

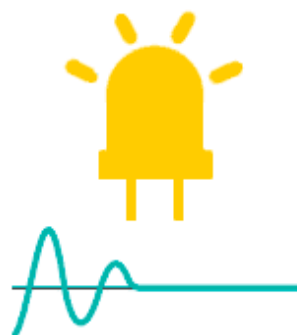
LUIS LLAMAS

Ingeniería, informática y diseño

TUTORIALES ARDUINO TUTORIALES ARDUINO AVANZADOS

CÓMO HACER UN CONTROL PID DE ILUMINACIÓN CONSTANTE CON ARDUINO

27 JULIO, 2021



En esta entrada vamos a ver cómo hacer un control de iluminación constante por PID con una simple Led y una resistencia LDR.

Llevamos mucho hablado sobre PID en el blog en la sección de [teoría de control](#). Así hemos visto [el controlador PID](#), [cómo ajustar un controlador PID](#) y [cómo implementar un control PID en Arduino](#)

Ahora toca pasar a un ejemplo práctico de PID. Uno de los más apropiados para empezar es un control de iluminación, porque el montaje es muy sencillo y además podemos visualizar fácilmente el resultado.

Únicamente necesitamos enfrentar un LED con una resistencia LDR, aunque valdría cualquier otro tipo de emisor o sensor de luz. En el montaje, enfrentamos el LED con la resistencia LDR de la siguiente forma.

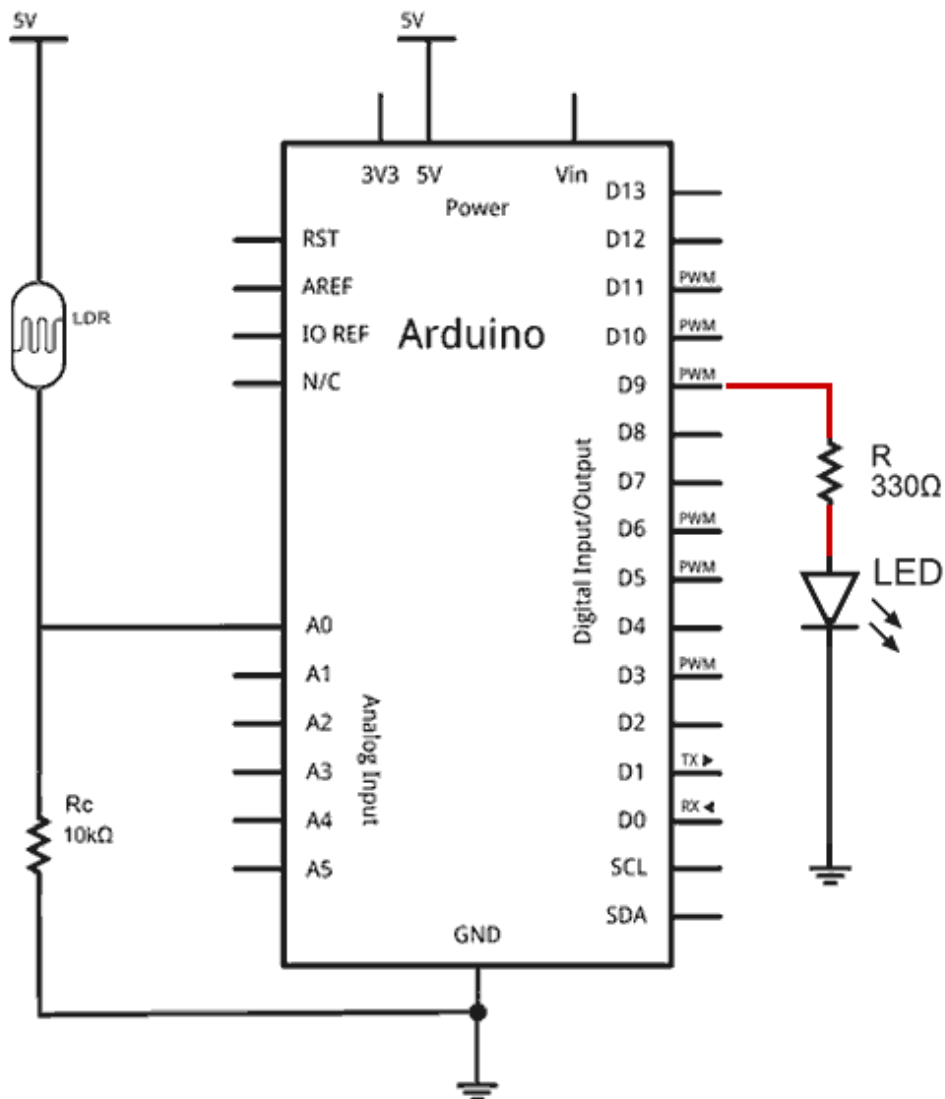
Anuncio:

Aire
Acondicionado
Midea 2550W...

