

BLOQUE 2

Introdución.

Este conxunto de retos, pretenden familiarizarnos cos elementos que vimos o primeiro día (saídas dixitais, programación en visualino, lecturas analóxicas e comunicación serie.

Neste novo bloque de retos repetiremos conceptos vistos no anterior e introduciremos uns novos:

- Compoñentes
 - o Resistencia LDR (uso como sensor de luz).
 - o Sensor de temperatura LM36
 - o Zumbador
- Programación
 - o Sentencias de control, o condicional simple.

Reto guiado 2.1. Lectura dun LDR.

RETO 2.1

Realiza un proxecto no que, usando o USB, realices unha **comunicación en serie** co PC, amosando cada segundo os **valores analóxicos que vai xerar o sensor de luz** ao tapar/iluminar o mesmo.

Resólveo en TINKERCAD e posteriormente na túa placa.

CURSO DE ARDUINO

José Pino Vázquez

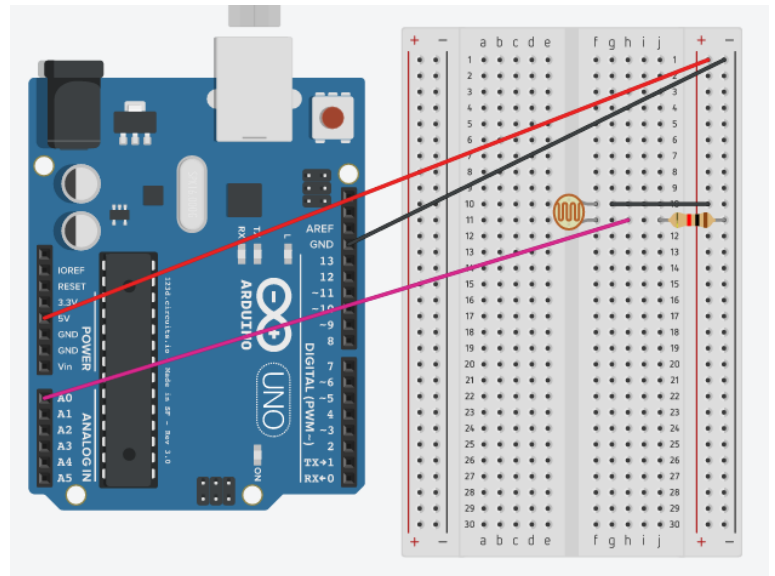


Neste reto pretendemos amosar por pantalla o valor da intensidade da luz recibido por unha resistencia LDR.

O circuito

A resistencia variable LDR utilízase como sensor de luz xa que a súa resistencia variará coa cantidade de luz que esta recibe. Para o noso exemplo montaremos un circuíto con esta resistencia (co LDR) e outra resistencia de 1kΩ.

Dependendo da posición das resistencias, estas devolverannos un valor que incrementará ou diminuirá coa luz. Se queremos obter unha variación que sexa proporcional á intensidade da luz, teremos que conectar o LDR cunha pata a 5V, ler o valor e posteriormente pechar o circuíto coa resistencia a GND. Se queremos obter un valor inverso, tan só teríamos que intercambiar a posición das dúas resistencias.

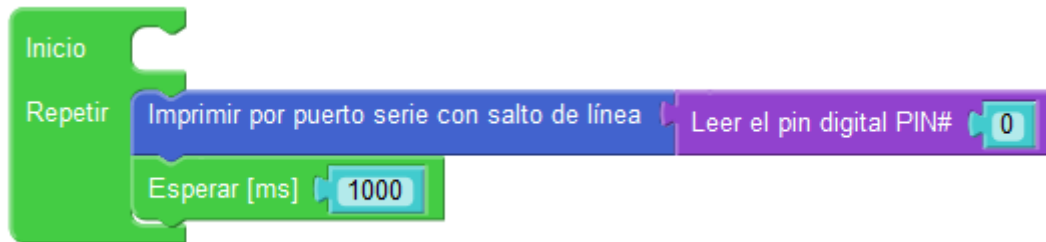


Fíxate que enviaremos un conector dende A0 ata a posición h11, que servirá para ler a intensidade que pasa por ese punto, que será acorde ao valor que devolva a resistencia LDR, polo tanto un valor de “cantidade de luz detectada”

O código

O código que imos a desenvolver é realmente sinxelo, e serviranos para lembrar o que estivemos a facer o outro día co potenciómetro. Se queres podes programalo directamente no IDE, aínda que seguramente non precisas esta guía para facelo...

