Clap Controlled Car

Course Project

B.Sonu Praharshan (EC22B1011)

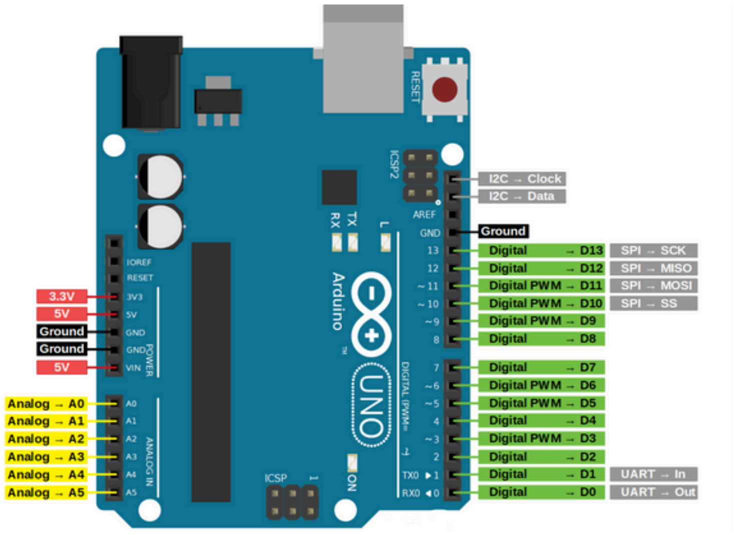
M.Advik (EC22B1014)

Theory:

Arduino Uno R3:

The Arduino Uno R3 is a microcontroller board featuring the ATmega328P microcontroller.It includes essential components such as digital and analog input/output pins, a voltage regulator, USB interface for programming and power, reset button, power pins, ICSP header for advanced programming, and onboard LEDs. It's widely used for prototyping and project development due to its simplicity, versatility, and extensive community support.

Working:



1. Powering:

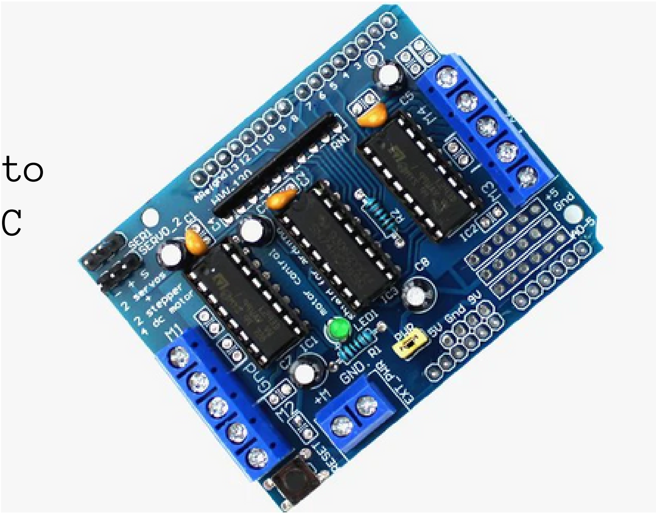
Connect via USB or external power supply (7-12V).
Voltage regulator ensures stable 5V supply.

2. Programming:

Use Arduino IDE.
Write code, upload via USB.
Features setup and loop functions.

3. Interfacing:
14 digital I/O pins, 6 analog input pins.
Supports UART, I2C, and SPI communication.
4. Troubleshooting:
Serial Monitor for debug messages.
Onboard LEDs for feedback.
Check wiring and connections.
5. Expansion:
Add-on shields for sensors, motor control, communication.

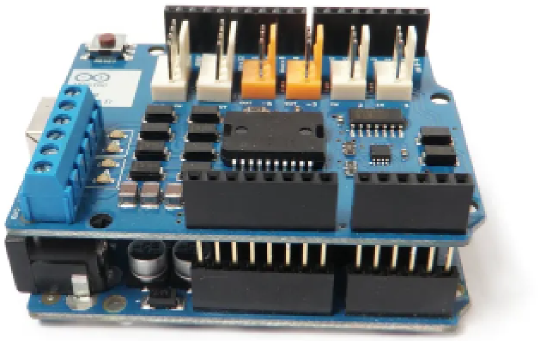
Arduino L293D Motor Driver Shield:



The Arduino L293D Motor Driver Shield is an expansion board designed to simplify the control of DC motors and stepper motors with an Arduino microcontroller.

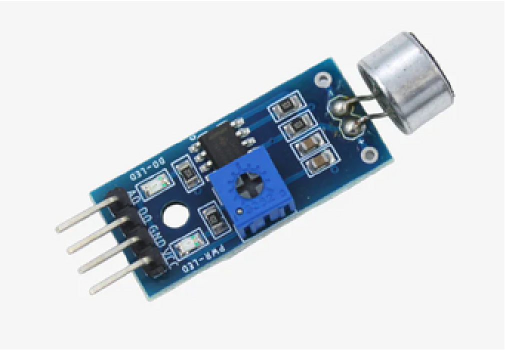
Working:

1.Initialize Motor Pins:
Initialize the digital pins used to control the motor driver.
Configure pins for direction control (e.g., IN1 and IN2 for one motor) and speed control (e.g., PWM pin for one motor).
2.Control Motor Direction:
Function to set the direction of the motor rotation.
Accept parameters such as forward, backward, left, or right.
Adjust the logic levels on the direction pins (e.g., IN1 and IN2) accordingly.



