

# DFRduino Beginner Kit For Arduino V3

## SKU:DFR0100

From Robot Wiki

### Contents

- 1 Introduction
- 2 Getting Started with Arduino
- 3 Tutorial
  - 3.1 1.Blinking a LED
  - 3.2 2.SOS Beacon
  - 3.3 3.Traffic Light
  - 3.4 4.Fading Light
  - 3.5 5.RGB LED
  - 3.6 6.Alarm
  - 3.7 7.Temperature Alarm
  - 3.8 8.Detecting vibration
  - 3.9 9.Ambient Light controlled LED
  - 3.10 10.Moving a Servo
  - 3.11 11.Interact with Servo
  - 3.12 12.RGB Light Dimmer
  - 3.13 13.Motor Fan
  - 3.14 14.Infrared controlled Light
  - 3.15 15.Infrared controlled LED Matrix
- 4 Old Version



DFRduino Beginner Kit For Arduino (SKU:DFR0100)

### Introduction

Welcome to the electronic interaction world! DFRobot proudly presents the Arduino beginner kit for those who are interested in learning about Arduino. Starting from basic LED control to more advanced IR remote control, this kit will guide you through the world of microcontrollers and physical computing.

A DFRduino UNO R3 (Compatible with Arduino Uno), the most stable and commonly used Arduino processor, together with DFRobot's best selling prototype shield are included in this kit.

# Getting Started with Arduino

Introduction (<http://arduino.cc/en/Guide/Introduction>): What Arduino is and why you'd want to use it.

Installation: Step-by-step instructions for setting up the Arduino software and connecting it to an Arduino Uno.

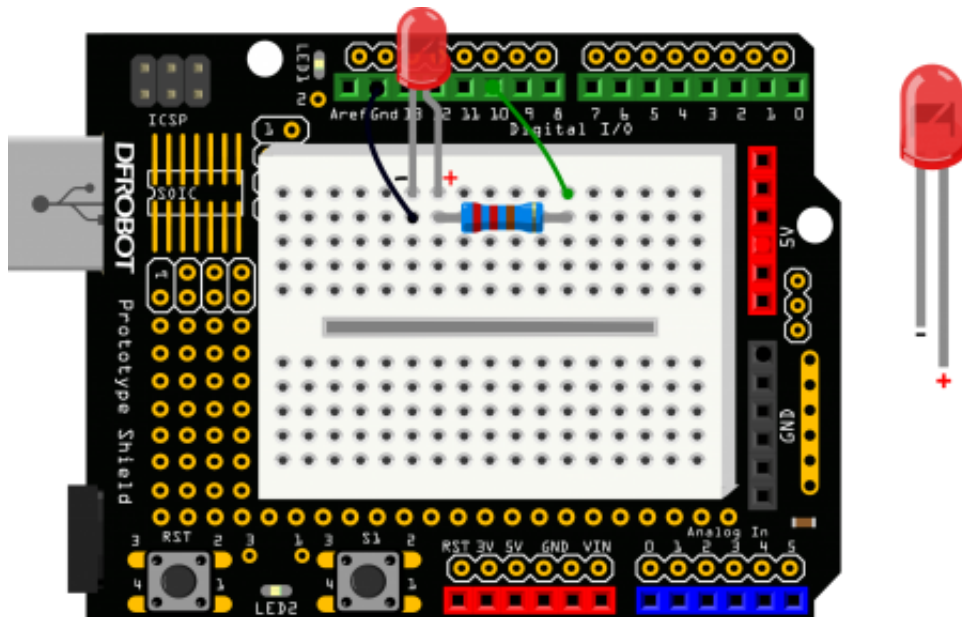
Windows (<http://arduino.cc/en/Guide/Windows>) Mac OS X (<http://arduino.cc/en/Guide/MacOSX>)

Environment (<http://arduino.cc/en/Guide/Environment>): Description of the Arduino development environment and how to change the default language.

Libraries (<http://arduino.cc/en/Guide/Libraries>): Using and installing Arduino libraries.

## Tutorial

### 1. Blinking a LED



```

?
1 /*
2  # Description:
3  # Turns on an LED on for one second, then off for one second, repeatedly.
4  */
5  int ledPin = 10;
6  void setup() {
7      pinMode(ledPin, OUTPUT);
8  }
9  void loop() {
10     digitalWrite(ledPin,HIGH);
11     delay(1000);

```

```
12     digitalWrite(ledPin,LOW);
13     delay(1000);
14}
```

## 2.SOS Beacon

The connection diagram is the same with Blinking a LED tutorial.

