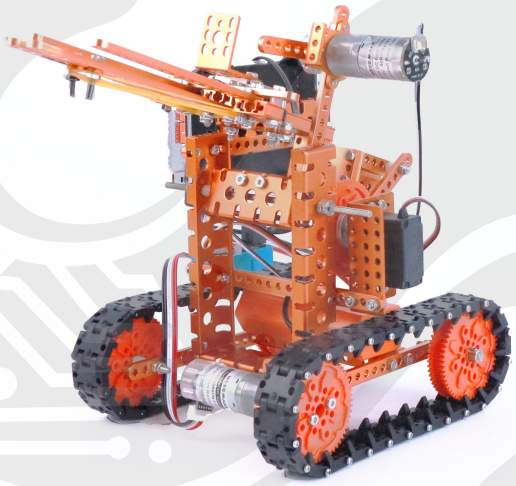
K. Ball Launcher

Material

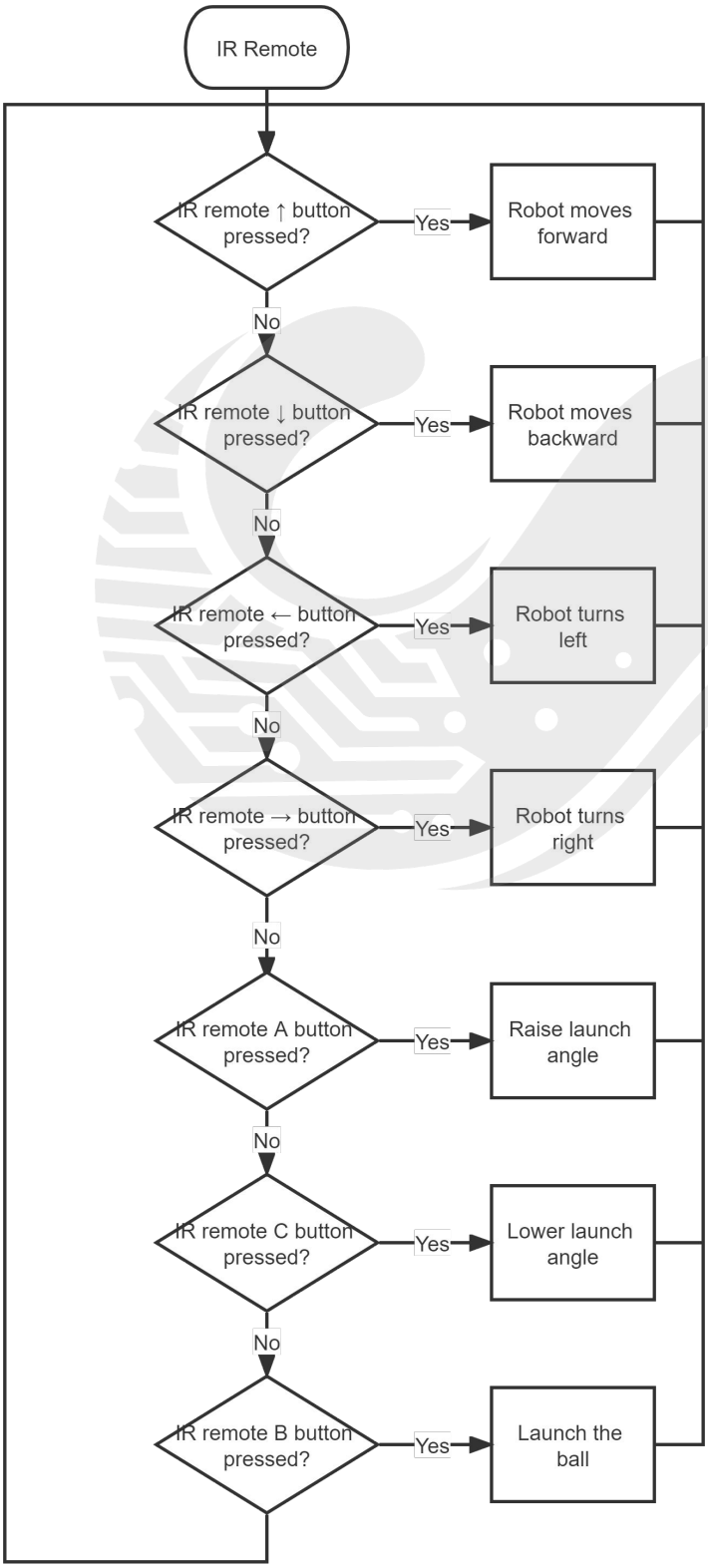
1. Ball Launcher Robot
2. Computer



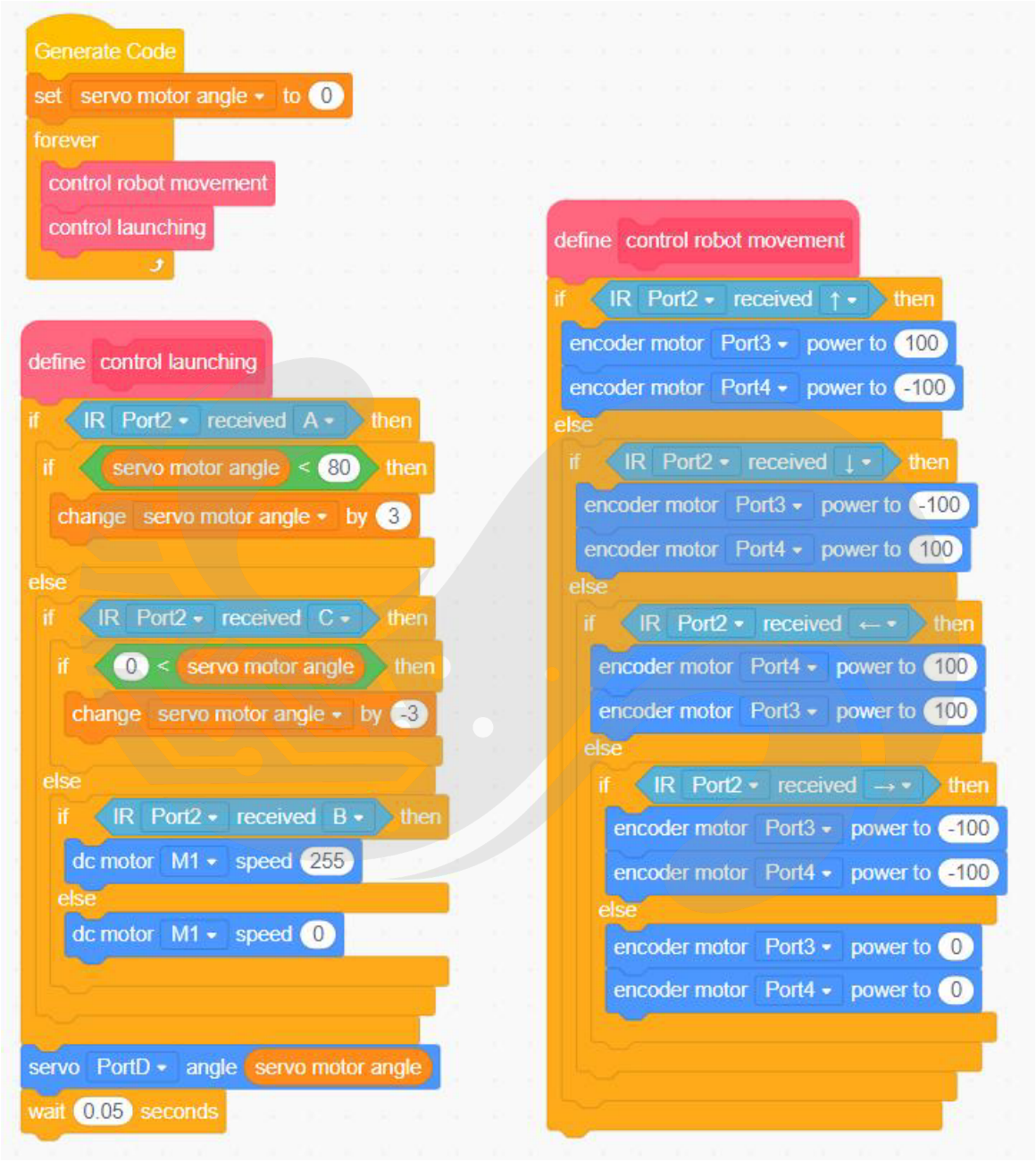
Project #1: Create a remote-controlled ball-launching robot.

In this educational activity, we will learn to operate the robot chassis using the up, down, left, and right buttons on the remote controller. We will also discover how to control the shooting platform's elevation by using the A and C buttons. Finally, we will have the opportunity to showcase their skills by initiating a launch with the press of the B button.