O código resultante sería o seguinte, pero imos a estudalo...



O bloque de **inicio ... repetir** servirá como ben sabes para comezar a fixar as nosas ordes, no noso caso tan só programamos **repetir** (en realidade ao introducir instrucións en **repetir**, el xa fixa ordes en **inicio** aínda que non o vexamos).

A seguinte orde, indícalle a Arduino que envíe mediante o porto USB a información que se lle fixe á dereita.

Imprimir por puerto serie con salto de línea

No seguinte bloque, lemos datos por unha entrada analóxica, que xustamente se enviarán a orde de arriba, e polo tanto comunicaranse por USB.

Leer el pin analógico PIN#

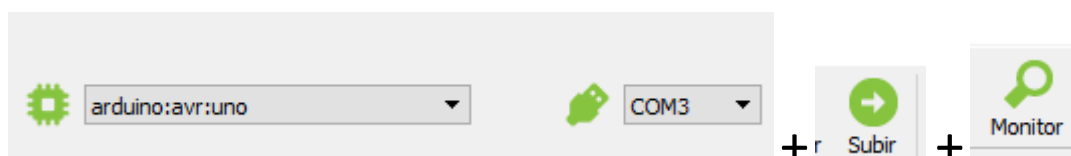
Na seguinte indicamos o número de porto no que lemos o valor:

0

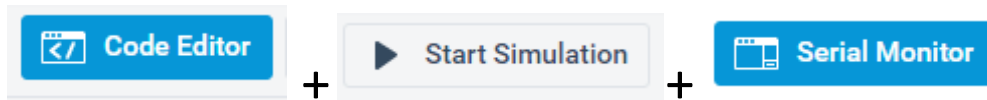
Por último xa amosamos dous bloques conxuntos, que o que farán é indicarlle ao robot que agarde 1 segundo antes de ler o seguinte valor.

Funcionará??...

Agora tan só queda probalo, e podemos facelo de varios xeitos, por un lado podemos enviar o programa directamente á placa dende o visualino..



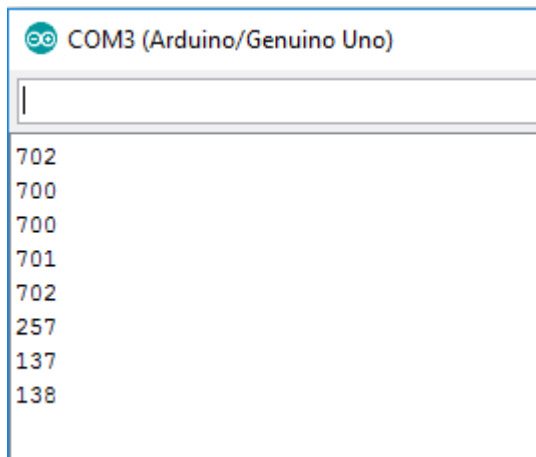
Outra opción e copiar o código que xera o visualino e pegalo no simulador de 123Dcircuits (co circuíto montado, loxicamente)



A derradeira será a de pegar o código de visualino no IDE de Arduino e enviar o programa á placa... unha vez subido este, abrimos o monitor para ler os valores.



Para comprobar o funcionamento ilumina ou tapa o LDR, no simulador se fas clic sobre o LDR podes simular a presenza ou non de luz.



Reto guiado 2.2. Acender un lede dependendo da cantidade de luz

RETO 2.2

CURSO DE ARDUINO

José Pino Vázquez

Realiza un proxecto no que usando unha resistencia LDR sexas capaz de acender un lede se detectamos pouca luz, e que ese lede se apague se detectamos moita. Aproveita o circuito anterior, e conecta o lede á saída dixital 3.

Resólveo en TINKERCAD e posteriormente na túa placa.



