Project1 :

```
#include <Servo.h>
#define speedL 3
#define IN1 9
#define IN2 8
#define IN3 7
#define IN4 6
#define speedR 5
#define flame 2
#define buzzer 10

char Reading;
int pos, flame_detected = 0;
Servo myservo;

void setup() {
  Serial.begin(9600);
  myservo.attach(11);
  myservo.write(90); // Center the servo

  for (int i = 3; i <= 9; i++) {
    pinMode(i, OUTPUT);
  }
  pinMode(flame, INPUT);
  pinMode(buzzer, OUTPUT);
}

void forword() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  analogWrite(speedL, 150);
  analogWrite(speedR, 150);
}

void backword() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  analogWrite(speedL, 150);
  analogWrite(speedR, 150);
}

void left() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  analogWrite(speedL, 0);
  analogWrite(speedR, 150);
}

void right() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
  analogWrite(speedL, 150);
  analogWrite(speedR, 0);
}

void stopp() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
  analogWrite(speedL, 0);
  analogWrite(speedR, 0);
}

void loop() {
  if (Serial.available() > 0) {
    Reading = Serial.read();
    switch (Reading) {
      case 'F': forword(); break;
      case 'B': backword(); break;
      case 'R': right(); break;
      case 'L': left(); break;
      case 'S': stopp(); break;
      case 'Q':
        // Sweep servo and check for flames
        for (pos = 90; pos <= 180; pos += 1) {
          if (digitalRead(flame) == 0) { flame_detected++; }
          myservo.write(pos);
          delay(15);
        }
        for (pos = 180; pos >= 0; pos -= 1) {
          if (digitalRead(flame) == 0) { flame_detected++; }
          myservo.write(pos);
          delay(15);
        }
        myservo.write(90); // Return servo to center

        // If flames were detected, sound the buzzer
        if (flame_detected > 0) {
          for (int i = 0; i < 10; i++) {
            digitalWrite(buzzer, 1);
            delay(100);
            digitalWrite(buzzer, 0);
            delay(100);
          }
          flame_detected = 0; // Reset flame counter
        }
        break;
    }
  }
}
```

Project2 :

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

RF24 radio(7, 8);

const byte address[6] = "00001";
```

```
void setup() {
  Serial.begin(9600);
  radio.begin();
  radio.openWritingPipe(address);
  radio.setPALevel(RF24_PA_MIN);
  radio.stopListening();
}

void loop() {
  const char text[] = "Abdallah";
  radio.write(&text, sizeof(text));
  Serial.println("Name sent!");
  delay(1000);
}
```

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

RF24 radio(7, 8);

const byte address[6] = "00001";

void setup() {
  Serial.begin(9600);
  radio.begin();
  radio.openReadingPipe(0, address);
  radio.setPALevel(RF24_PA_MIN);
  radio.startListening();
}

void loop() {
  if (radio.available()) {
    char text[32] = "";
    radio.read(&text, sizeof(text));
    Serial.print("Received: ");
    Serial.println(text);
  }
}
```