**BLUETOOTH CONTROLLED ROBOT**

|  |
| --- |
| // Motor Driver Pins  #define ENA 3   // Enable Pin for Motor 1  #define ENB 11  // Enable Pin for Motor 2  #define IN1 5   // Input 1 for Motor 1  #define IN2 6   // Input 2 for Motor 1  #define IN3 9   // Input 1 for Motor 2  #define IN4 10  // Input 2 for Motor 2  // Speed of motors (adjust as needed)  int motorSpeed = 150;  void setup() {    // Motor pins as output    pinMode(ENA, OUTPUT);    pinMode(ENB, OUTPUT);    pinMode(IN1, OUTPUT);    pinMode(IN2, OUTPUT);    pinMode(IN3, OUTPUT);    pinMode(IN4, OUTPUT);    // Start serial communication    Serial.begin(9600);  // Ensure HC-05 is set to 9600 baud rate  }  void loop() {    // Check if data is available    if (Serial.available() > 0) {      char command = Serial.read();  // Read the command from the Bluetooth module      // Process the received command      switch (command) {        case 'F':  // Move Forward          moveForward();          break;        case 'B':  // Move Backward          moveBackward();          break;        case 'L':  // Turn Left          turnLeft();          break;        case 'R':  // Turn Right          turnRight();          break;        case 'S':  // Stop          stopMotors();          break;        default:          // If an unknown command is received, stop the motors          stopMotors();          break;      }    }  }  // Function to move the car forward  void moveForward() {    digitalWrite(IN1, HIGH);    digitalWrite(IN2, LOW);    digitalWrite(IN3, HIGH);    digitalWrite(IN4, LOW);    analogWrite(ENA, motorSpeed);    analogWrite(ENB, motorSpeed);  }  // Function to move the car backward  void moveBackward() {    digitalWrite(IN1, LOW);    digitalWrite(IN2, HIGH);    digitalWrite(IN3, LOW);    digitalWrite(IN4, HIGH);    analogWrite(ENA, motorSpeed);    analogWrite(ENB, motorSpeed);  }  // Function to turn the car left  void turnLeft() {    digitalWrite(IN1, LOW);    digitalWrite(IN2, HIGH);    digitalWrite(IN3, HIGH);    digitalWrite(IN4, LOW);    analogWrite(ENA, motorSpeed);    analogWrite(ENB, motorSpeed);  }  // Function to turn the car right  void turnRight() {    digitalWrite(IN1, HIGH);    digitalWrite(IN2, LOW);    digitalWrite(IN3, LOW);    digitalWrite(IN4, HIGH);    analogWrite(ENA, motorSpeed);    analogWrite(ENB, motorSpeed);  }  // Function to stop the motors  void stopMotors() {    digitalWrite(IN1, LOW);    digitalWrite(IN2, LOW);    digitalWrite(IN3, LOW);    digitalWrite(IN4, LOW);  } |