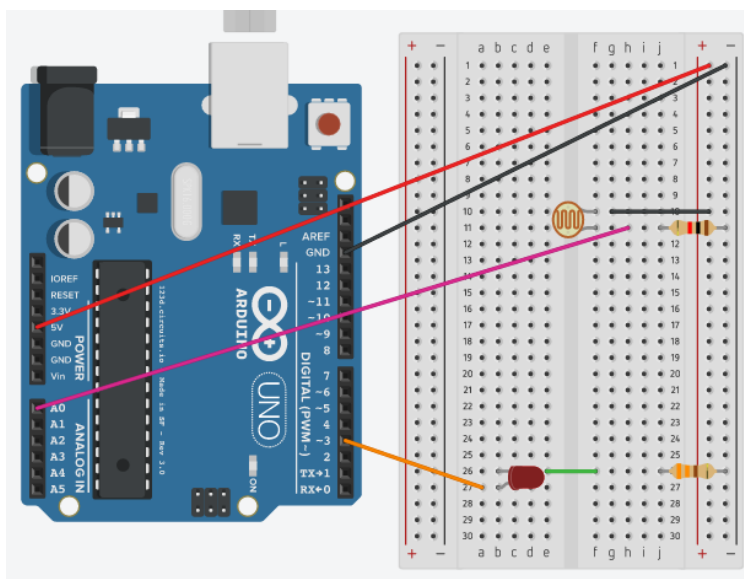
Imos a complica un pouco o reto anterior, e para iso, acenderemos un lede cando a cantidade de luz detectada sexa inferior a 300, quedando apagado cando esta sexa superior a este valor.

O circuito

Aproveitaremos o circuío anterior, e cos coñecementos do primeiro día, montaremos un lede na placa de conexións. Un exemplo podería ser o que se achega.

Como se pode apreciar, o lede vermello está conectado a GND a través da resistencia de 330Ω

O seu ánodo está conectado á saída 3 de Arduino, que por suposto controlaremos nos, o que nos permitirá poñela en Alto ou Baixo dependendo da nosa programación.

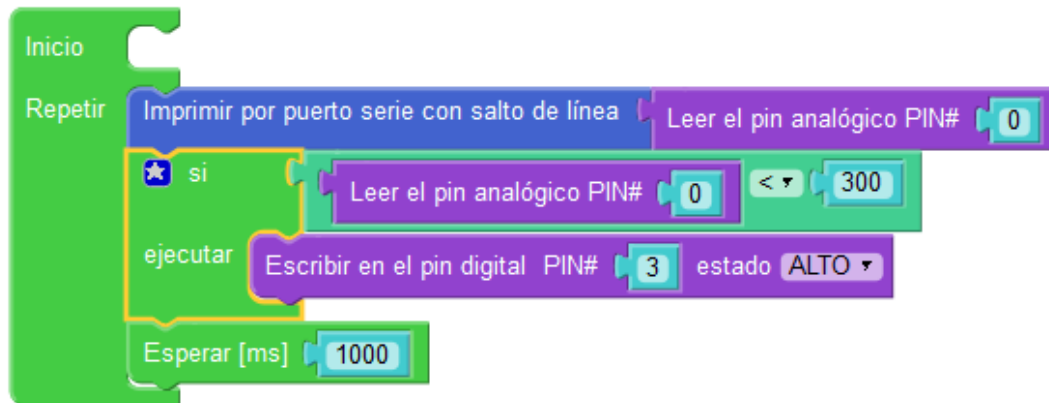


O código

Ao igual que o circuío, imos a deixar o código que había e amplialo (así tamén podemos comprobar co monitor serie se o lede está a acenderse cando llo indicamos).

Neste caso non só imos a escribir por serie o valor do pin de entrada A0, senón que dependendo dese valor, poremos en alto ou non a saída dixital 3 (e dicir, acenderemos ou non o lede).