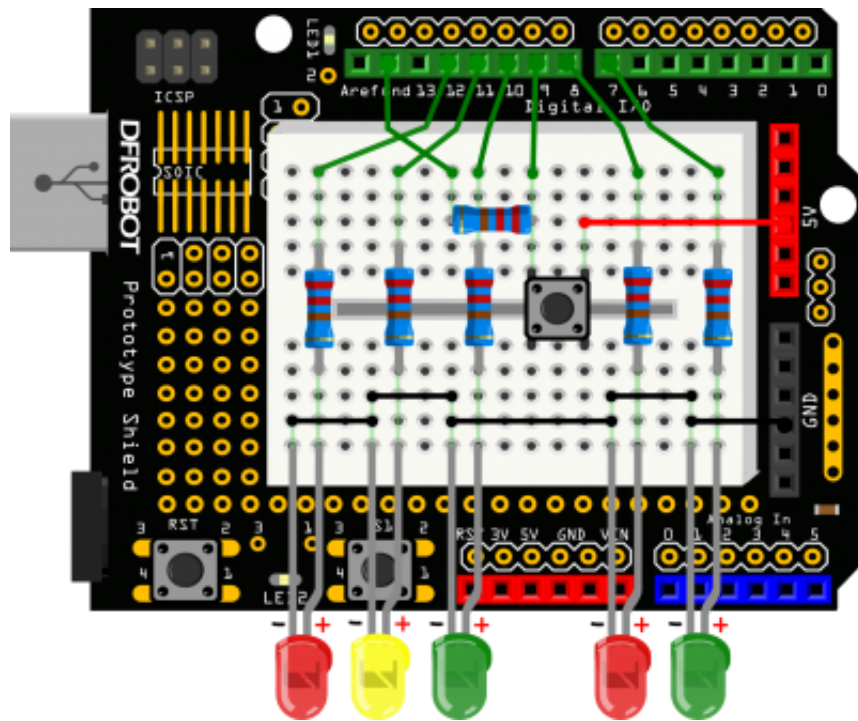
```
?
1 /*
2  # Description:
3  # Send SOS Beacon by led
4  */
5  int ledPin = 10;
6  void setup() {
7      pinMode(ledPin, OUTPUT);
8  }
9  void loop() {
10
11     // S(...) three dot
12     for(int x=0;x<3;x++){
13         digitalWrite(ledPin,HIGH);
14         delay(150);
15         digitalWrite(ledPin,LOW);
16         delay(100);
17     }
18
19
20     delay(100);
21     // O(---) three dash
22     for(int x=0;x<3;x++){
23         digitalWrite(ledPin,HIGH);
24         delay(400);
25         digitalWrite(ledPin,LOW);
26         delay(100);
27     }
28
29     delay(100);
30
31     //S(...) three dot
32     for(int x=0;x<3;x++){
33         digitalWrite(ledPin,HIGH);
34         delay(150);
35         digitalWrite(ledPin,LOW);
```

```

36     delay(100);
37     }
38
39     delay(5000);
40 }

```

### 3.Traffic Light



```

?
1 /*
2   Traffic Light
3   This code copied from the book Beginning-Arduino.
4 */
5 int carRed = 12; //assign the car lights
6 int carYellow = 11;
7 int carGreen = 10;
8 int button = 9; //button pin
9 int pedRed = 8; //assign the pedestrian lights
10 int pedGreen = 7;
11 int crossTime =5000; //time for pedestrian to cross
12 unsigned long changeTime;//time since button pressed
13
14 void setup() {
15     pinMode(carRed, OUTPUT);

```

```
16  pinMode(carYellow, OUTPUT);
17  pinMode(carGreen, OUTPUT);
18  pinMode(pedRed, OUTPUT);
19      pinMode(pedGreen, OUTPUT);
20      pinMode(button, INPUT);
21      digitalWrite(carGreen, HIGH); //turn on the green lights
22  digitalWrite(pedRed, HIGH);
23}
24
25void loop() {
26  int state = digitalRead(button);
27      //check if button is pressed and it is over 5 seconds since last button press
28      if(state == HIGH && (millis() - changeTime)> 5000){
29          //call the function to change the lights
30          changeLights();
31      }
32}
33
34void changeLights() {
35  digitalWrite(carGreen, LOW); //green off
36  digitalWrite(carYellow, HIGH); //yellow on
37  delay(2000); //wait 2 seconds
38
39  digitalWrite(carYellow, LOW); //yellow off
40  digitalWrite(carRed, HIGH); //red on
41      delay(1000); //wait 1 second till its safe
42
43  digitalWrite(pedRed, LOW); //ped red off
44  digitalWrite(pedGreen, HIGH); //ped green on
45
46  delay(crossTime); //wait for preset time period
47
48  //flash the ped green
49      for (int x=0; x<10; x++) {
50          digitalWrite(pedGreen, HIGH);
51          delay(250);
52          digitalWrite(pedGreen, LOW);
53          delay(250);
54      }
55
56      digitalWrite(pedRed, HIGH); //turn ped red on
57  delay(500);
58
59  digitalWrite(carRed, LOW); //red off
60  digitalWrite(carYellow, HIGH); //yellow on
```

```
61   delay(1000);
62   digitalWrite(carYellow, LOW); //yellow off
63       digitalWrite(carGreen, HIGH);
64
65   changeTime = millis(); //record the time since last change of lights
66   //then return to the main program loop
67}
```

## 4.Fading Light

The connection diagram is the same with Blinking a LED tutorial.

