

Formación en Tecnología, Programación y Robótica

Itinerario Educación
Secundaria
Comunidad Autónoma
de Madrid
Actividad práctica



Índice

Actividad con Arduino.....	3
1_ Nombre, apellidos y correo electrónico de EducaMadrid del participante.....	3
2_ Título de la unidad didáctica.....	3
3_ Curso.....	3
4_ Descripción del proyecto.....	3
5_ Competencias que se trabajan.....	3
6_ Objetivos.....	3
7_ Contenidos.....	3
1° entramos a TINKERCAD.....	4
2° Detector PIR.....	4
3° Conectamos el PIR a arduino.....	4
4° diseño programa uso PIR.....	5
5° LDR. Light Depending Resistor.....	7
6° Potenciómetro.....	8
7° conexión de potenciómetro y LDR a Arduino.....	8
8° diseño del programa para comprobar el comportamiento del circuito anterior.....	9
9° diseño del programa para activar un led cuando la señal de luz esté por debajo de la mitad "valorLDR" <500....	10
10° diseño completo (por fin llegamos al final).....	11
11°Contenido extra.....	13
12°Contesta a estas preguntas.....	14
8_ Criterios de evaluación.....	14
9_ Código del programa Arduino.....	15
10_ Esquema de conexionado.....	16
11_ Fotografía del montaje.....	16

Actividad con Arduino

1 Nombre, apellidos y correo electrónico de EducaMadrid del participante.

Maria Gracia Romero Fernandez
maria.romero35@educa.madrid.org

2 Título de la unidad didáctica

ILUMINACIÓN PASILLO

3 Curso

4º ESO

4 Descripción del proyecto

Se pretende iluminar un pasillo cuando haya poca luz ambiente y además se detecte el paso de una persona

Si hay luz suficiente o no pasa persona, no se debe iluminar el pasillo.

Se realizará de forma virtual usando la plataforma tinkercad

<https://www.tinkercad.com>

5 Competencias que se trabajan

- Competencia digital en tanto que se trabaja con una aplicación informática y se propone el uso de internet para recabar información.
- Competencia "aprender a aprender" en tanto que se propone al alumno que sea el mismo quien desarrolle el proyecto ayudado por el contenido de la unidad didáctica que a continuación se describe.

6 Objetivos

- Distinguir entre salidas y entradas
- Distinguir entre entradas analógicas o digitales
- Nociones básicas de programación
- Reconocer los distintos componentes que se usan

7 Contenidos

Se pretende realizar el proyecto paso a paso, siguiendo las indicaciones que a continuación se describen.

1º entramos a TINKERCAD

Partimos de que al alumno ya ha entrado en la aplicación online TINKERCAD

Si no es así deberá buscar la web, darse de alta e iniciar un proyecto

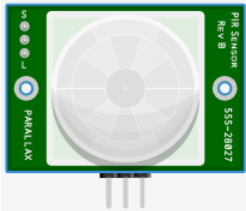


2º Detector PIR



Buscar información en internet sobre detector PIR:

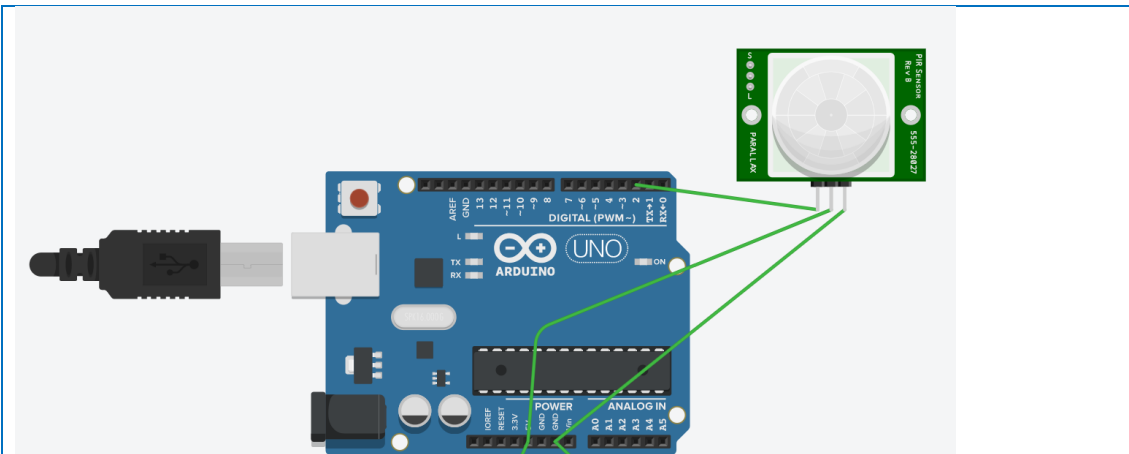
El detector PIR que ofrece TINKERCAD es así



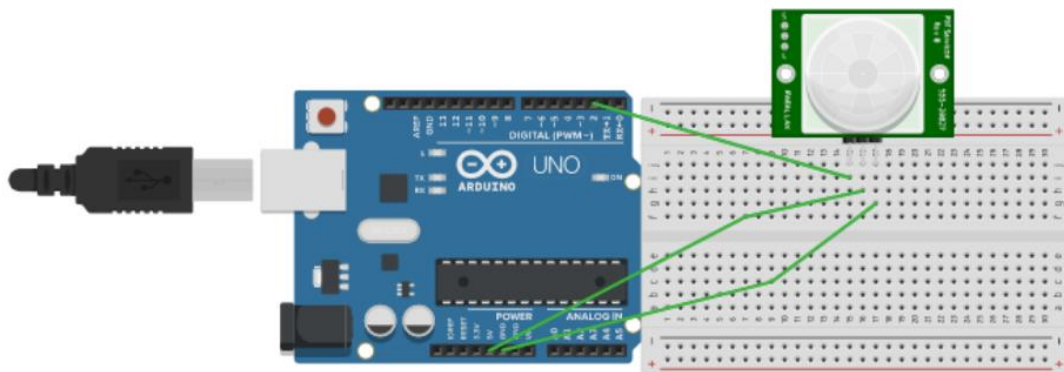
Averiguar qué patilla hay que conectarla a positivo, cual a negativo y cual es la señal