Program Idea



Reference Code

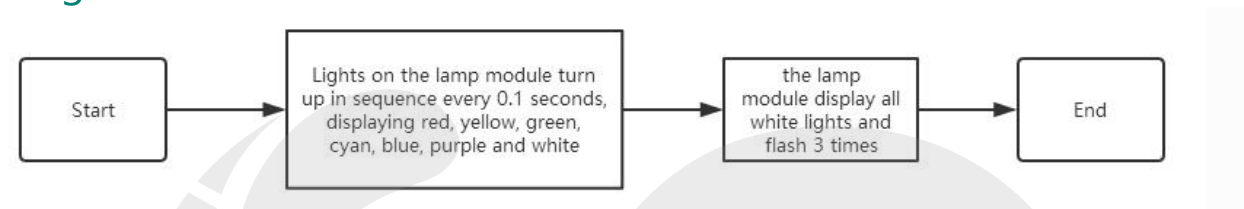


Note: **The 3D printing files for the ball launcher's storage box can be downloaded from the official website.**

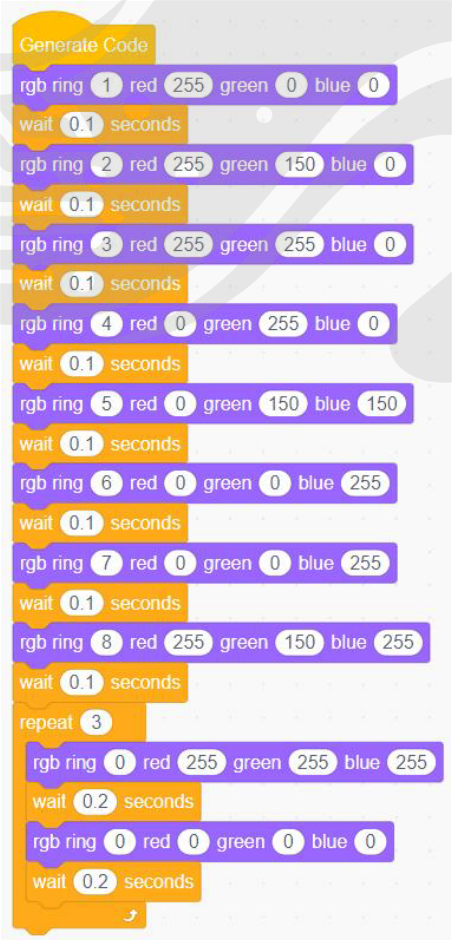
Project #2: Add launch prompt function

Utilize the RGB LED-8 module plugged on the mainboard to create a captivating visual display. The module will showcase a sequence of vibrant colors, including red, orange, yellow, green, cyan, blue, purple, and white. As a signal prompt preceding the launch of the projectile, All 8 RGB lights will flash three times in white in a visually striking pattern.