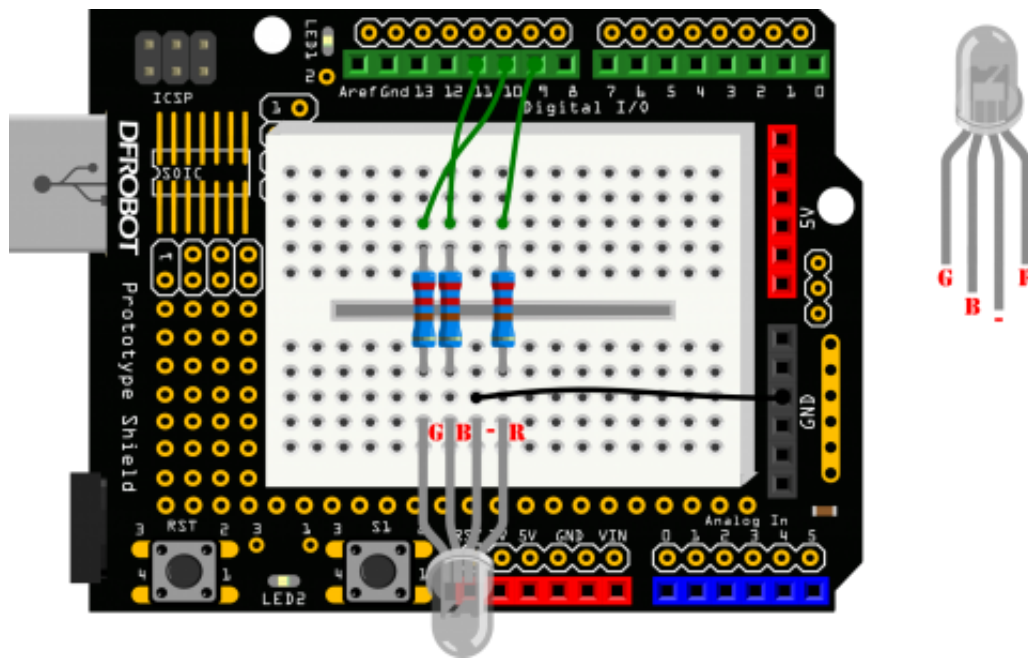
```
?
1 /*
2   Fading Light
3   This example shows how to fade an LED on pin 10 using the analogWrite() function.
4 */
5 int ledPin = 10;      // the pin that the LED is attached to
6
7 void setup() {
8     // declare pin 9 to be an output:
9     pinMode(ledPin,OUTPUT);
10    // initialize serial communication at 9600 bits per second:
11    Serial.begin(9600);
12}
13
14void loop(){
15    fadeOn(1000,5);
16    fadeOff(1000,5);
17}
18
19void fadeOn(unsigned int time,int increament){
20    //change the brightness by FOR statement
21    for (byte value = 0 ; value < 255; value+=increament){
22        // print out the value:
23        Serial.println(value);
24        // set the brightness of pin 10:
25        analogWrite(ledPin, value);
26        delay(time/(255/5));
27    }
28}
29
```

```

30 void fadeOff(unsigned int time,int decreament){
31     //change the brightness by FOR statement
32     for (byte value = 255; value >0; value-=decreament){
33         Serial.println(value);
34         analogWrite(ledPin, value);
35         delay(time/(255/5));
36     }
37 }

```

## 5.RGB LED



```

?
1 /*
2   RGB LED
3 */
4 int redPin = 9;    // the pin that the red LED is attached to
5 int greenPin = 10; // the pin that the green LED is attached to
6 int bluePin = 11; // the pin that the blue LED is attached to
7
8 void setup(){
9     pinMode(redPin, OUTPUT);
10    pinMode(greenPin, OUTPUT);
11    pinMode(bluePin, OUTPUT);
12
13}
14 void loop(){

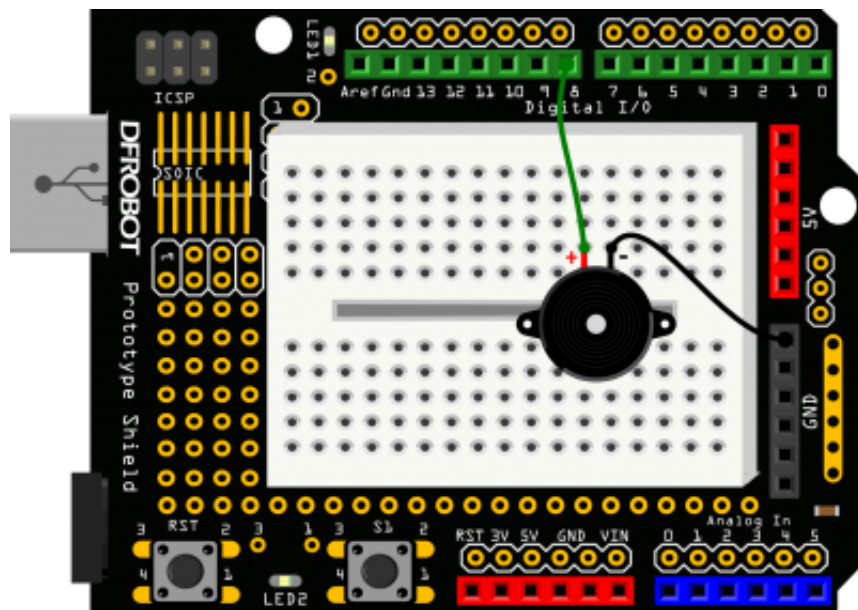
```

```

15 // call the function to change the colors of LED randomly.
16 colorRGB(random(0,255),random(0,255),random(0,255)); //R:0-255 G:0-255 B:0-255
17 delay(1000);
18}
19
20void colorRGB(int red, int green, int blue){
21   analogWrite(redPin,constrain(red,0,255));
22   analogWrite(greenPin,constrain(green,0,255));
23   analogWrite(bluePin,constrain(blue,0,255));
24}

```

## 6.Alarm



```

?
1 /*
2   Alarm
3 */
4 float sinVal;
5 int toneVal;
6
7 void setup(){
8   pinMode(8, OUTPUT);
9 }
10
11void loop(){
12   for(int x=0; x<180; x++){
13       // convert degrees to radians then obtain value

