**RGB светодиод**

Программный код

void setup() {

pinMode(RED, OUTPUT);

pinMode(GRN, OUTPUT);

pinMode(BLU, OUTPUT);

}

void loop() {

digitalWrite(RED, HIGH);

digitalWrite(GRN, LOW);

digitalWrite(BLU, LOW);

delay(1000);

digitalWrite(RED, LOW);

digitalWrite(GRN, HIGH);

digitalWrite(BLU, LOW);

delay(1000);

digitalWrite(RED, LOW);

digitalWrite(GRN, LOW);

digitalWrite(BLU, HIGH);

delay(1000);

}

1. ​с помощью директивы #define мы заменили номер пинов 11, 12 и 13 на соответствующие имена RED, GRN и BLU. Это сделано для удобства, чтобы не запутаться в скетче и понимать какой цвет мы включаем;
2. в процедуре void loop() мы поочередно включаем все три цвета на RGB.

Программный код для плавного мигания RGB светодиода

void setup() {

pinMode(RED, OUTPUT);

pinMode(GRN, OUTPUT);

pinMode(BLU, OUTPUT);

}

void loop() {

for (int i = 0; i <= 255; i++) {

analogWrite(RED, i);

delay(2);

}

for (int i = 255; i >= 0; i--) {

analogWrite(RED, i);

delay(2);

}

for (int i = 0; i <= 255; i++) {

analogWrite(GRN, i);

delay(2);

}

for (int i = 255; i >= 0; i--) {

analogWrite(GRN, i);

delay(2);

}

for (int i = 0; i <= 255; i++) {

analogWrite(BLU, i);

delay(2);

}

for (int i = 255; i >= 0; i--) {

analogWrite(BLU, i);

delay(2);

}

}

**Основная логика:**

Мы можем подавать на все три цветовых входа на светодиоде различное значение ШИМ-сигнала в диапазоне от 0 до 255, что позволит нам получить на RGB LED Arduino практически любой оттенок света

IV. Рекомендации:

1. Если хотите, чтобы лампа горела определенное время после срабатывания, добавьте в код таймер (используйте millis() вместо delay() для неблокирующей работы).

2. Для защиты Arduino от обратных токов реле добавьте транзистор и защитный диод (см. предыдущие инструкции).

3. Настройте датчик:

- Time Delay на минимум (чтобы управлять временем через код).

- Sensitivity под вашу зону обнаружения.

- Режим H (Repeatable Trigger), если хотите, чтобы датчик продлевал время работы при непрерывном движении.