Program Idea



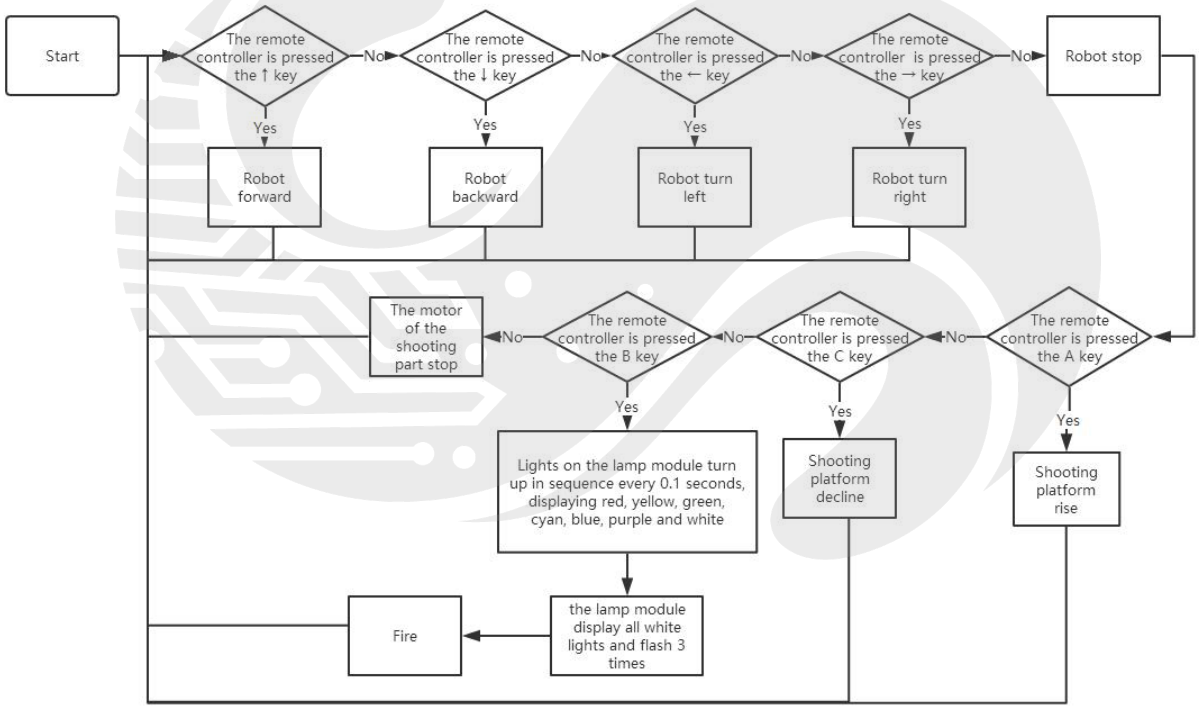
Reference Code



Project #3: Optimize the program of the remote-controlled ball-launching robot

By combining project 1 and 2, when the B button on the remote controller is pressed, the robot will initiate a light signal prompt before launching. Once the prompt is complete, the robot will proceed with the launching action.

Program Idea



Reference Code

```
Generate Code
set servo motor angle to 0
forever
  control robot movement
  control launching
```

```
define control robot movement
if IR Port2 received ↑ then
  encoder motor Port3 power to 100
  encoder motor Port4 power to -100
else
  if IR Port2 received ↓ then
    encoder motor Port3 power to -100
    encoder motor Port4 power to 100
  else
    if IR Port2 received ← then
      encoder motor Port4 power to 100
      encoder motor Port3 power to 100
    else
      if IR Port2 received → then
        encoder motor Port3 power to -100
        encoder motor Port4 power to -100
      else
        encoder motor Port3 power to 0
        encoder motor Port4 power to 0
```

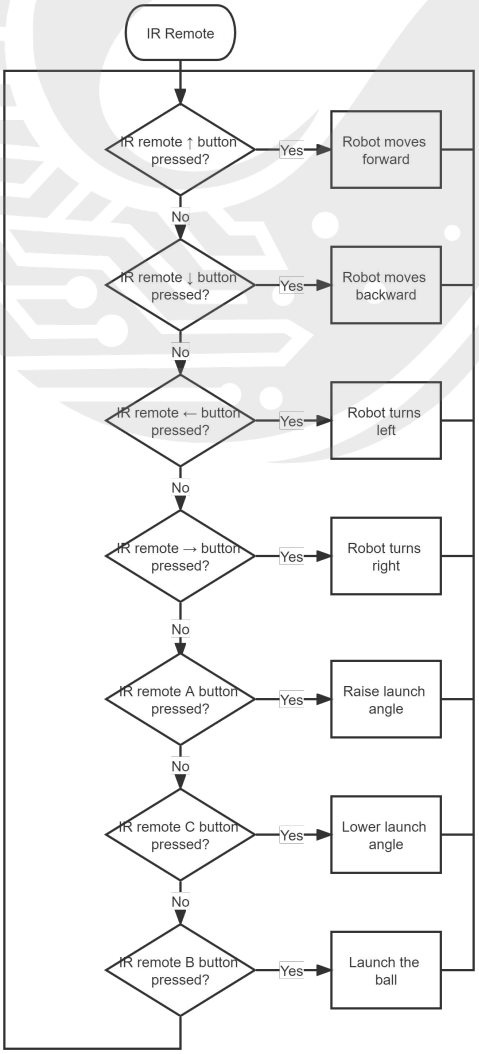
```
define control launching
if IR Port2 received A then
  if servo motor angle < 80 then
    change servo motor angle by 2
else
  if IR Port2 received C then
    if 0 < servo motor angle then
      change servo motor angle by -2
    else
      if IR Port2 received B then
        rgb ring 1 red 255 green 0 blue 0
        wait 0.1 seconds
        rgb ring 2 red 255 green 150 blue 0
        wait 0.1 seconds
        rgb ring 3 red 255 green 255 blue 0
        wait 0.1 seconds
        rgb ring 4 red 0 green 255 blue 0
        wait 0.1 seconds
        rgb ring 5 red 0 green 150 blue 150
        wait 0.1 seconds
        rgb ring 6 red 0 green 0 blue 255
        wait 0.1 seconds
        rgb ring 7 red 255 green 150 blue 255
        wait 0.1 seconds
        rgb ring 8 red 255 green 255 blue 255
        wait 0.1 seconds
        repeat 3
          rgb ring 0 red 255 green 255 blue 255
          wait 0.2 seconds
          rgb ring 0 red 0 green 0 blue 0
          wait 0.2 seconds
        dc motor M1 speed 255
        wait 0.8 seconds
        dc motor M1 speed 0
        wait 1 seconds
      else
        dc motor M1 speed 0
servo PortD angle servo motor angle
wait 0.05 seconds
```