3º Conectamos el PIR a arduino

Decidimos conectar la salida del PIR a la entrada 2 de Arduino



En un diseño real debe usarse una protoboard



4º diseño programa uso PIR

Diseñar un programa que permita averiguar como se comporta el detector PIR.

Para ello visualizaremos la entrada 2 por el monitor serie.

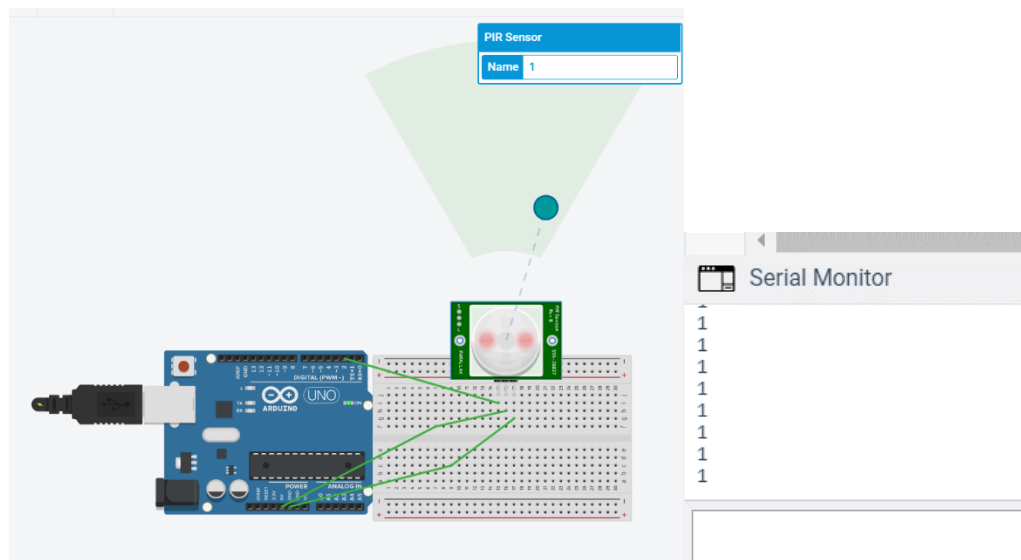
El programa se muestra a continuación incompleto.

El alumno debe introducir en cada zona sombreada la instrucción adecuada

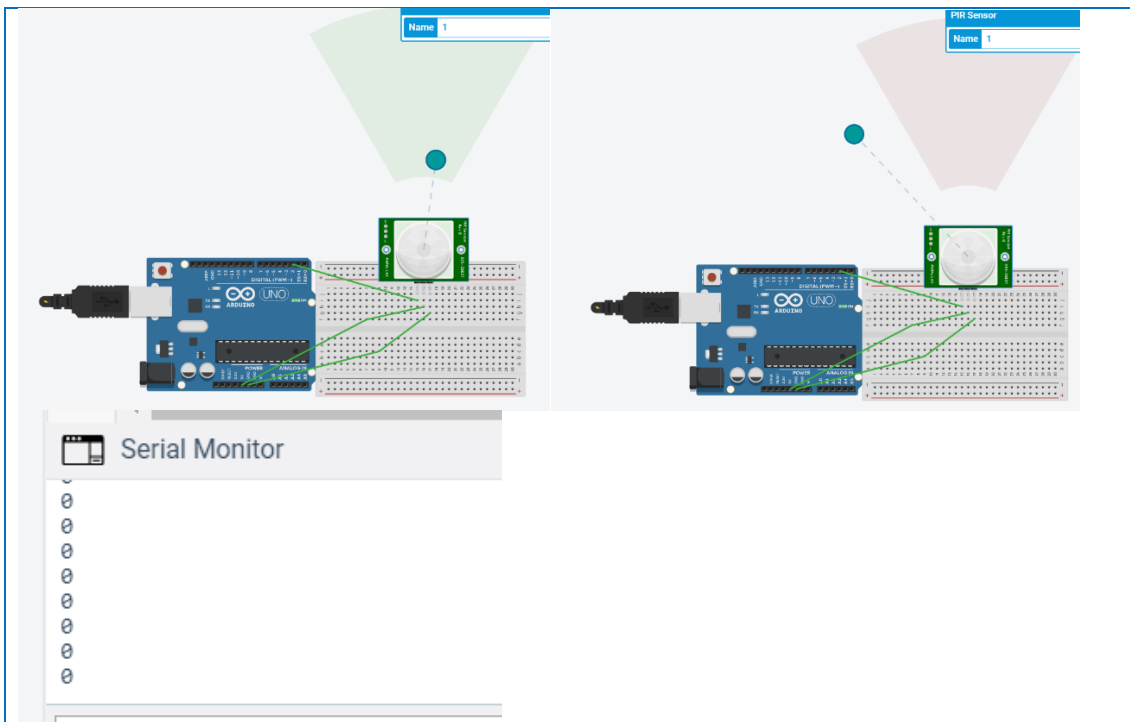
```
1 //Aprendiendo a usar el detector PIR
2 [REDACTED] // a la variable PIR de asigno el número 2.
3 [REDACTED] //Para acordarme que a esta entrada conectaré la salida del detector PIR
4 void setup() {
5
6     Serial.begin(9600);
7
8     [REDACTED] // declaro la entrada PIR como entrada digital
9 }
10
11 void loop() {
12 [REDACTED]; //declaro la variable valorPIR y le asigno lo que
13 [REDACTED] //arduino lea por la entrada digital donde está conectado el detector PIR
14 [REDACTED] //muestro en el monitor serie el valor de la entrada 2
15 [REDACTED] // pongo un retraso de 1 segundo
16
17 }
```

```
int valorPIR = digitalRead(PIR);
Serial.println(valorPIR);
int PIR=2;
pinMode(PIR, INPUT);
delay(1000);
```