```



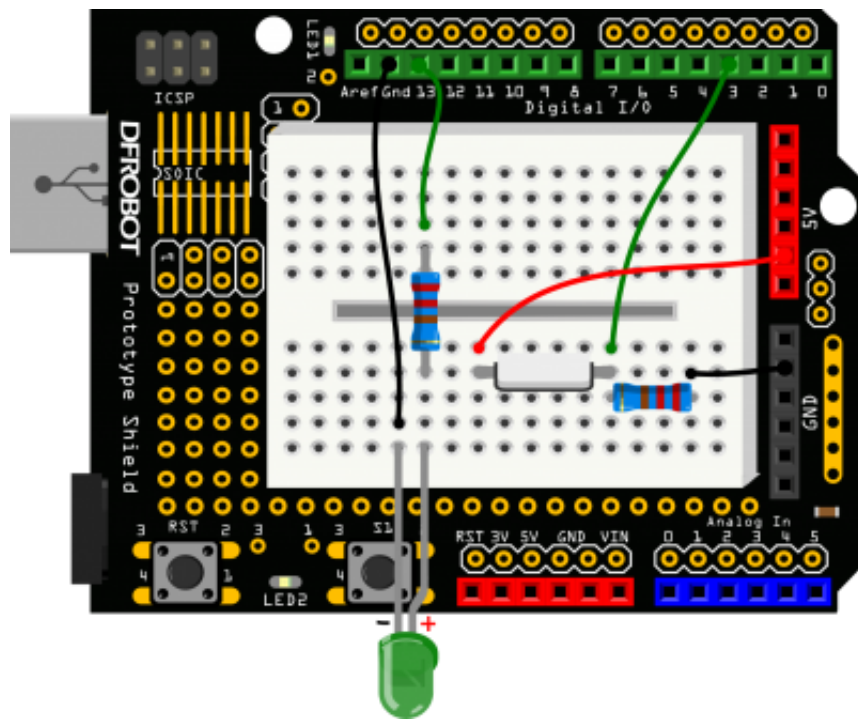


```

17  data = (double) val * (5/10.24); // convert the voltage to temperature
18
19  if(data>27){          // If the temperature is over 27 degree, buzzer will alarm.
20      for(int x=0; x<180; x++){
21          sinVal = (sin(x*(3.1412/180)));
22          toneVal = 2000+(int(sinVal*1000));
23          tone(8, toneVal);
24          delay(2);
25      }
26  } else {              // If the temperature is below 27 degree, buzzer will not alarm
27      noTone(8);
28  }
29
30  if(millis() - tepTimer > 500){ // output the temperature value per 500ms
31      tepTimer = millis();
32      Serial.print("temperature: ");
33      Serial.print(data);
34      Serial.println("C");
35  }
36}

```

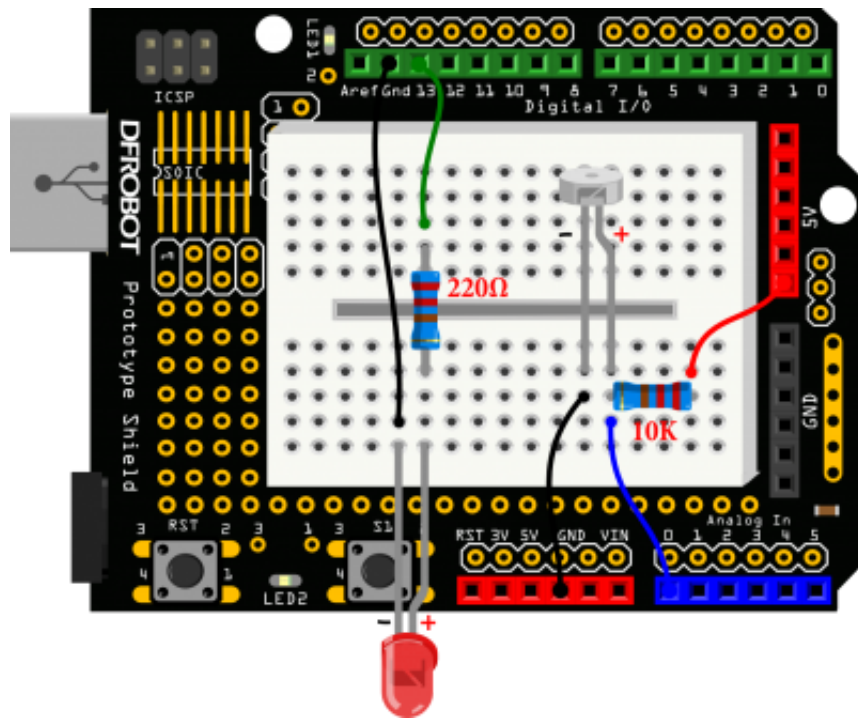
## 8. Detecting vibration



?

```
1 /*
2  Detecting vibration
3 */
4
5 int SensorLED = 13;      //LED PIN
6 int SensorINPUT = 3;    //Connect the sensor to digital Pin 3 which is Interrupts 1
7 unsigned char state = 0;
8
9 void setup() {
10  pinMode(SensorLED, OUTPUT);
11  pinMode(SensorINPUT, INPUT);
12
13  // Trigger the blink function when the falling edge is detected
14  attachInterrupt(1, blink, RISING);
15 }
16
17 void loop(){
18     if(state!=0){
19         state = 0;
20         digitalWrite(SensorLED,HIGH);
21         delay(500);
22     }
23     else
24         digitalWrite(SensorLED,LOW);
25 }
26
27 void blink(){            //Interrupts function
28     state++;
29 }
```