ARDUINO

Project #1: Create a Remote-controlled Ball-launching Robot Arduino

In this educational activity, we will learn to operate the robot chassis using the up, down, left, and right buttons on the remote controller. We will also discover how to control the shooting platform's elevation by using the A and C buttons. Finally, we will have the opportunity to showcase their skills by initiating a launch with the press of the B button.

Program Idea



Reference Code

```
#include<WeELF328P.h>

void sleep(unsigned long duration);
//		control launching()
void f_control_launching();
//		control robot movement()
void f_control_robot_movement();
void update();

WeInfraredReceiver ir(PORT_2);
WeDCMotor dc_1(1);
Servo servo_D;
WeEncoderMotor encoder_3(PORT_3);
WeEncoderMotor encoder_4(PORT_4);
double v_servo_motor_angle;		//servo motor angle;

void setup(){
		ir.begin();
		servo_D.attach(PORT_D);

		v_servo_motor_angle = 0;		//servo motor angle
}

void loop(){
		update();
		f_control_robot_movement();		//control robot movement
		f_control_launching();	//control launching
}

void sleep(unsigned long duration){
		unsigned long end = millis() + duration;
		while(millis() < end)update();
}
```

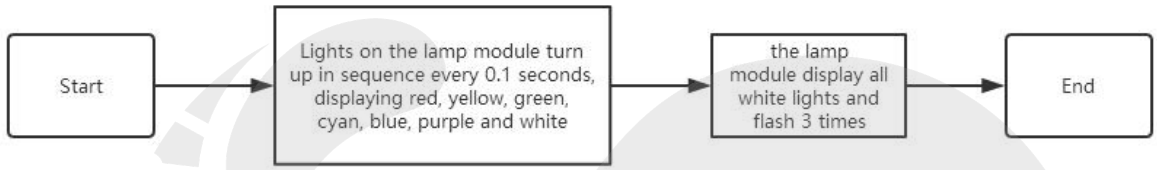
Reference Code

```
//		control launching()
void f_control_launching(){
		if(ir.isKeyPressed(IR_CONTROLLER_A)){
				if(v_servo_motor_angle/*servo motor angle*/ < 80){
				v_servo_motor_angle += 3;		//servo motor angle
				}
		}else if(ir.isKeyPressed(IR_CONTROLLER_C)){
				if(0 < v_servo_motor_angle/*servo motor angle*/){
				v_servo_motor_angle += -3;		//servo motor angle
				}
		}else if(ir.isKeyPressed(IR_CONTROLLER_B)){
				dc_1.run(255);
		}else{
				dc_1.run(0);
		}
		servo_D.write(v_servo_motor_angle/*servo motor angle*/);
		sleep(50);
}

//		control robot movement()
void f_control_robot_movement(){
		if(ir.isKeyPressed(IR_CONTROLLER_UP)){
				encoder_3.run(100);
				encoder_4.run(-100);
		}else if(ir.isKeyPressed(IR_CONTROLLER_DOWN)){
				encoder_3.run(-100);
				encoder_4.run(100);
		}else if(ir.isKeyPressed(IR_CONTROLLER_LEFT)){
				encoder_4.run(100);
				encoder_3.run(100);
		}else if(ir.isKeyPressed(IR_CONTROLLER_RIGHT)){
				encoder_3.run(-100);
				encoder_4.run(-100);
		}else{
				encoder_3.run(0);
				encoder_4.run(0);
		}
}

void update(){
		ir.loop();
}
```

Project #2: Add Launch Prompt Arduino

Utilize the RGB LED-8 module plugged on the mainboard to create a captivating visual display. The module will showcase a sequence of vibrant colors, including red, orange, yellow, green, cyan, blue, purple, and white. As a signal prompt preceding the launch of the projectile, All 8 RGB lights will flash three times in white in a visually striking pattern.