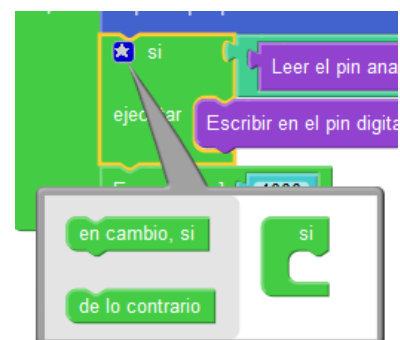
Unha primeira aproximación a solución sería:



Non é unha solución válida, pero serviranos para estudar un pouco o condicional si. Este condicional avalía a condición que ten á dereita (se o valor do pin analóxico A0<300), se é correcto, executarase a sección que hai en executar, neste caso poñer o pin 3 dixital en alto, o que é o mesmo, e vendo o circuío, acender o lede vermello).

Este código ten un erro, xa que unha vez o lede se acenda, non se apaga, e iso é así xa que en ningún momento se lle está a dicir que o lede se apague se a luz detectada é maior ou igual a 300 (ímos a ser pouco escrupulosos e tan só mirar o maior).

Pódese facer de varias maneiras, pero nos ímola facer da máis óptima, neste caso sería a de engadir unha sentenza de “se non se cumpre, fai isto outro”, para iso prememos sobre a estreliña que hai antes do si, e seleccionaremos a opción “de lo contrario” e arrastrámo-lo cara a dereita de xeito que quede como a imaxe inferior



Unha vez feito, podemos premer sobre a estreliña... e tan só nos queda indicarlle ao programa que debe facer se non se cumpre a condición, no noso caso poñer o pin dixital 3 a baixo...

Co cal o código queda algo como así:



Funcionará??...

Pois claro que si !!

Reto 2.3. Monitorización do estado

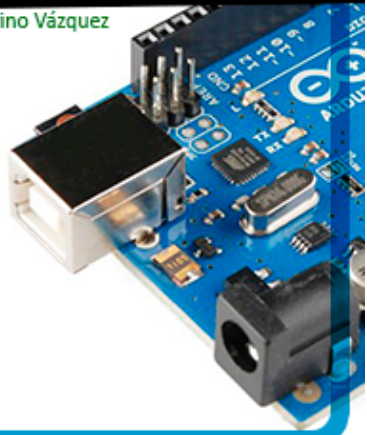
RETO 2.3

Amplia o reto anterior para que se comunique ao PC por comunicación serie o estado do lede. Indicando polo tanto cando o lede está aceso e cando está apagado.

Resólveo en TINKERCAD e posteriormente na túa placa.

CURSO DE ARDUINO

José Pino Vázquez



Reto guiado 2.4. Temperatura

Quizais se te estean a ocorrer milleiros de exercicios e posibilidades ao combinar diversos sensores e actuadores, no noso caso e de xeito sinxelo imos a usar dous máis, o sensor de ultrasons e un buzzer.

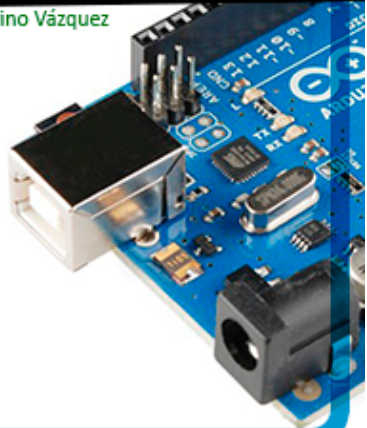
RETO 2.4

Seguindo coas entradas analóxicas, realizaremos un reto que consistirá en que se a temperatura é superior a 23º (aproxíma este valor á temperatura que teñamos na aula), sone unha alarma e se acenda un lede vermello. No caso contrario o lede permanecerá apagado e non sonará a alarma.

Resólveo en TINKERCAD e posteriormente na túa placa.