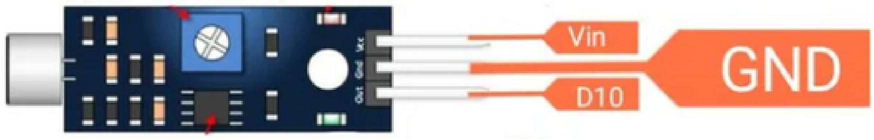
3.Control Motor Speed:
Function to control the speed of the motor.
Accept a speed value as input (e.g., PWM duty cycle).
Adjust the PWM signal on the speed control pin (e.g., ENA for one motor) to regulate motor speed.

LM393 Sound Sensor Module:



The LM393 Sound Sensor Module is equipped with an electret microphone that converts sound waves into electrical signals. These weak signals are then amplified by an operational amplifier (often LM393) to a level suitable for detection. The module outputs an analog voltage signal proportional to the detected sound level. By setting a threshold level, the module can trigger events when sound surpasses this threshold. This detection is facilitated by voltage comparators within the LM393 chip, which compare the amplified signal with the preset threshold. Clap detection, for instance, involves monitoring the module's output for transitions indicative of a clap sound, enabling it to trigger actions accordingly.

Working:



1.Initialize Pins:
Set up the digital pin connected to the sound sensor module for input.
2.Read Sound Level:
Read the analog value from the sound sensor module.
Convert the analog value to a meaningful sound level measurement, if required.
3.Clap Detection:
Implement a function to detect claps based on the sound level readings.
Define a threshold above which a sound is considered a clap.
4.Trigger Action on Clap:
Define actions to be triggered when a clap is detected.
These actions could include controlling motors, turning on/off LEDs, or any other desired behavior.

DC Geared Motor:

A DC Geared Motor combines a DC motor with a gearbox to achieve specific speed and torque requirements. The DC motor converts electrical energy into mechanical motion, while the gearbox modifies speed and torque through gear reduction. This combination allows for precise control over motor speed and torque, making DC Geared Motors suitable for various applications such as robotics, automation, and consumer electronics. Control methods include Pulse Width Modulation (PWM) for speed control and H-bridge circuits for direction control.

Apparatus:

- Arduino Uno
- L298D Motor Driver
- Sound Sensor
- 4-Gear Motors
- 4-Rubber Wheels
- Battery Holder
- 2-(3.7V)Battery

Procedure:

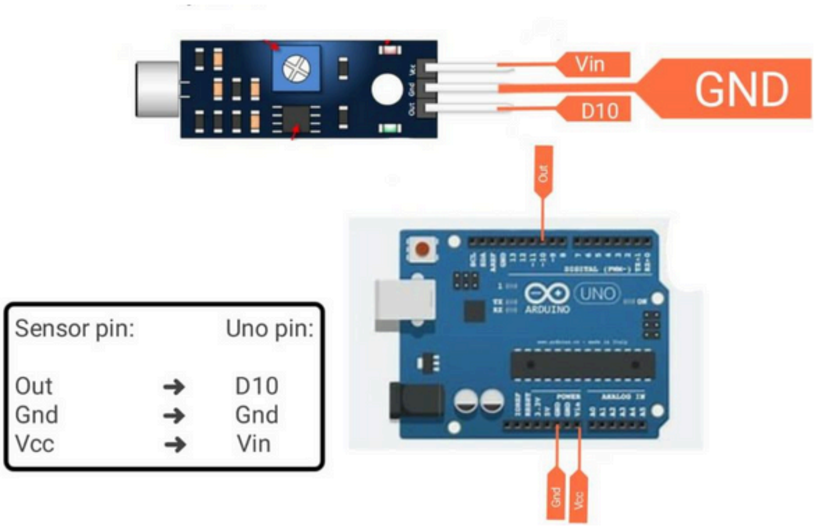
- Cut a piece of cardboard into a size of 10*14 cm to serve as the chassis.
- Use a hot glue gun to attach four gear motors onto the cardboard chassis.
- Solder wires to the "+" and "-" terminals of each motor for power connection.
- Place rubber wheels onto the gear motors to facilitate movement.
- Connect the motor driver to the Arduino Uno, aligning the pins correctly.
- Wire the motors to the motor driver by connecting the wires of each motor to the corresponding terminals (M1, M2, etc.) on the motor driver.
- Attach the sound sensor to the chassis using double-sided tape.
- Connect the sensor's pins (+5V, GND, OUT) to Arduino Uno as follows: OUT to D10, GND to GND, VCC to +5V.
- Fix a battery holder to the chassis using double-sided tape.
- Connect the GND wire of the holder to the motor driver's Power GND terminal. Connect the +5V wire to the VCC or +5V terminal of the motor driver.
- Plug the USB cable into the Arduino Uno for power and communication.
- Remove the OUT pin of the sound sensor temporarily for successful uploading of the code.