Program Idea



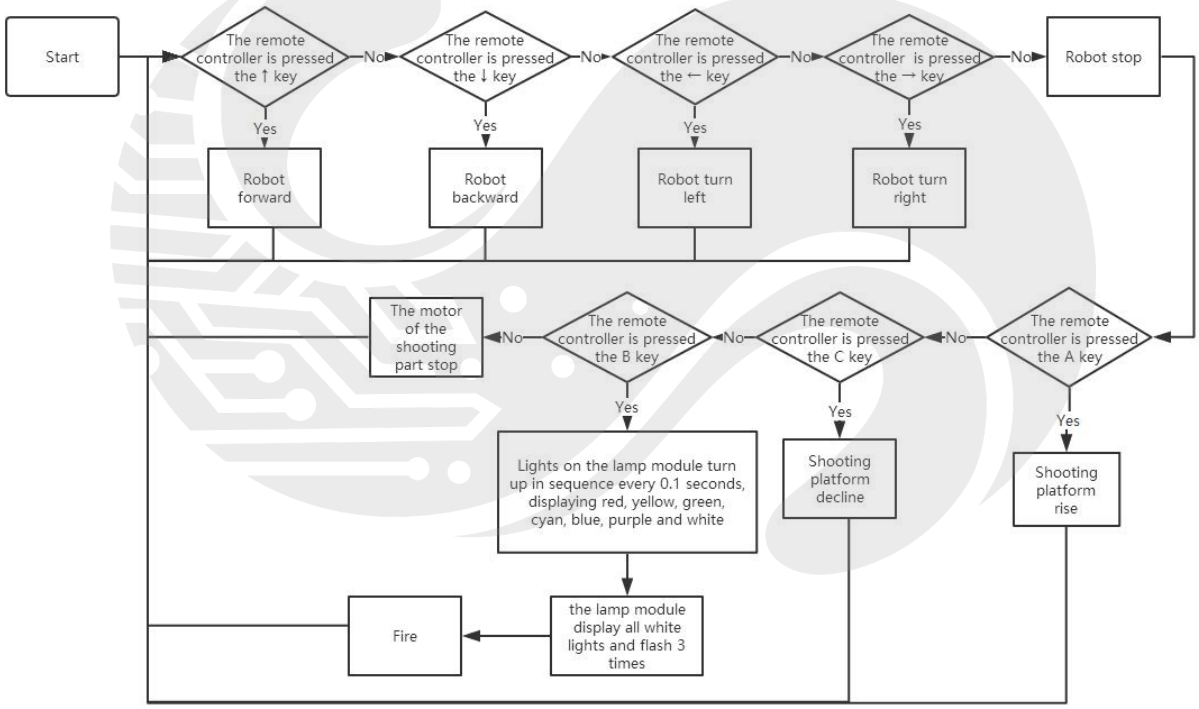
Reference Code

```
#include<WeELF328P.h>

WeRGBLed rgb_led_ring(PORT_0);

void setup(){
        rgb_led_ring.setColor(1, 255, 0, 0);
        rgb_led_ring.show();
        delay(100);
        rgb_led_ring.setColor(2, 255, 150, 0);
        rgb_led_ring.show();
        delay(100);
        rgb_led_ring.setColor(3, 255, 255, 0);
        rgb_led_ring.show();
        delay(100);
        rgb_led_ring.setColor(4, 0, 255, 0);
        rgb_led_ring.show();
        delay(100);
        rgb_led_ring.setColor(5, 0, 150, 150);
        rgb_led_ring.show();
        delay(100);
        rgb_led_ring.setColor(6, 0, 0, 255);
        rgb_led_ring.show();
        delay(100);
        rgb_led_ring.setColor(7, 0, 0, 255);
        rgb_led_ring.show();
        delay(100);
        rgb_led_ring.setColor(8, 255, 150, 255);
        rgb_led_ring.show();
        delay(100);
        for(int i=3; i>0; --i){
                rgb_led_ring.setColor(0, 255, 255, 255);
                rgb_led_ring.show();
                delay(200);
                rgb_led_ring.setColor(0, 0, 0, 0);
                rgb_led_ring.show();
                delay(200);
        }
}

void loop(){
}
```

Project #3: Optimize the Program of the Remote-controlled Ball-Launching Arduino

By combining project 1 and 2, when the B button on the remote controller is pressed, the robot will initiate a light signal prompt before launching. Once the prompt is complete, the robot will proceed with the launching action.