CURSO DE ARDUINO

José Pino Vázquez

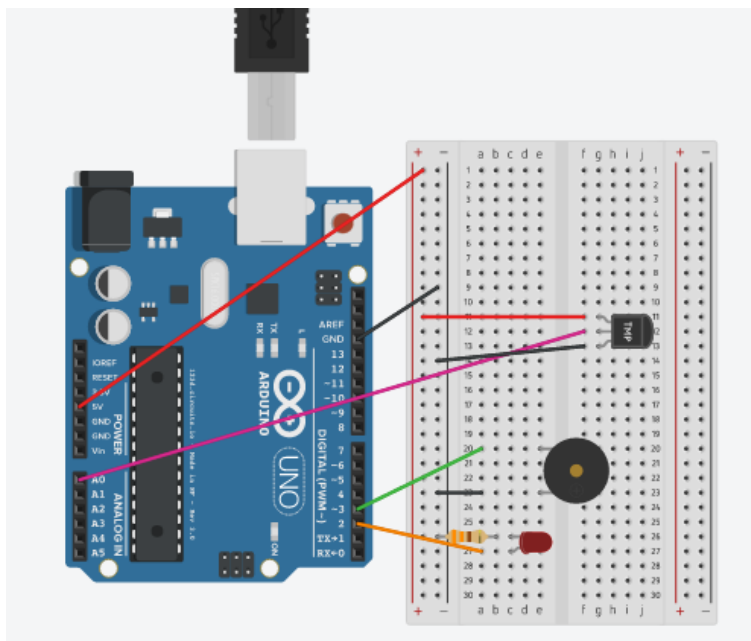


Aínda que pareza sinxelo.. imos a ter que repensar todo...

O circuío

A montaxe do circuío é bastante sinxela, basicamente é a montaxe dun lede controlando a súa voltaxe positiva mediante a saída dixital número 2, e coa súa resistencia de 330Ω, un sensor de temperatura coas conexións dos extremos a GND e a 5V e a central a unha entrada analóxica (A0 no noso caso), e por último, un zumbador conectado por unha pata a negativo e a outra controlada dende o pin dixital 3...

De xeito sinxelo seguro que xa saberás amosar os valores de “temperatura” que nos devolve o TMP36 enviándoos por serie ao ordenador.



O código Rev 1

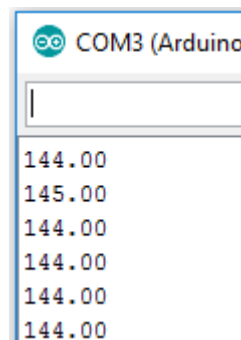
Como primeira aproximación podíamos realizar o seguinte código, en diferencia cos anteriores, tan só estamos a engadir o zumbador que emite un tono (DO) durante 200 milisegundos.

Deixamos o delay (esperar) de 1000 ms co fin de que o zumbador emita o son a saltos e para que nos de tempo a ler o valor recibido polo porto serie.



Funcionará??...

Non, e se miramos o monitor serie seguro que nos damos de conta que hai algo mal...



Pero por que estamos a 144 grados?? , isto e así porque o sensor de temperatura non nos devolve a temperatura, senón e seguindo as instrucións do fabricante (podes ver o datasheet nas ligazóns do día 1), o sensor devolve unha saída de tensión de 10 mV (mili voltios) por cada grao de temperatura, ademais lembrar que Arduino nas súas entradas analóxicas pode ler valores cun máximo de 1024 para 5V e 0 para 0V. Polo tanto cunha simple regra de 3, podemos calcular o valor real...o valor en voltios dos 144 que recibimos serán...

$$144 * 5 / 1024$$

E decir, **analogRead(0)*5/1024**, vale, temos os voltios, e agora por cada 10mv e un grado, non? pois enton multiplicamos por 100 (de voltios a 10 milivoltios) e xa temos a temperatura.... 70 grafiños.... bueno, si, debe haber algo mal....

O TMP36, a diferencia do tmp35, permite ler valores negativos de grados, de tal xeito que mide dende os -50º, polo tanto teremos que restarlle 50 ao resultado final da operación que quedaría algo como esto...

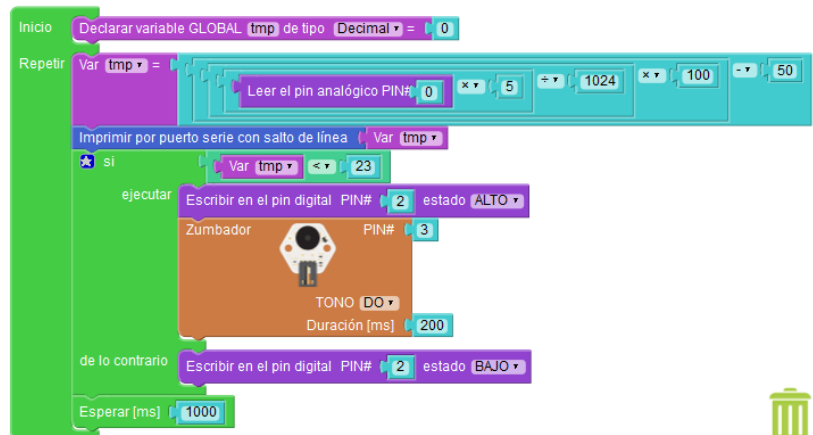
$$\text{analogRead}(0) * 5 / 1024 - 50$$