Después de realizar el programa deberás ser capaz de comprobar que mientras hay movimiento dentro de la zona de detección el valor de valorPIR es 1

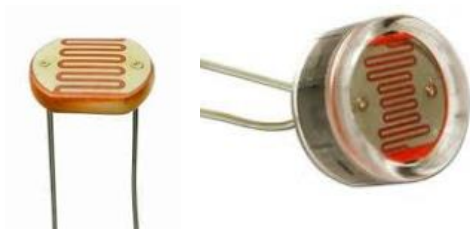


Si no hay movimiento dentro de la zona de detección o estamos fuera de la zona, el valor de valorPIR es 0 (es un detector de movimiento, por lo tanto si el objeto no se mueve el resultado será 0)



5° LDR. Light Depending Resistor

Averigua en internet qué es una LDR



El valor de resistencia es muy alto cuando no hay luz y bajo cuando hay mucha luz

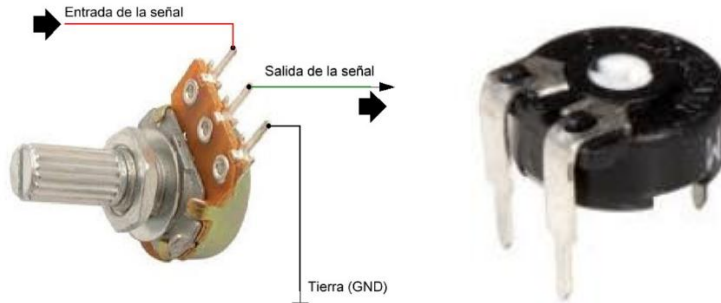
Esta es la LDR que ofrece TINKERCAD



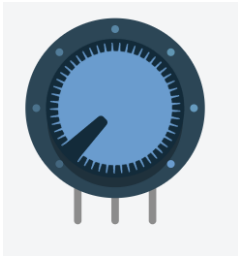
Para poder usar la LDR debemos conectarla con una resistencia o potenciómetro

6º Potenciómetro

Averigua en internet qué es un potenciómetro

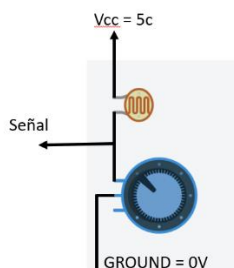


Usaremos este potenciómetro que ofrece TINKERCAD



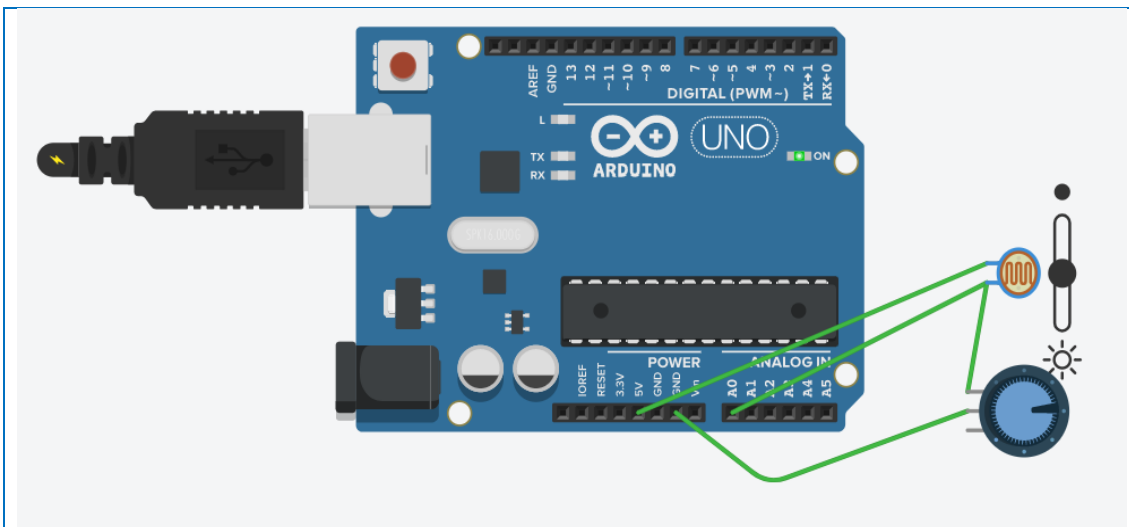
7º conexión de potenciómetro y LDR a Arduino

Con el siguiente montaje podemos tener en "señal" más o menos tensión (entre 0v y 5v) dependiendo del valor de la LDR y del potenciómetro



La "señal" se conectará a la entrada A0 de Arduino

A0 es una entrada analógica que tiene un valor entre 0 y 1024



Al potenciómetro le daremos un valor de 2000Ω

8º diseño del programa para comprobar el comportamiento del circuito anterior

El programa se muestra a continuación incompleto.