Code:

```
#include<AFMotor.h>
int sound = 10;
int st = 0;
AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);
int cont = 0;
void Forward()
{
 motor1.setSpeed(255);
 motor1.run(FORWARD);
 motor2.setSpeed(255);
 motor2.run(FORWARD);
 motor3.setSpeed(255);
 motor3.run(FORWARD);
 motor4.setSpeed(255);
 motor4.run(FORWARD);
 delay(1500);
 motor1.run(RELEASE);
 motor2.run(RELEASE);
 motor3.run(RELEASE);
 motor4.run(RELEASE);
}
void Backward()
{
 motor1.setSpeed(255);
 motor1.run(BACKWARD);
 motor2.setSpeed(255);
 motor2.run(BACKWARD);
 motor3.setSpeed(255);
 motor3.run(BACKWARD);
 motor4.setSpeed(255);
 motor4.run(BACKWARD);
 delay(1500);
 motor1.run(RELEASE);
 motor2.run(RELEASE);
 motor3.run(RELEASE);
 motor4.run(RELEASE);
}
void Left()
{
  motor1.setSpeed(255);
  motor1.run(BACKWARD);
  motor2.setSpeed(255);
  motor2.run(BACKWARD);
  motor3.setSpeed(255);
  motor3.run(FORWARD);
  motor4.setSpeed(255);
  motor4.run(FORWARD);
  delay(500);
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}
void Right()
{
  motor1.setSpeed(255);
  motor1.run(FORWARD);
  motor2.setSpeed(255);
  motor2.run(FORWARD);
  motor3.setSpeed(255);
  motor3.run(BACKWARD);
  motor4.setSpeed(255);
  motor4.run(BACKWARD);
  delay(500);
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
 }
```

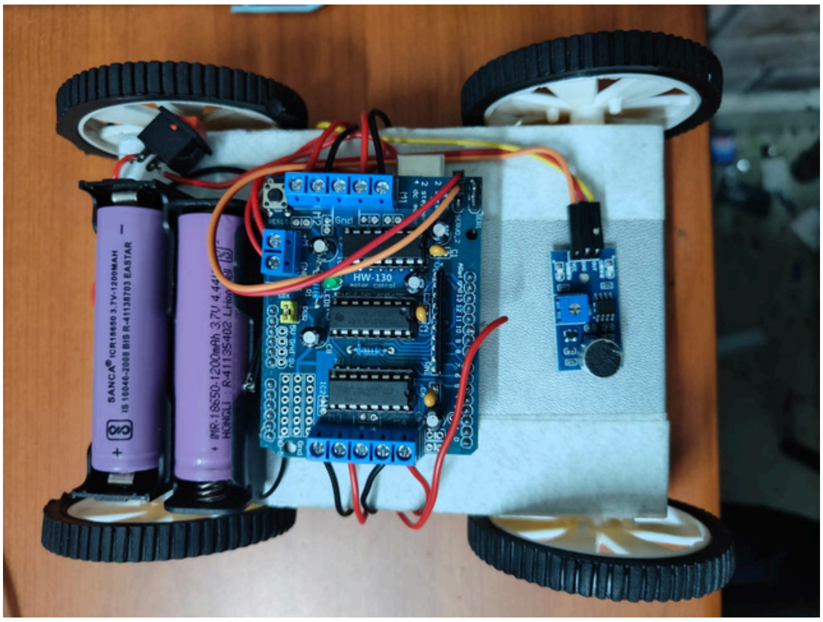
```
void setup()
{
//put your setup code here,to run once;
pinMode (sound, INPUT);
}
void loop()
{//put your main code here,to run repeatedly;
  if (digitalRead(sound) == HIGH)
  { delay(10);
    if (cont == 0)
      st = millis();
    cont = cont + 1;
    while (digitalRead(sound) != LOW)
    {  if (millis() - st > 2000)
      { Serial.print(cont);
        Serial.println(" aplausos");
        doOrders(cont);
        cont = 0;
   }}}
  if (millis() - st > 2000 && cont != 0) {
    Serial.print(cont);
    Serial.println(" aplausos");
    doOrders(cont);
    cont = 0;
  }}

void doOrders(int apl) {
 if (cont==1)
{Forward();}
else if (cont == 2)
{Backward();}
else if (cont == 3)
{Left();}
else if (cont == 4)
{Right();}}
```

Connections:



Sensor pin:		Uno pin:
Out	➔	D10
Gnd	➔	Gnd
Vcc	➔	Vin

Hardware:



Result:

After completing the assembly and uploading the code, I successfully created a clap-controlled car. The car moves forward when it detects a clap (sound above the threshold) using the sound sensor and stops when no sound is detected.