Bridge Connection PINs
 pinMode(Motor_driver_enA, OUTPUT);
 pinMode(Motor_driver_enB, OUTPUT);
 pinMode(Motor_driver_enC, OUTPUT);
 pinMode(Motor_driver_enD, OUTPUT);
 pinMode(Motor_driver_in1, OUTPUT);
 pinMode(Motor_driver_in2, OUTPUT);
 pinMode(Motor_driver_in3, OUTPUT);
 pinMode(Motor_driver_in4, OUTPUT);
 pinMode(Motor_driver_in5, OUTPUT);
 pinMode(Motor_driver_in6, OUTPUT);
 pinMode(Motor_driver_in7, OUTPUT);
 pinMode(Motor_driver_in8, OUTPUT);

 // Set Motor Rotation Direction
 digitalWrite(Motor_driver_in1, HIGH);
 digitalWrite(Motor_driver_in2, LOW);
 digitalWrite(Motor_driver_in3, HIGH);
 digitalWrite(Motor_driver_in4, LOW);
 digitalWrite(Motor_driver_in5, HIGH);
 digitalWrite(Motor_driver_in6, LOW);
 digitalWrite(Motor_driver_in7, HIGH);
 digitalWrite(Motor_driver_in8, LOW);

 Serial.begin(115200);
}
```

```
void loop() {
 // Read and Interpret Wheel Velocity Commands
 if (Serial.available())
 {
  char chr = Serial.read();
  // Right Wheel Motor
  if(chr == 'r')
  {
   is_right_wheel_cmd = true;
   is_left_wheel_cmd = false;
   value_idx = 0;
   is_cmd_complete = false;
  }
  // Left Wheel Mo tor
  else if(chr == 'l')
  {
   is_right_wheel_cmd = false;
   is_left_wheel_cmd = true;
   value_idx = 0;
  }
  // Positive direction
  else if(chr == 'p')
  {
   if(is_right_wheel_cmd && !is_right_wheel_forward)
   {
    // change the direction of the rotation
    digitalWrite(Motor_driver_in1, HIGH - digitalRead(Motor_driver_in1));
    digitalWrite(Motor_driver_in2, HIGH - digitalRead(Motor_driver_in2));
    digitalWrite(Motor_driver_in5, HIGH - digitalRead(Motor_driver_in5));
    digitalWrite(Motor_driver_in6, HIGH - digitalRead(Motor_driver_in6));
    is_right_wheel_forward = true;
  }
  else if(is_left_wheel_cmd && !is_left_wheel_forward)
  {
    // change the direction of the rotation
    digitalWrite(Motor_driver_in3, HIGH - digitalRead(Motor_driver_in3));
    digitalWrite(Motor_driver_in4, HIGH - digitalRead(Motor_driver_in4));
    digitalWrite(Motor_driver_in7, HIGH - digitalRead(Motor_driver_in7));
    digitalWrite(Motor_driver_in8, HIGH - digitalRead(Motor_driver_in8));
    is_left_wheel_forward = true;
  }
}
// Negative direction
else if(chr == 'n')
{
  if(is_right_wheel_cmd && is_right_wheel_forward)
  {
    // change the direction of the rotation
    digitalWrite(Motor_driver_in1, HIGH - digitalRead(Motor_driver_in1));
    digitalWrite(Motor_driver_in2, HIGH - digitalRead(Motor_driver_in2));
    digitalWrite(Motor_driver_in5, HIGH - digitalRead(Motor_driver_in5));
    digitalWrite(Motor_driver_in6, HIGH - digitalRead(Motor_driver_in6));
    is_right_wheel_forward = false;
  }
  else if(is_left_wheel_cmd && is_left_wheel_forward)
  {
    // change the direction of the rotation
    digitalWrite(Motor_driver_in3, HIGH - digitalRead(Motor_driver_in3));
    digitalWrite(Motor_driver_in4, HIGH - digitalRead(Motor_driver_in4));
    digitalWrite(Motor_driver_in7, HIGH - digitalRead(Motor_driver_in7));
    digitalWrite(Motor_driver_in8, HIGH - digitalRead(Motor_driver_in8));
    is_left_wheel_forward = false;
  }
}
// Separator
else if(chr == ',')
{
  if(is_right_wheel_cmd)
  {
    right_wheel_cmd_vel = atof(value);
    analogWrite(Motor_driver_enA, right_wheel_cmd);
    analogWrite(Motor_driver_enC, right_wheel_cmd);
  }
  else if(is_left_wheel_cmd)
  {
    left_wheel_cmd_vel = atof(value);
    analogWrite(Motor_driver_enB, right_wheel_cmd);
    analogWrite(Motor_driver_enD, right_wheel_cmd);
    is_cmd_complete = true;
  }
  // Reset for next command
  value_idx = 0;
  value[0] = '0';
  value[1] = '0';
```

```
      value[2] = '.';
      value[3] = '0';
      value[4] = '0';
      value[5] = '\0';
    }
    // Command Value
    /*else
    {
      if(value_idx < 5)
      {
        value[value_idx] = chr;
        value_idx++;
      }
    }*/
  }
}
```