El alumno debe introducir en cada zona sombreada la instrucción adecuada

```
1 // Aprendiendo a usar la LDR
2 [ ] // A la variable LDR le asigno A0, para acordarme
3 [ ] // que el valor de la LDR se conectará a la entrada analógica A0
4
5 [ ] // declaro la variable valorLDR y la inicializo a 0
6 void setup()
7 {
8   Serial.begin(9600);
9 }
10
11 void loop()
12 {
13 [ ] // asigno a la variable valorLDR lo que arduino lea
14 [ ] // por la entrada donde está conectada la LDR
15 [ ] // se muestra el valor de la variable valorLDR en la pantalla
16 }
```

`int LDR = A0;`

`Serial.println(valorLDR);`

`int valorLDR = 0;`

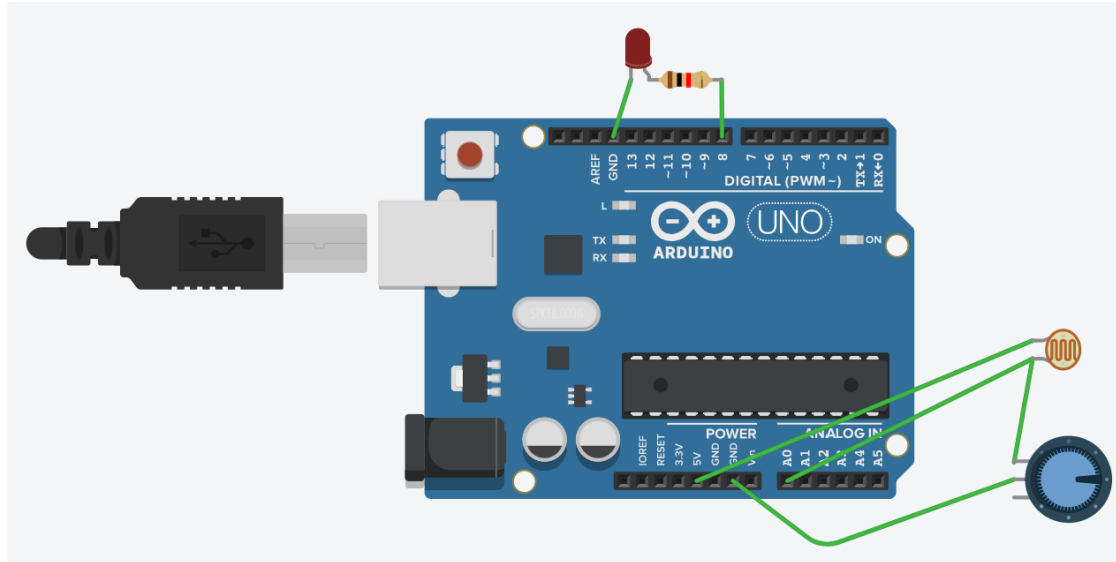
`valorLDR = analogRead(LDR);`

Al ejecutar el programa se observa en el monitor serie los distintos valores que toma "valorLDR"

Situando el potenciómetro a la mitad y la luz de la LDR a la mitad el valor debe ser aproximadamente 500

9º diseño del programa para activar un led cuando la señal de luz esté por debajo de la mitad "valorLDR" <500

Se decide situar el led a la salida 8



El programa se muestra a continuación incompleto.

El alumno debe introducir en cada zona sombreada la instrucción adecuada

```
1 // Aprendiendo a usar la LDR
2 int LDR = A0; // A la variable LDR le asigno A0, para acordarme
3 // que el valor de la LDR se conectará a la entrada analógica A0
4
5 int valorLDR = 0; // declaro la variable valorLDR y la inicializo a 0
6 // A la variable led le asigno 8, para acordarme
7 // que a la salida 8 le conectaré el led
8 void setup()
9 {
10 pinMode(led, OUTPUT); // declaro el pin donde está conectado el led como salida
11 Serial.begin(9600);
12 }
13
14 void loop()
15 {
16 valorLDR = analogRead(LDR); // asigno a la variable valorLDR lo que arduino lee
17 // por la entrada donde está conectada la LDR
18 Serial.println(valorLDR); // se muestra el valor de la variable valorLDR en la pantalla
19 // si valorLDR es menor de 500 activo la salida donde está el led (8)
20
21 // de lo contrario desactivo la salida donde está el led (8)
22
23 }
```

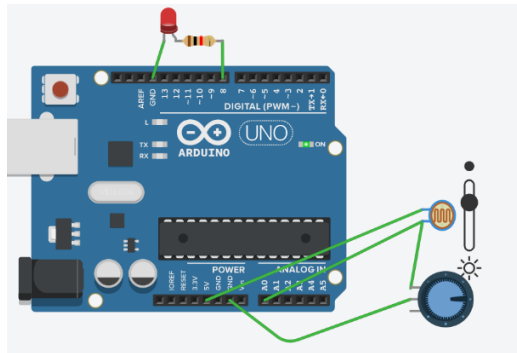
```
if (valorLDR < 500)
{ digitalWrite(led, HIGH); }
```

```
int led = 8;
```