## 9.Ambient Light controlled LED

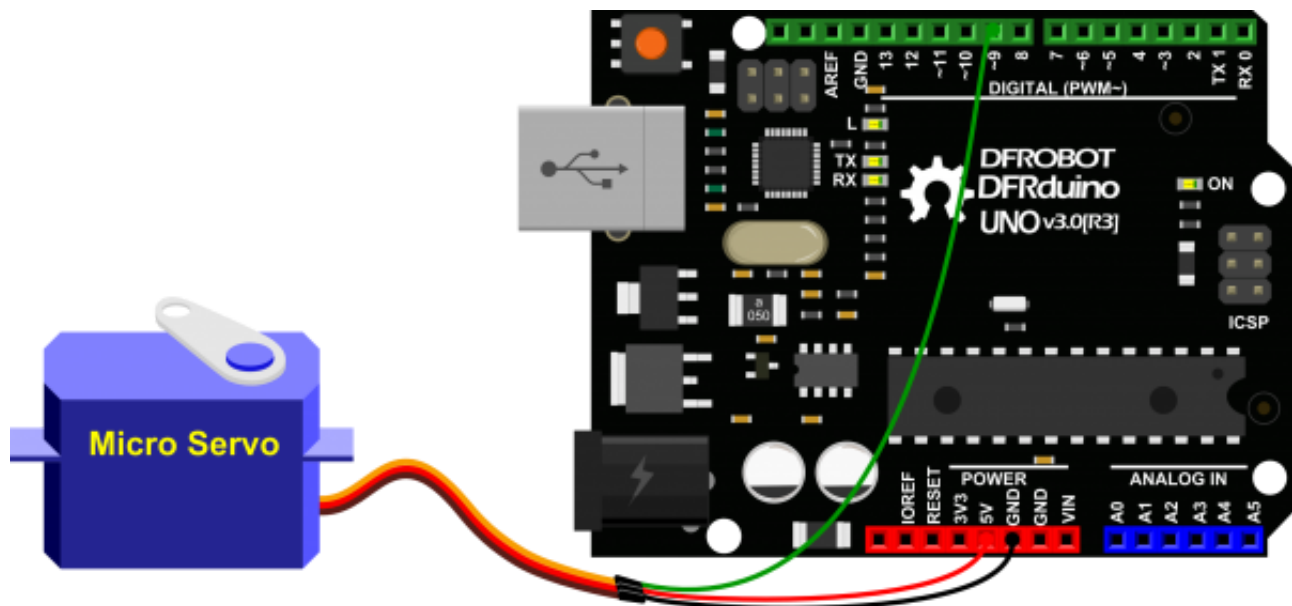


```

?
1  /*
2   Ambient Light controlled LED
3  */
4  int LED = 13;           //Led pin
5  int val = 0;
6
7  void setup(){
8      pinMode(LED,OUTPUT);
9      Serial.begin(9600);
10 }
11
12 void loop(){
13     val = analogRead(0);    // read voltage value
14     Serial.println(val);
15     if(val<1000){          // if the value is less than 1000, LED turns off
16         digitalWrite(LED,LOW);
17     }else{                 // if the value is more than 1000, LED turns on
18         digitalWrite(LED,HIGH);
19     }
20     delay(10);
21 }

```

## 10.Moving a Servo

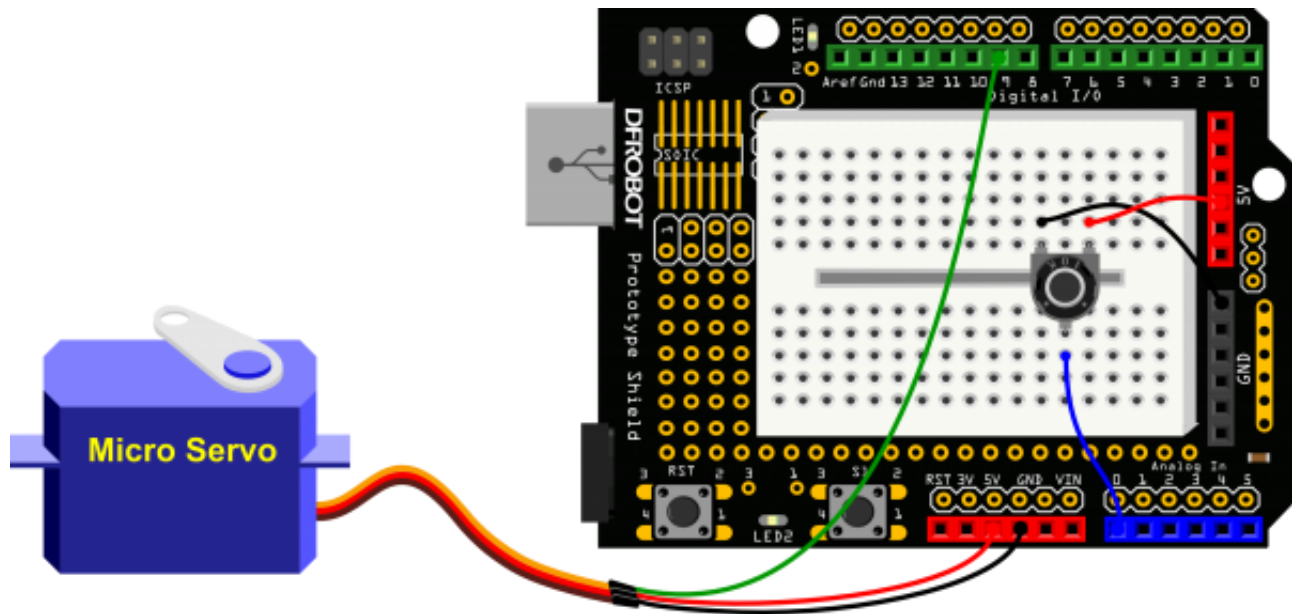


```

?
1 // Moving a Servo
2 // by BARRAGAN <http://barraganstudio.com>
3 // This example code is in the public domain.
4 #include <Servo.h>
5 Servo myservo;           // create servo object to control a servo
6                           // a maximum of eight servo objects can be created
7 int pos = 0;             // variable to store the servo position
8
9 void setup() {
10   myservo.attach(9);    // attaches the servo on pin 9 to the servo object
11 }
12
13 void loop() {
14   for(pos = 0; pos < 180; pos += 1){ // goes from 0 degrees to 180 degrees
15                                     // in steps of 1 degree
16     myservo.write(pos);           // tell servo to go to position in variable 'pos'
17     delay(15);                    // waits 15ms for the servo to reach the position
18   }
19
20   for(pos = 180; pos>=1; pos--=1) { // goes from 180 degrees to 0
21     degrees
22     myservo.write(pos);           // tell servo to go to position in variable 'pos'
23     delay(15);                    // waits 15ms for the servo to reach the position
24   }
25 }

