**MICRO PROJECT   
GROUP-10**

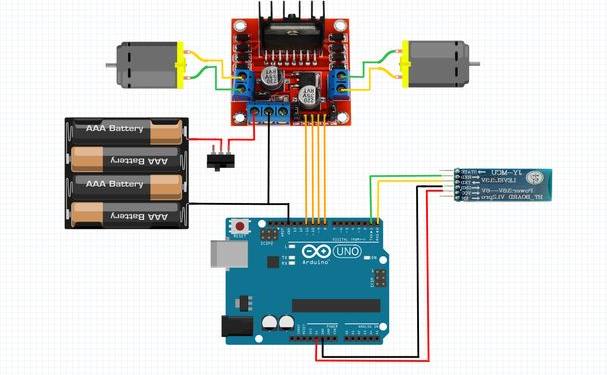
**Name: Niloy Sarker Name: Nasiat Hasan Fahim Name: Rezaul Karim  
Reg: 2020331033 Reg: 20203311013 Reg: 2020331029  
  
Name: Abid Ullah Muhib Name: Iqbal Mahmood Sajid  
Reg: 2020331089 Reg: 2020331103**

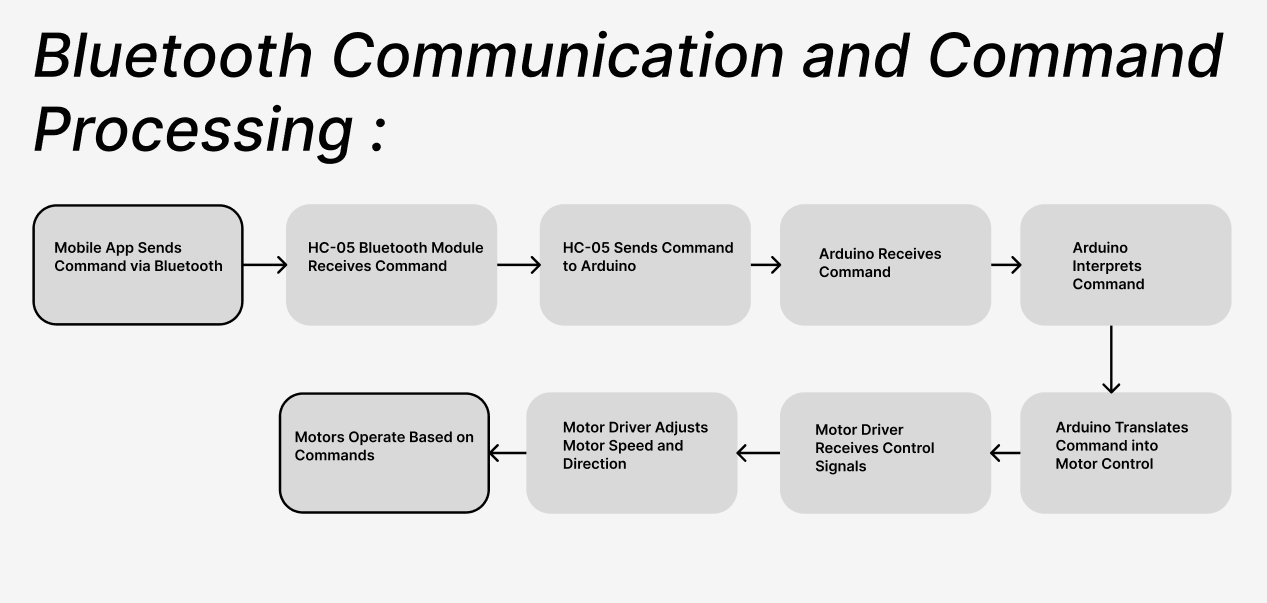
**Components and Their Functions**

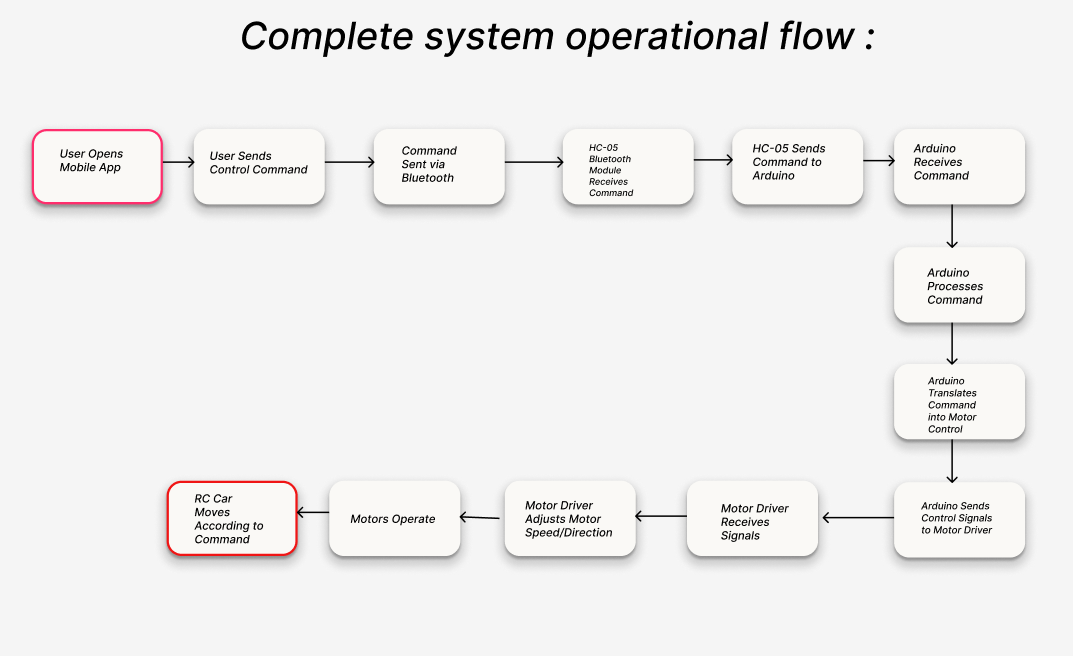
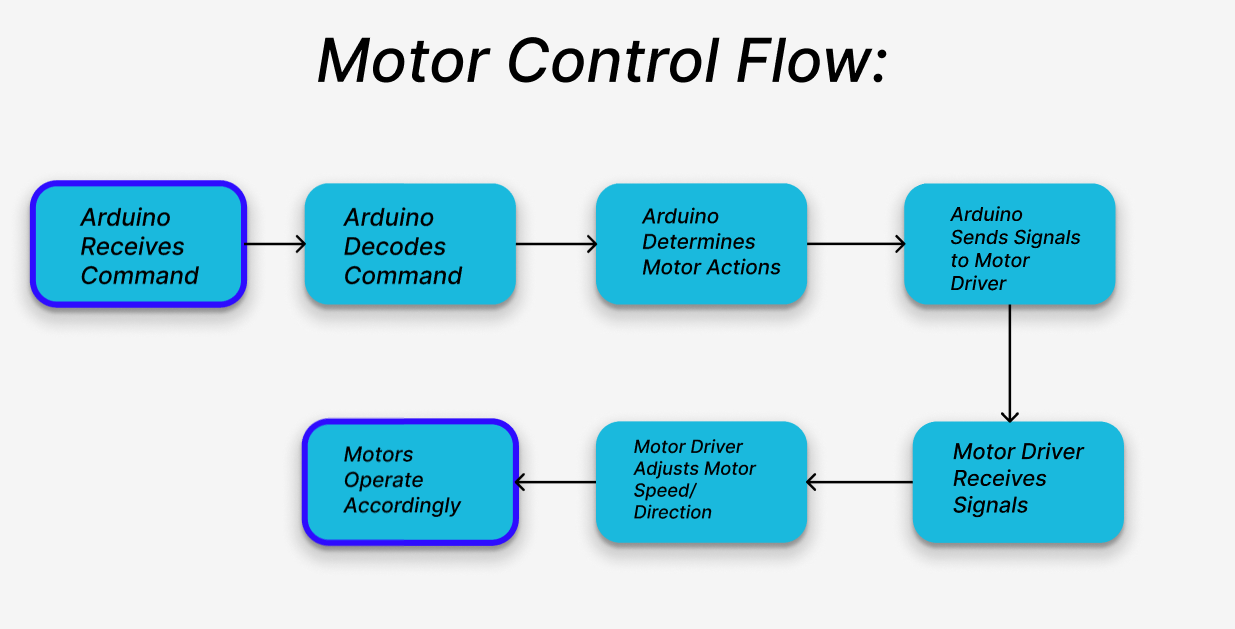
1. **Arduino Microcontroller:**
   * **Function:** The Arduino serves as the central controller of the RC car. It receives commands from the mobile app via the HC-05 Bluetooth module and processes these commands to control the car’s motors.
   * **Implementation:** We programmed the Arduino with a sketch that interprets Bluetooth signals and translates them into motor control commands. The sketch determines the direction and speed of the motors based on the received commands.
2. **HC-05 Bluetooth Module:**
   * **Function:** The HC-05 Bluetooth module enables wireless communication between the mobile app and the Arduino. It receives commands from the app and sends them to the Arduino.
   * **Implementation:** We paired the HC-05 with the mobile app through Bluetooth. We connected the HC-05’s TX pin to the Arduino’s RX pin and the HC-05’s RX pin to the Arduino’s TX pin. The module was powered through the Arduino’s 5V and GND pins and communicated data serially with the Arduino.
3. **Motor Driver (L298N):**
   * **Function:** The motor driver controls the direction and speed of the car’s motors. It acts as an interface between the Arduino and the motors, providing the necessary current and voltage to drive them.