Program Idea



Reference Code

```
#include<WeELF328P.h>

void sleep(unsigned long duration);
//		control launching()
void f_control_launching();
//		control robot movement()
void f_control_robot_movement();
void update();

WeInfraredReceiver ir(PORT_2);
WeRGBLed rgb_led_ring(PORT_0);
WeDCMotor dc_1(1);
Servo servo_D;
WeEncoderMotor encoder_3(PORT_3);
WeEncoderMotor encoder_4(PORT_4);
double v_servo_motor_angle;		//servo motor angle;

void setup(){
		ir.begin();
		servo_D.attach(PORT_D);

		v_servo_motor_angle = 0;		//servo motor angle
}

void loop(){
		update();
		f_control_robot_movement();		//control robot movement
		f_control_launching();	//control launching
}

void sleep(unsigned long duration){
		unsigned long end = millis() + duration;
		while(millis() < end)update();
}

//		control launching()
void f_control_launching(){
		if(ir.isKeyPressed(IR_CONTROLLER_A)){
				if(v_servo_motor_angle/*servo motor angle*/ < 80){
				v_servo_motor_angle += 2;		//servo motor angle
				}
		}else if(ir.isKeyPressed(IR_CONTROLLER_C)){
				if(0 < v_servo_motor_angle/*servo motor angle*/){
				v_servo_motor_angle += -2;		//servo motor angle
```

Reference Code

```
}
        }else if(ir.isKeyPressed(IR_CONTROLLER_B)){
                rgb_led_ring.setColor(1, 255, 0, 0);
                rgb_led_ring.show();
                sleep(100);
                rgb_led_ring.setColor(2, 255, 150, 0);
                rgb_led_ring.show();
                sleep(100);
                rgb_led_ring.setColor(3, 255, 255, 0);
                rgb_led_ring.show();
                sleep(100);
                rgb_led_ring.setColor(4, 0, 255, 0);
                rgb_led_ring.show();
                sleep(100);
                rgb_led_ring.setColor(5, 0, 150, 150);
                rgb_led_ring.show();
                sleep(100);
                rgb_led_ring.setColor(6, 0, 0, 255);
                rgb_led_ring.show();
                sleep(100);
                rgb_led_ring.setColor(7, 255, 150, 255);
                rgb_led_ring.show();
                sleep(100);
                rgb_led_ring.setColor(8, 255, 255, 255);
                rgb_led_ring.show();
                sleep(100);
                for(int i=3; i>0; --i){
                        rgb_led_ring.setColor(0, 255, 255, 255);
                        rgb_led_ring.show();
                        sleep(200);
                        rgb_led_ring.setColor(0, 0, 0, 0);
                        rgb_led_ring.show();
                        sleep(200);
                }
                dc_1.run(255);
                sleep(800);
                dc_1.run(0);
                sleep(1000);
        }else{
                dc_1.run(0);
        }
        servo_D.write(v_servo_motor_angle/*servo motor angle*/);
        sleep(50);
}
```

Reference Code

```
//              control robot movement()
void f_control_robot_movement(){
                if(ir.isKeyPressed(IR_CONTROLLER_UP)){
                                encoder_3.run(100);
                                encoder_4.run(-100);
                }else if(ir.isKeyPressed(IR_CONTROLLER_DOWN)){
                                encoder_3.run(-100);
                                encoder_4.run(100);
                }else if(ir.isKeyPressed(IR_CONTROLLER_LEFT)){
                                encoder_4.run(100);
                                encoder_3.run(100);
                }else if(ir.isKeyPressed(IR_CONTROLLER_RIGHT)){
                                encoder_3.run(-100);
                                encoder_4.run(-100);
                }else{
                                encoder_3.run(0);
                                encoder_4.run(0);
                }
}

void update(){
                ir.loop();
}
```