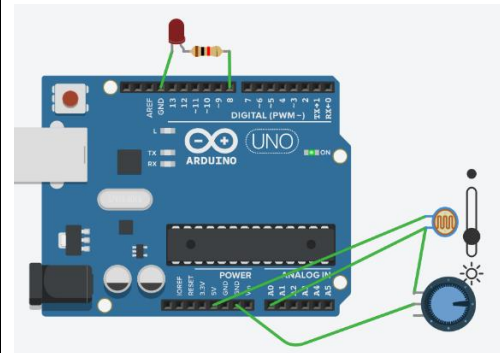
```
else
{ digitalWrite(led, LOW); }
```

Se ejecuta y se comprueba que cuando hay poca luz, el led se enciende

A la LDR le llega poca luz
y el Led se enciende



A la LDR le llega mucha luz
y el Led se apaga



Se puede ajustar el valor del potenciómetro para que el paso de encendido a apagado del Led se encuentre a luminosidades distintas de la LDR

El programa completo es este

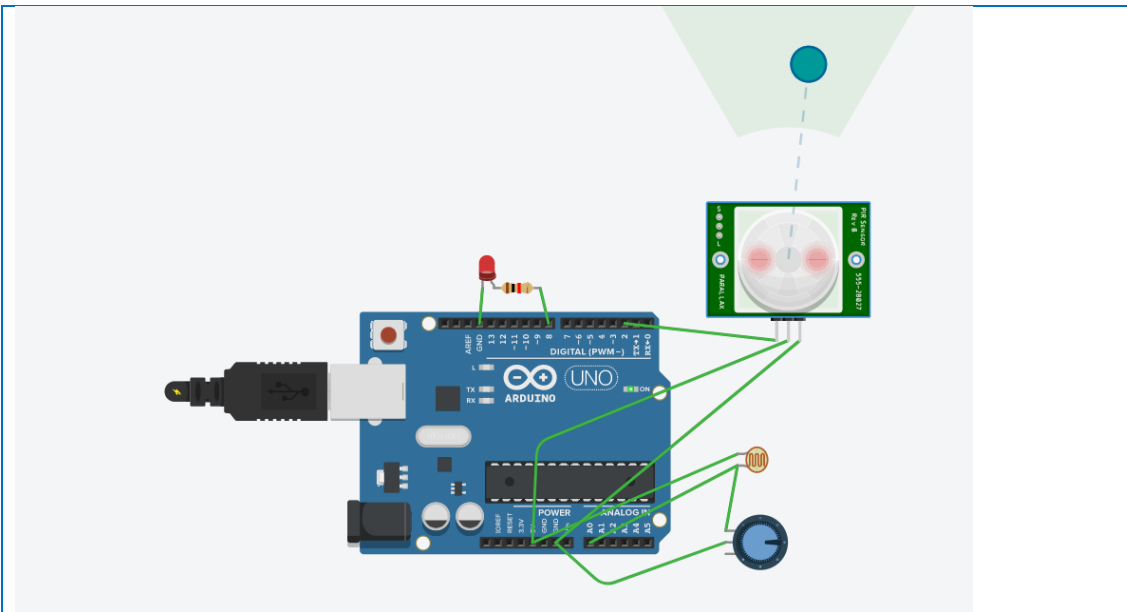
```

1 // programa completo
2 int LDR = A0; // A la variable LDR le asigno A0, para acordarme
3 // que el valor de la LDR se conectará a la entrada analógica A0
4
5 int valorLDR = 0; // declaro la variable valorLDR y la inicializo a 0
6 int led = 8; // A la variable led le asigno 8, para acordarme
7 // que a la salida 8 le conectaré el led
8 int PIR = 2; // A la variable PIR le asigno 2, aquí conectaré la entrada del PIR
9
10 void setup()
11 {
12   pinMode(led, OUTPUT); // declaro el pin donde está conectado el led como salida
13   pinMode(PIR, INPUT); // declaro el pin donde está conectado el PIR como entrada
14   Serial.begin(9600);
15 }
16
17 void loop()
18 {
19   valorLDR = analogRead(LDR); // asigno a la variable valorLDR lo que arduino lea
20   // por la entrada donde está conectada la LDR
21
22   int valorPIR = digitalRead(PIR);
23   Serial.println(valorLDR); // se muestra el valor de la variable valorLDR en la pantalla
24   Serial.println(valorPIR); // se muestra el valor de la variable valorPIR en la pantalla
25   if ((valorLDR < 500) && (valorPIR==1)) // si valorLDR es menor de 500 y valor PIR = 1
26   { digitalWrite(led, HIGH); } // activo la salida donde está el led (8)
27   else // de lo contrario desactivo la salida donde está el led (8)
28   { digitalWrite(led, LOW); }
29 }

```

10° diseño completo (por fin llegamos al final)

Ahora que ya entendemos como se comporta cada componente por separado debemos juntarlos todos para conseguir que se active el led cuando la LDR detecte poca luz y el PIR detecte movimiento (el paso de una persona)



El programa es este

Averigua qué partes del programa son nuevas.