```

## 11. Interact with Servo

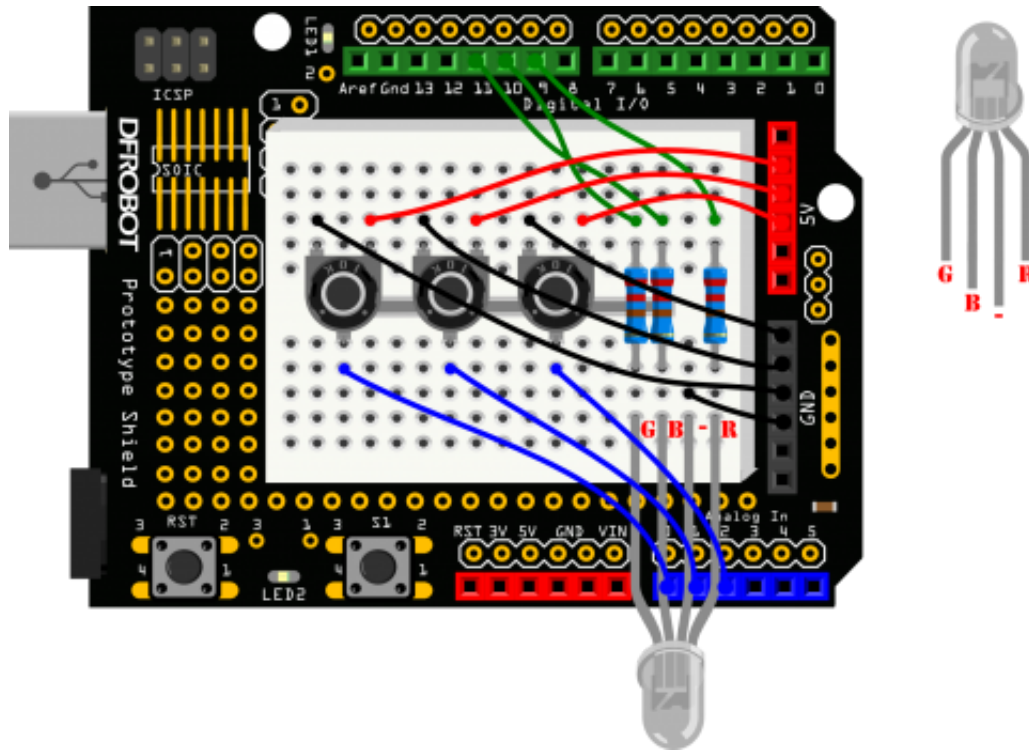


```

?
/*
1  Interact with Servo
2  Controlling a servo position using a potentiometer (variable resistor)
3  by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>
4 */
5
6 #include <Servo.h>
7 Servo myservo;           // create servo object to control a servo
8
9 int potpin = 0;          // analog pin used to connect the potentiometer
10 int val;                 // variable to read the value from the analog pin
11
12 void setup() {
13   myservo.attach(9);     // attaches the servo on pin 9 to the servo object
14 }
15
16 void loop() {
17   val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and
18   1023)
19   val = map(val, 0, 1023, 0, 179); // scale it to use it with the servo (value between 0 and
20   20180)
21   myservo.write(val);     // sets the servo position according to the scaled value
22   delay(15);             // waits for the servo to get there
  }