   * **Implementation:** We connected the control pins of the L298N motor driver to the Arduino’s digital output pins. We also connected the motors to the output terminals of the motor driver. The motor driver adjusts the speed and direction of the motors based on the control signals from the Arduino and includes built-in protection against overcurrent and overheating.
4. **Motors:**
   * **Function:** The motors provide the mechanical movement for the RC car, driving the wheels and allowing the car to move forward, backward, and turn.
   * **Implementation:** We used two DC motors, one for each wheel, and connected them to the motor driver’s output terminals. The motor driver controlled their speed and direction based on commands received from the Arduino.
5. **Battery:**
   * **Function:** The battery supplies power to the entire RC car system, including the Arduino, motor driver, and motors.
   * **Implementation:** We used a rechargeable battery (typically 7.4V LiPo or similar) to power the car. We connected the positive and negative terminals of the battery to the power input terminals of the motor driver and also ensured that the Arduino was powered either from the motor driver’s output or directly from the battery.
6. **Mobile App:**
   * **Function:** The mobile app provides a user interface for controlling the RC car, sending commands via Bluetooth to the HC-05 module.
   * **Implementation:** We designed the app to include control buttons for forward, backward, left, and right movements. These commands were transmitted via Bluetooth to the HC-05, which then forwarded them to the Arduino. The app could be either a custom-built application or an existing Bluetooth control app.

