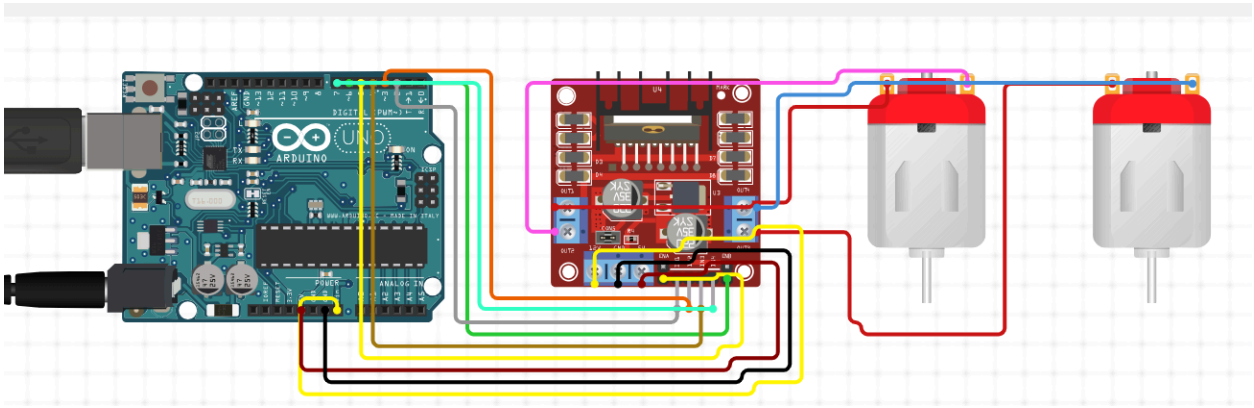
Motor controller Tutorial

Using a motor controller, Arduino devices can control DC motors, enabling variable speed, and direction. A motor controller will allow movement and steering through the use of two motors.

Hardware

- Arduino
- Motor controller
- DC 12v motors (2)
- 12v battery
- 9v battery
- wiring

Wiring



Follow this diagram to wire the motor controller to the arduino, and both motors to the motor controller.