```

## 12.RGB Light Dimmer



```

?
1 /*
2   RGB Light Dimmer
3 */
4 int redPin = 9;           // R - digital 9
5 int greenPin = 10;       // G - digital 10
6 int bluePin = 11;        // B - digital 11
7 int potRedPin = 0;        // potentiometer 1 - analog 0
8 int potGreenPin = 1;     // potentiometer 2 - analog 1
9 int potBluePin = 2;      // potentiometer 3 - analog 2
10
11 void setup(){
12   pinMode(redPin,OUTPUT);
13   pinMode(greenPin,OUTPUT);
14   pinMode(bluePin,OUTPUT);
15   Serial.begin(9600);
16 }
17
18 void loop(){
19   int potRed = analogRead(potRedPin);
20   int potGreen = analogRead(potGreenPin);
21   int potBlue = analogRead(potBluePin);
22

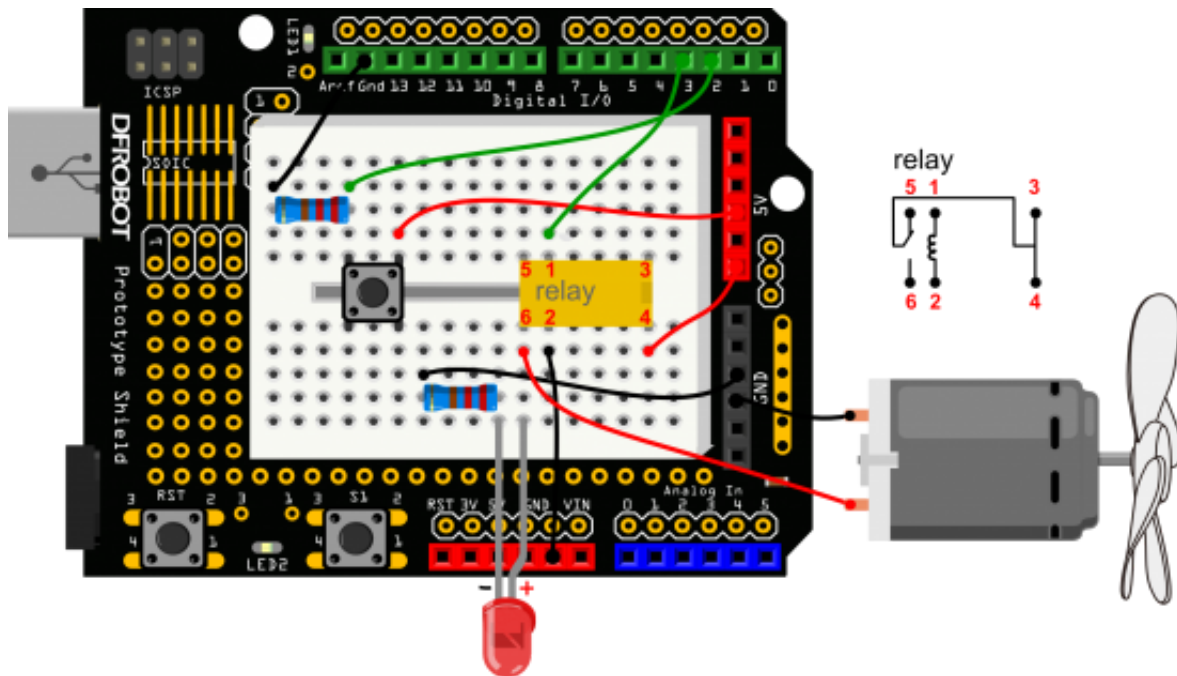
```

```

23 int val1 = map(potRed,0,1023,0,255);
24 int val2 = map(potGreen,0,1023,0,255);
25 int val3 = map(potBlue,0,1023,0,255);
26
27 Serial.print("Red:");
28 Serial.print(val1);
29 Serial.print("Green:");
30 Serial.print(val2);
31 Serial.print("Blue:");
32 Serial.println(val3);
33
34 colorRGB(val1,val2,val3);
35}
36
37void colorRGB(int red, int green, int blue){
38  analogWrite(redPin,constrain(red,0,255));
39  analogWrite(greenPin,constrain(green,0,255));
40  analogWrite(bluePin,constrain(blue,0,255));
41}

```

### 13.Motor Fan



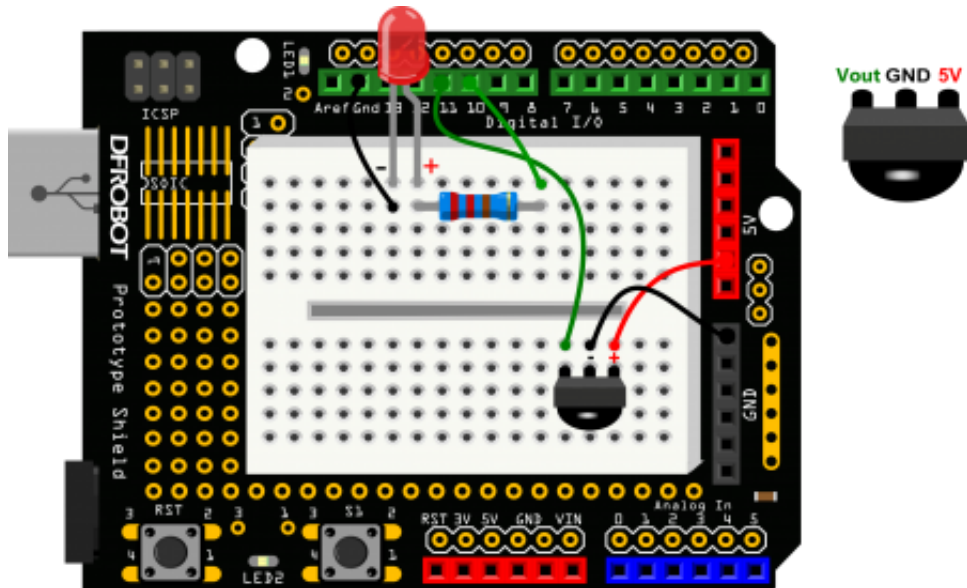
- ?
- 1 /\*
- 2 Motor Fan



```
3 */
4
5 int buttonPin = 2;           // button pin -- Digital 2
6 int relayPin = 3;           // relay pin -- Digital 3
7 int relayState = HIGH;
8 int buttonState;
9 int lastButtonState = LOW;
10 long lastDebounceTime = 0;
11 long debounceDelay = 50;
12
13 void setup() {
14   pinMode(buttonPin, INPUT);
15   pinMode(relayPin, OUTPUT);
16
17   digitalWrite(relayPin, relayState);
18 }
19 void loop() {
20   // read the state of the switch into a local variable:
21   int reading = digitalRead(buttonPin);
22
23   // check to see if you just pressed the button
24   // (i.e. the input went from LOW to HIGH), and you've waited
25   // long enough since the last press to ignore any noise:
26
27   // If the switch changed, due to noise or pressing:
28   if (reading != lastButtonState) {
29     lastDebounceTime = millis();
30   }
31
32   if ((millis() - lastDebounceTime) > debounceDelay) {
33     // whatever the reading is at, it's been there for longer
34     // than the debounce delay, so take it as the actual current state:
35
36     // if the button state has changed:
37     if (reading != buttonState) {
38       buttonState = reading;
39
40       // only toggle the Relay if the new button state is HIGH
41       if (buttonState == HIGH) {
42         relayState = !relayState;
43       }
44     }
45   }
46
47   // set the relay:
```

```
48 digitalWrite(relayPin, relayState);
49
50 // save the reading. Next time through the loop,
51 // it'll be the lastButtonState:
52 lastButtonState = reading;
53}
```