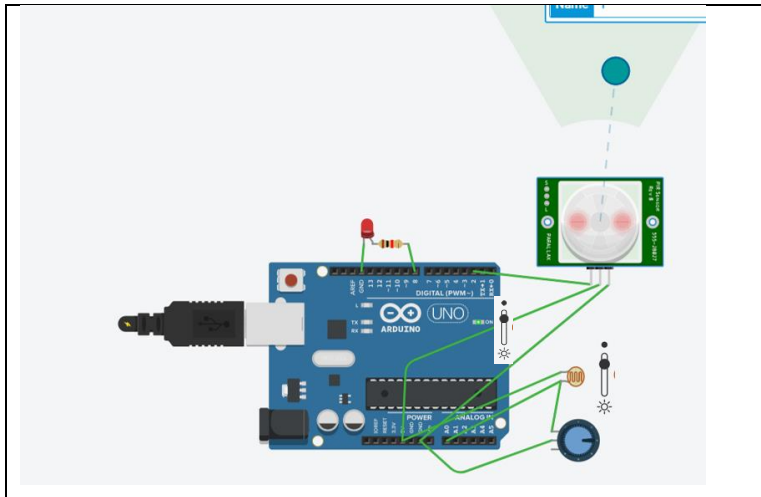
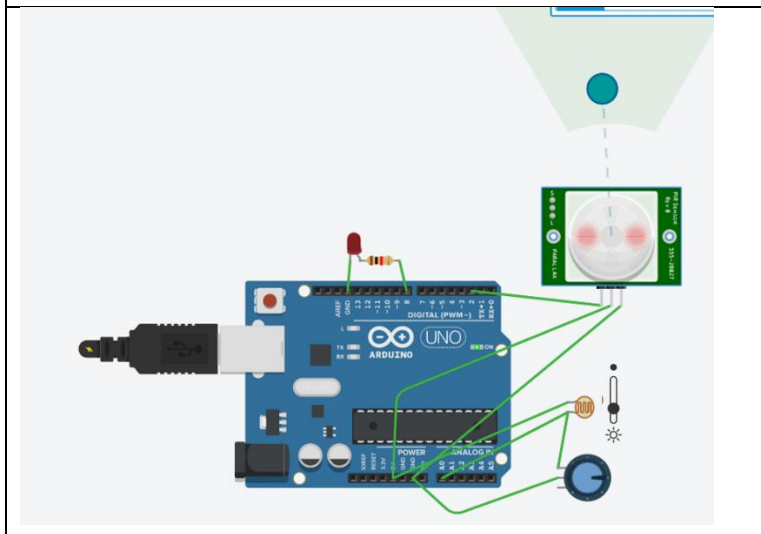
¿Cuál crees que es el “corazón” del programa? ¿La parte más significativa?

```
1 // programa completo
2 int LDR = A0; // A la variable LDR le asigno A0, para acordarme
3 // que el valor de la LDR se conectará a la entrada analógica A0
4
5 int valorLDR = 0; // declaro la variable valorLDR y la inicializo a 0
6 int led = 8; // A la variable led le asigno 8, para acordarme
7 // que a la salida 8 le conectaré el led
8 int PIR = 2; // A la variable PIR le asigno 2, aquí conectaré la entrada del PIR
9
10 void setup()
11 {
12   pinMode(led, OUTPUT); // declaro el pin donde está conectado el led como salida
13   pinMode(PIR, INPUT); // declaro el pin donde está conectado el PIR como entrada
14   Serial.begin(9600);
15 }
16
17 void loop()
18 {
19   valorLDR = analogRead(LDR); // asigno a la variable valorLDR lo que arduino lea
20 // por la entrada donde está conectada la LDR
21
22   int valorPIR = digitalRead(PIR);
23   Serial.println(valorLDR); // se muestra el valor de la variable valorLDR en la pantalla
24   Serial.println(valorPIR); // se muestra el valor de la variable valorPIR en la pantalla
25   if ((valorLDR < 500) &&( valorPIR==1) ) // si valorLDR es menor de 500 y valor PIR = 1
26   { digitalWrite(led, HIGH); } // activo la salida donde está el led (8)
27   else // de lo contrario desactivo la salida donde está el led (8)
28   { digitalWrite(led, LOW); }
29 }
```

Aquí se muestra la parte más importante del programa

```
   if ((valorLDR < 500) &&( valorPIR==1) ) ,
   { digitalWrite(led, HIGH); }
   else
   { digitalWrite(led, LOW); }
}
```

Prueba el programa y comprobaras que:

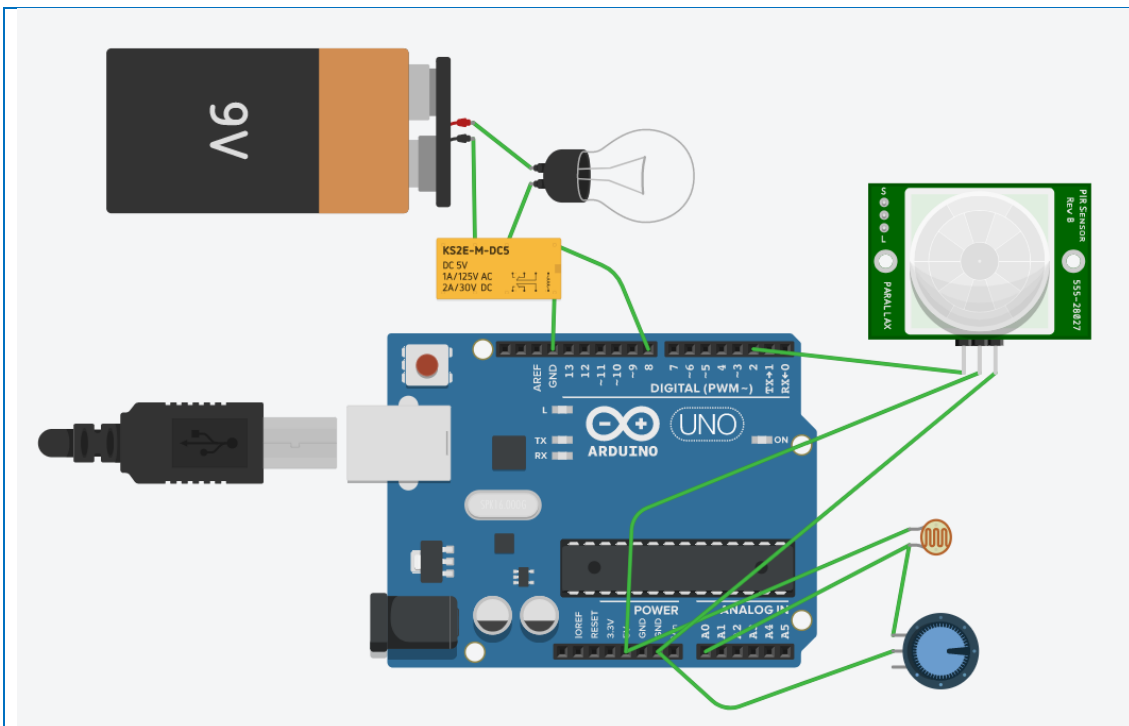
	<p>Se activa el led cuando hay movimiento y poca luz</p> <p>El monitor muestra 1 287</p>
	<p>No se activa el led cuando hay mucha luz, aunque haya movimiento</p> <p>El monitor muestra 1 578</p>