\$ 94.999

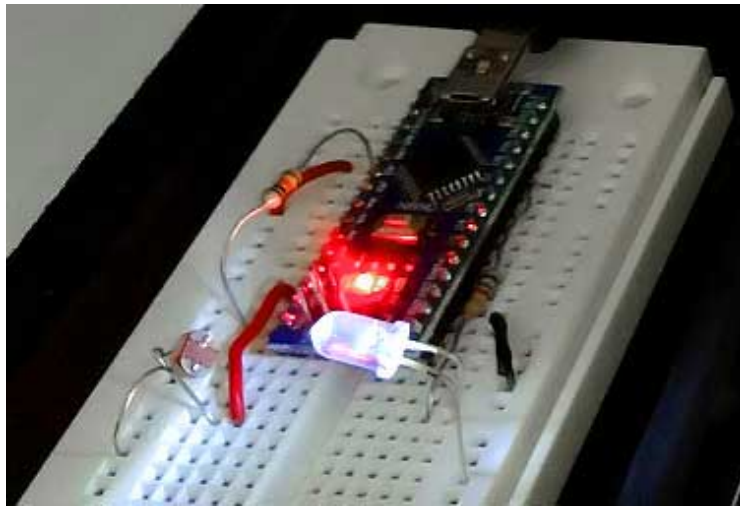
Ai
A
S

\$



Ahora queremos que el LED se encienda y se apague ante una perturbación, por ejemplo cuando ponemos la mano encima del montaje y tapamos algo la luz ambiente.

Lo primero que tenemos que hacer es elegir un consigna apropiada, un punto intermedio entre totalmente iluminado, y total (o bastante oscuro). De lo contrario, el PID no tendrá capacidad para "jugar" la salida.



Por ejemplo, si elegís una consigna inferior a la que obtenéis con el LED apagado, el controlador no va a poder bajar de ahí, por mucho que apague el LED. De forma similar, si la luz que recibe el LDR es muy superior a la del LED encendido, este no va a tener apenas efecto, y el controlador poco podrá hacer. Tiene lógica ¿no?

Así que lo primero que deberíais hacer es medir el valor que registra el LDR con el LED apagado y totalmente iluminado, para determinar una consigna intermedia realmente alcanzable para el controlador.

Ahora es una cuestión de simplemente subir el siguiente código,

Copiar

```
1.  #include <PIDController.hpp>
2.
3.  const long A = 1000; //Resistencia en oscuridad en KΩ
4.  const int B = 15; //Resistencia a la luz (10 Lux) en KΩ
5.  const int Rc = 10; //Resistencia calibracion en KΩ
6.
7.  const int PIN_LDR = A0;
8.  const int PIN_LED = 9;
9.
10. PID::PIDParameters < double > parameters(0.4, 6.0, 0.0001);
11. PID::PIDController < double > pidController(parameters);
12.
13. void setup()
14. {
15.     Serial.begin(115200);
16.     pinMode(PIN_LED, OUTPUT);
17.
18.     pidController.Input = analogRead(PIN_LDR);
19.     pidController.Setpoint = 350;
20.
21.     pidController.TurnOn();
22. }
23.
24. float readLDR()
```

```

25.  {
26.      float mean = 0;
27.      for (auto i = 0; i < 100; i++) {
28.          auto V = (float) analogRead(PIN_LDR);
29.          mean += (V * A * 10) / (B * Rc * (1024 - V));
30.      }
31.      mean /= 100;
32.      return mean;
33.  }
34.
35.  void loop()
36.  {
37.      auto ilum = readLDR();
38.      Serial.println(ilum);
39.
40.      pidController.Input = ilum;
41.      pidController.Update();
42.
43.      analogWrite(PIN_LED, (int) pidController.Output);
44.  }

```

Donde hemos usado la [librería PIDController](#) (que ya que la he hecho, pues la uso). Lógicamente tendréis que ajustar los valores de K y la consigna para vuestro montaje. Pero, a modo orientativo, no estarán muy lejos de los del ejemplo.

En el código, básicamente estamos realizando la lectura de la LDR con una media (simplona) de 100 mediciones. Esta medida la introducimos como entrada del controlador PID, y su salida en la señal PWM a aplicar al LED.

Una vez que tenemos todo listo, encendemos el montaje. El resultado debería ser algo como el siguiente video, donde al tapar el montaje reacciona variando la intensidad del LED para mantener la iluminación que hemos designado.

Controlador PID para iluminación constante co...



Además, como vemos, podemos emplear el [serial plotter](#) del propio IDE de Arduino para graficar la respuesta del sistema y ajustar / verificar su comportamiento.

Este es un ejemplo práctico y real del PID, que requiere un montaje muy sencillo. Esto lo hace muy apropiado para experimentos con los más pequeños, o en el ámbito de la docencia.

En próximas entradas de la sección de teoría de control veremos algo un poquito más complejo. Un regulador de velocidad constante en un motor DC con encoder. ¡Hasta la próxima!

Anuncio:

Tienda de Artículos

Encuentra en Nuestra Tienda
Artículos de Marca a Bajo Precio

Familia Bercomat

 PID

PREVIOUS

CÓMO USAR UN EXTENSOR DE BUS I2C TCA9548A Y ARDUINO

NEXT

ENVIAR Y RECIBIR MENSAJES POR MQTT CON ARDUINO Y LA LIBRERÍA PUBSUBCLIENT

TAMBIÉN TE PUEDE GUSTAR