## 14.Infrared controlled Light





```

?
1 /*
2     Infrared controlled Light
3 */
4 #include <IRremote.h>
5 int RECV_PIN = 11;
6 int ledPin = 10;
7 boolean ledState = LOW;
8 IRrecv irrecv(RECV_PIN);
9 decode_results results;
10
11 void setup(){
12   Serial.begin(9600);
13   irrecv.enableIRIn();
14   pinMode(ledPin,OUTPUT);
15 }
16
17 void loop() {
18   if (irrecv.decode(&results)) {
19     Serial.println(results.value, HEX);
20

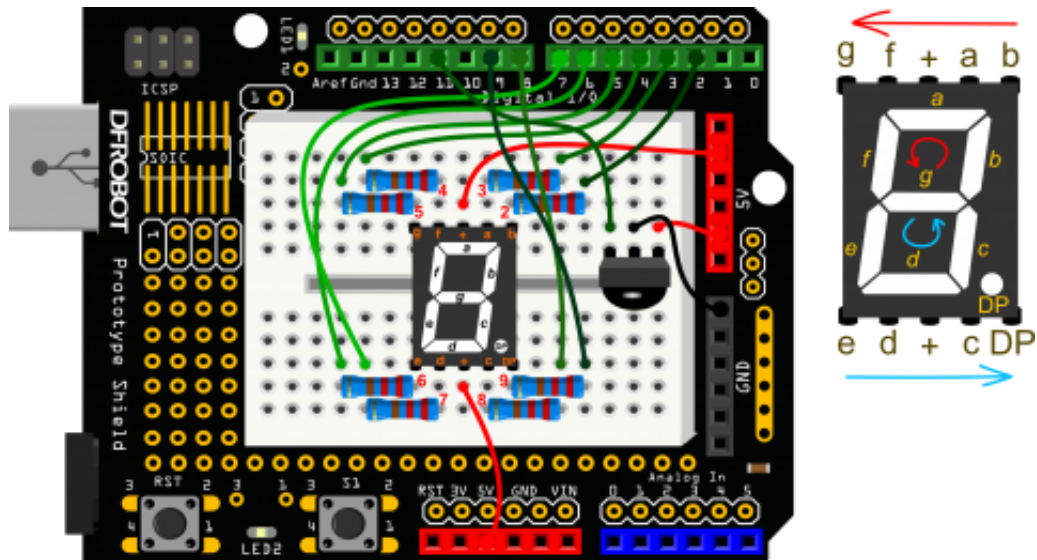
```

```

21     if(results.value == 0xFD00FF){
22         ledState = !ledState;
23         digitalWrite(ledPin,ledState);
24     }
25     irrecv.resume();
26 }
27}

```

## 15.Infrared controlled LED Matrix





```

?
1 #include <IRremote.h>
2 int RECV_PIN = 11;
3 IRrecv irrecv(RECV_PIN);
4 decode_results results;
5 int currentNumber = 0;
6
7 long codes[12]=                // this array is used to store infrared codes
8 {
9     0xFD30CF,0xFD08F7,          // 0 ,1
10    0xFD8877,0xFD48B7,          // 2 ,3
11    0xFD28D7,0xFDA857,          // 4 ,5
12    0xFD6897,0xFD18E7,          // 6 ,7
13    0xFD9867,0xFD58A7,          // 8 ,9
14    0xFD20DF,0xFD609F,          // advance, move back
15};
16
17int number[10][8] =             //the array is used to store the number 0~9
18{
19 {0,0,0,1,0,0,0,1},//0
20 {0,1,1,1,1,1,0,1},//1
21 {0,0,1,0,0,0,1,1},//2
22 {0,0,1,0,1,0,0,1},//3
23 {0,1,0,0,1,1,0,1},//4

```

```
24 {1,0,0,0,1,0,0,1},//5
25 {1,0,0,0,0,0,0,1},//6
26 {0,0,1,1,1,1,0,1},//7
27 {0,0,0,0,0,0,0,1},//8
28 {0,0,0,0,1,1,0,1} //9
29};
30
31void numberShow(int i) {           //this function is used to display numbers
32  for(int pin = 2; pin <= 9 ; pin++){
33    digitalWrite(pin, number[i][pin - 2]);
34  }
35}
36
37void setup(){
38  Serial.begin(9600);
39  irrecv.enableIRIn();
40
41  for(int pin = 2 ; pin <= 9 ; pin++){
42    pinMode(pin, OUTPUT);
43    digitalWrite(pin, HIGH);
44  }
45}
46
47void loop() {
48  if (irrecv.decode(&results)) {
49    for(int i = 0; i <= 11; i++){
50
51      if(results.value == codes[i]&& i <= 9){
52        numberShow(i); // display number 0~9 on the digital control
53        currentNumber = i;
54        Serial.println(i);
55        break;
56      }
57
58      else if(results.value == codes[10]&& currentNumber != 0){
59        currentNumber--;
60        numberShow(currentNumber);
61        Serial.println(currentNumber);
62        break;
63      }
64
65      //
66      else if(results.value == codes[11]&& currentNumber != 9){
67        currentNumber++;
68        numberShow(currentNumber);
```

```
69         Serial.println(currentNumber);
70         break;
71     }
72 }
73
74 Serial.println(results.value, HEX);
75 irrecv.resume();
76 }
77}
```

- 
- This page was last modified on 30 June 2014, at 09:23.