Tampoco se activará el led si hay poca luz y no hay movimiento

11° Contenido extra

Si lo que se pretende es iluminar un pasillo, deberemos usar una lámpara que ilumine más que un led. Pero Arduino no proporciona suficiente intensidad o tensión para dicha lámpara, por lo tanto debemos actuar así:

Arduino activará un relé y el contacto del relé será quien active la lámpara



12° Contesta a estas preguntas

Al lado de cada componente indica si es una salida o entrada a Arduino

PIR

Led

LDR y potenciómetro

Relé

Al lado de cada componente indica si es analógico o digital

PIR

Led

LDR y potenciómetro

Relé

8 Criterios de evaluación

- El alumno identifica adecuadamente entradas de salidas
- El alumno identifica adecuadamente la diferencia entre analógico y digital

- Al ser una unidad didáctica con muchos pasos guiados, el profesor irá comprobando el grado de autonomía de cada alumno y si es capaz de llegar al final

9 Código del programa Arduino

```
// programa completo
int LDR = A0; // A la variable LDR le asigno A0, para acordarme
// que el valor de la LDR se conectará a la entrada analógica A0

int valorLDR = 0; // declaro la variable valorLDR y la inicializo a 0
int led = 8; // A la variable led le asigno 8, para acordarme que a la salida 8 le
conectaré el led
int PIR = 2; // A la variable PIR le asigno 2, aquí conectaré la entrada del
PIR

void setup()
{
  pinMode(led, OUTPUT); // declaro el pin donde está conectado el led como salida
  pinMode(PIR, INPUT); // declaro el pin donde está conectado el PIR como entrada