**Making It from Scratch**

1. **Wiring and Connections:**
   * **HC-05 to Arduino:** We connected the HC-05 module’s TX pin to the Arduino’s RX pin and the HC-05’s RX pin to the Arduino’s TX pin. The VCC and GND pins of the HC-05 were connected to the Arduino’s 5V and GND, respectively.
   * **Motor Driver to Arduino:** We connected the control pins of the L298N motor driver to the Arduino’s digital output pins. The motor terminals were attached to the output pins of the motor driver.
   * **Motors to Motor Driver:** We connected the DC motors to the motor driver’s output terminals.
   * **Battery to Motor Driver and Arduino:** We connected the positive and negative terminals of the battery to the power input terminals of the motor driver. We ensured that the Arduino was powered appropriately, either from the motor driver’s output or directly from the battery.
2. **Programming:**
   * **Arduino Sketch:** We wrote and uploaded a sketch to the Arduino that handles Bluetooth communication and motor control. The sketch reads commands from the Bluetooth module, processes them, and sends appropriate signals to the motor driver.
3. **Testing:**
   * **Initial Testing:** We tested each component individually to ensure they functioned correctly. This involved verifying that the HC-05 could communicate with the mobile app and that the motor driver responded to commands from the Arduino.
   * **Integration Testing:** After assembling the complete system, we tested the RC car’s functionality with the mobile app. We ensured that the car responded correctly to all control commands and operated as expected.

****

 Fig: Circuit Diagram of Bluetooth Robot Controlled Car



Necessary Code

#define lmf 9

#define lmb 8

#define lms 10

#define rmf 5

#define rmb 4

#define rms 3

char ch;

int speed = 80;

void wheel(int a, int b){

  if(a>=0){

    digitalWrite(lmf, HIGH);

    digitalWrite(lmb, LOW);

  }

  else{

    a \*= -1;

    digitalWrite(lmf, LOW);

    digitalWrite(lmb, HIGH);

  }

  if(b>=0){

    digitalWrite(rmf, HIGH);

    digitalWrite(rmb, LOW);

  }

  else{

    b \*= -1;

    digitalWrite(rmf, LOW);

    digitalWrite(rmb, HIGH);

  }

  analogWrite(lms, a);

  analogWrite(rms, b);

}

void setup() {

  Serial.begin(115200);

}

void loop() {

  if(Serial.available()){

    ch = Serial.read();

    Serial.println(ch);

    if(ch == 'S'){

      wheel(0, 0);

    }

    if(ch >= '0' && ch <= '9'){

      speed = 70 + (20\*(ch - '0'));

      Serial.println(speed);

    }

    else if(ch == 'F'){

      wheel(speed, speed);

    }

    else if (ch == 'B') {

      wheel(-speed, -speed);

    }

    else if (ch == 'R') {

      wheel(speed, -speed);

    }

    else if (ch == 'L') {

      wheel(-speed, speed);

    }

  }

}