Motor controller code

```
int AENA = 6;              // Speed input for Motor A
  int Apin1 = 4;             // Forward for Motor A
```

```
int Apin2 = 5;           // Reverse for Motor A
int BENA = 3;            // Speed input for Motor B
int Bpin1 = 2;           // Forward for Motor B
int Bpin2 = 7;           // Reverse for Motor B
long time = 0;           // Variable to store time
String numberint;        // Defined integer to store number in second transmission

using namespace std;

#include <stdlib.h>      // Standard library
#include <ctype.h>       // Character handling functions
#include <RH_ASK.h>      // RadioHead library for ASK RF Transmitter
#ifdef RH_HAVE_HARDWARE_SPI
#include <SPI.h>         // SPI library (not actually used but needed to compile)
#endif

RH_ASK driver;           // Initialize RH_ASK object for communication

void setup() {
Serial.begin(9600);                    // Begin serial communication for debugging
if (!driver.init())
Serial.println("init failed");        // Print message if initialization fails

pinMode(AENA, OUTPUT);                 //
pinMode(Apin1, OUTPUT);                //
pinMode(Apin2, OUTPUT);                //Set pins as outputs
pinMode(BENA, OUTPUT);                 //
pinMode(Bpin1, OUTPUT);                //
pinMode(Bpin2, OUTPUT);                //
}

void loop() {
uint8_t buf[RH_ASK_MAX_MESSAGE_LEN];   // Buffer to hold received message
uint8_t buflen = sizeof(buf);          // Length of buffer
```

```
String rcv;                              // String to hold received message
String rcv2;                             // String for receiving distance(2nd message)
int numberint2 = 0;                      // Integer to hold number after conversion in integer

// Receive message from transmitter
if (driver.recv(buf, &buflen)) {
for (int i = 0; i < buflen; i++) {
rcv = rcv + (char)buf[i];   // Convert hexadecimal to alpha characters
}
}

// Check received commands and take appropriate actions
if (rcv == "square") {
square();
}

if (rcv == "turn") {  // Used to calibrate and measure the acuracy of the turn function
turnAround();
}
if (rcv == "distance") {                 // Run the code for mission 1B
time = 0;                                // Make sure the time starts at 0
buflen = sizeof(buf);                    // Reset the buffer length for the second transmission
rcv2 = ""; // Initialize rcv2
while (!driver.recv(buf, &buflen)) {} // Wait until a new transmission is received
for (int i = 0; i < buflen; i++) {
rcv2 += (char)buf[i];                    // save the second received message
}
numberint = " ";  //initialize numberint
for (int i = 0; i < buflen; i++) {     // pull the distance from the second received message
if (isdigit(rcv2[i])) {
```

```
                numberint = numberint + rcv2[i];
                }
                }
            if (numberint.length() > 0) {
  numberint2 = numberint.toInt();      //convert the distance to an
                integer
  time = map(numberint2, 15, 30, 10083, 20166); // Map the distance to
                a time range
  distanceFunc(); // Call the function to perform distance measurement
                }
                }
              if (rcv == "stop") {
             stop(); // Stop the motors
                }
                }

        // Function to power both motors and drive forward
                void drive() {
             digitalWrite(Apin1, HIGH);
              digitalWrite(Apin2, LOW);
             digitalWrite(Bpin1, HIGH);
              digitalWrite(Bpin2, LOW);
               analogWrite(AENA, 250);
               analogWrite(BENA, 255);
          delay(2000); // Delay for a set duration
                }

        // Function to perform a 90 degree clockwise turn
                void turn() {
             digitalWrite(Apin2, LOW);
             digitalWrite(Apin1, LOW);
             digitalWrite(Bpin2, LOW);
             digitalWrite(Bpin1, LOW);
       digitalWrite(Apin2, HIGH); // Motor A forward
  digitalWrite(Bpin1, HIGH); // Motor B reversed to turn 90 degrees
analogWrite(AENA, 255);    // Set both motors to full speed in reverse
                directions
           analogWrite(BENA, 255);    //
          delay(250); // how long to turn
                }
```

```
// Function to perform final actions after completing a task
void finished() {
  digitalWrite(Apin2, HIGH);
  digitalWrite(Apin1, LOW);
  analogWrite(AENA, 255);
  digitalWrite(Bpin2, LOW);
  digitalWrite(Bpin1, HIGH);
  analogWrite(BENA, 255);
  delay(5000); // Delay for a set duration
}

// Function to stop both motors
void stop() {
  time = 0;
  digitalWrite(Apin2, LOW);
  digitalWrite(Apin1, LOW);
  digitalWrite(Bpin2, LOW);
  digitalWrite(Bpin1, LOW);
  Serial.print("time: ");
  Serial.print(time);
}

// Function to drive the robot in a square pattern
void square() {
  drive(); // Drive forward
  turn();  // Turn 90 degrees
  drive(); // Drive forward
  turn();  // Turn 90 degrees
  drive(); // Drive forward
  turn();  // Turn 90 degrees
  drive(); // Drive forward
  finished(); // Perform some final actions
  stop(); // Stop the motors
}

// Function to perform Mission 1B
void distanceFunc() {
digitalWrite(Apin2, LOW);       //Set both motors to drive straight
  digitalWrite(Apin1, HIGH);
```

```
  digitalWrite(Bpin2, LOW);
  digitalWrite(Bpin1, HIGH);
  analogWrite(AENA, 255);
  analogWrite(BENA, 255);
delay(time); // Run for the mapped time for the specified distance
  time = 0; //reset time to 0
  stop(); // Stop the motors
}

// Function to perform a full turn (180 degrees)
void turnAround() {
  turn();
  turn();
  digitalWrite(Apin2, LOW);
  digitalWrite(Apin1, LOW);
  digitalWrite(Bpin2, LOW);
  digitalWrite(Bpin1, LOW);
}
```

Code developed using

- https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/
- Chat GPT (https://chat.openai.com/)