  Serial.begin(9600);
}

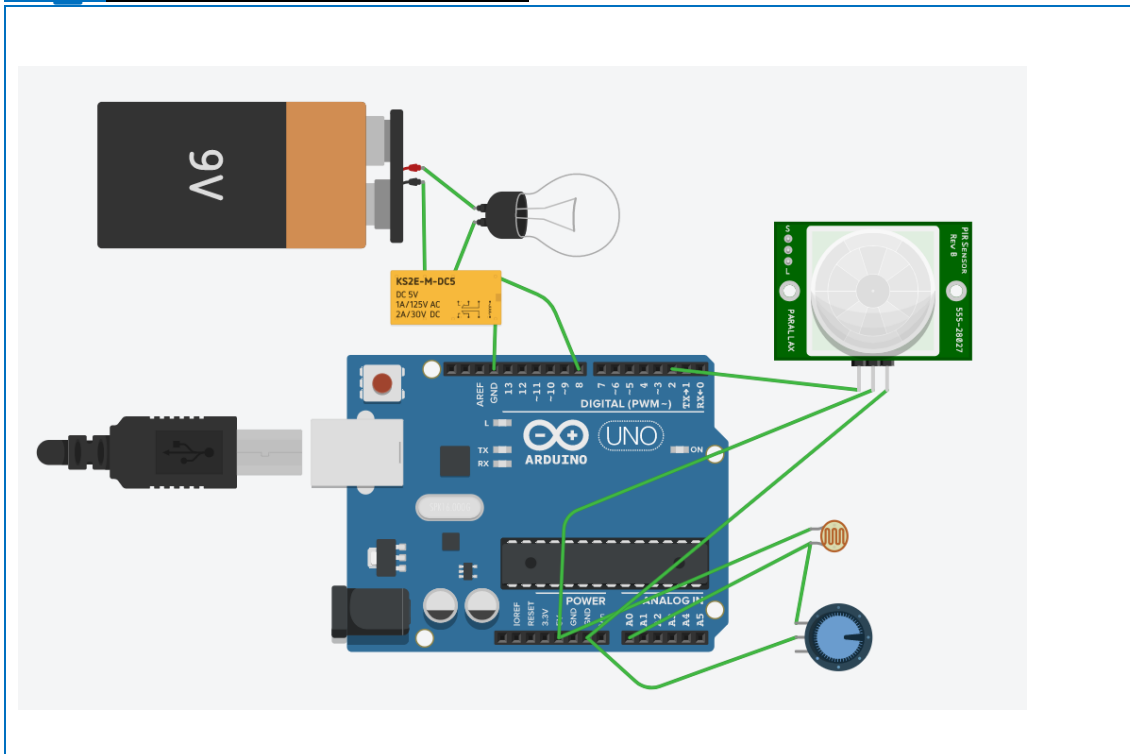
void loop()
{
  valorLDR = analogRead(LDR); // asigno a la variable valorLDR lo que arduino lea por la
  entrada donde está conectada la LDR

  int valorPIR = digitalRead(PIR);

  Serial.println(valorLDR); // se muestra el valor de la variable valorLDR en la
  pantalla
  Serial.println(valorPIR); // se muestra el valor de la variable valorPIR en la
  pantalla

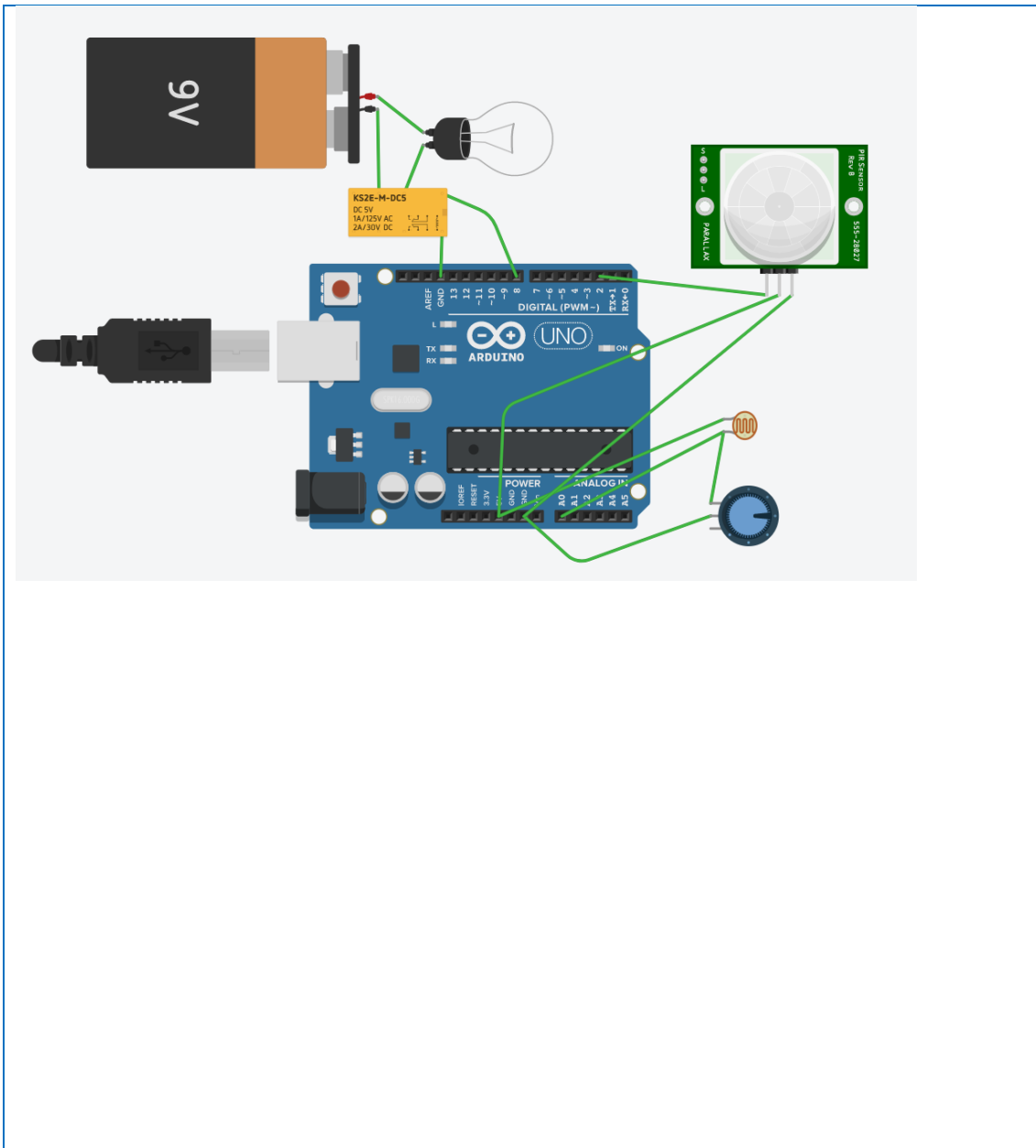
  if ((valorLDR < 500) &&( valorPIR==1) ) // si valorLDR es menor de 500 y valor PIR = 1
  { digitalWrite(led, HIGH); } // activo la salida donde está el led (8)
  else // de lo contrario desactivo la salida donde está el led
  (8)
  { digitalWrite(led, LOW); }
}
```

10 Esquema de conexionado



11 Fotografía del montaje

No dispongo de un Arduino físico ni los componentes usados en esta unidad didáctica
Por lo tanto lo he planteado como un proyecto completamente virtual usando la plataforma TINKERCAD



X	<p>Marque con una cruz si acepta que esta unidad didáctica quede recogida bajo una licencia Creative Commons (by - nc - sa) (Reconocimiento, No Comercial, Compartir Igual) para ser compartida por todos los miembros de la comunidad de docentes: http://es.creativecommons.org/blog/licencias/</p>
---	---

A background network diagram consisting of a complex web of light blue lines and circular nodes, resembling a circuit board or a data network, set against a dark teal background.

Telefonica

EDUCACIÓN
DIGITAL

SCOLARTIC_
Creando Código