Polo tanto teremos que volver ó código para inserir esta operación...

O código Rev 2

Agora xa estamos de volta de como vai o sensor de temperatura, polo tanto tan só temos que meter as operacións.. imos a usar unha variable co fin de non ter que calcular as cousas dúas veces, imaxina que unha variable e como unha “caixiña cun nome” onde gardamos un valor. A variable definímolade tipo “decimal” de xeito que utilice dous números decimais para non perder ningún valor durante as operacións.

O código despois de aplicar a mellora anterior, quedaría así... (igual te plantexas seguir usando visualino cando vexas que semellante lío de bloques fan un código de operación tan pequeno como comentabamos antes...)



O código en texto sería o seguinte, que creo que xa o preferirás e que en realidade teremos que traballar neste caso con el...

```
/** Global variables */
float tmp;
/** Function declaration */
void setup()
{
  tmp=0;
  pinMode(0,INPUT);
  Serial.begin(9600);
  pinMode(2,OUTPUT);
}

void loop()
{
  tmp=analogRead(0) * 5 / 1024 * 100 - 50;
  Serial.println(tmp);
  if (tmp < 23) {
    digitalWrite(2,HIGH);
    tone(3,261,200);
    delay(200);
  }else {
    digitalWrite(2,LOW);
  }
  delay(1000);
}
/** Function definition */
```

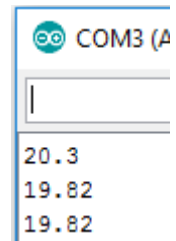
Se o executamos seguimos a ver que non funciona, e iso é debido a que ao multiplicar `analogRead(0) * 5`, ese 5 convirte o número a enteiro sen decimais, provocando que ao dividir entre 1024 o resultado de 0 en lugar de cero con algo e polo tanto perdendo os decimais.

A solución será que no código (ao final o código permítenos maior control sobre o programa), poñamos ese 5 con dous decimais... e así si que funciona.

```
tmp=analogRead(0) * 5.00 / 1024 * 100 - 50;
```

Funcionará??...

Agora proba o programa e segundo os valores que devolva o valor serie, tenta axustar os grados (en lugar de 23, pon un valor próximo ao que temos neste intre na aula), e proba a darlle calor o sensor a ver se se acenden ou apagan as alarmas. Podes aproveitar a saída do dissipador do portátil para quentalo, ou se ves que non o consegues, fai a comprobación ao revés e que se acendan as alarmas cando detecte unha temperatura superior á estimada.



```
COM3 (A  
|  
20.3  
19.82  
19.82
```

Podes poñer medio grado en lugar de saltar de grado en grado, pero precisarás programar en código... atreveste?

Reto 2.5. Saídas PWM

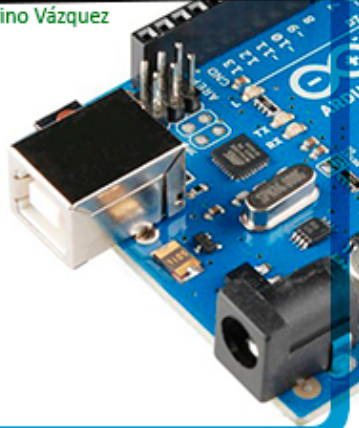
RETO 2.5

Construe un circuíto que nos permita que un lede se ilumine cunha intensidade proporcional ao valor dun potenciómetro. Podes aproveitar o circuíto do reto 1.7, usando unha saída PWM para o lede.

Resólveo en TINKERCAD e posteriormente na túa placa.

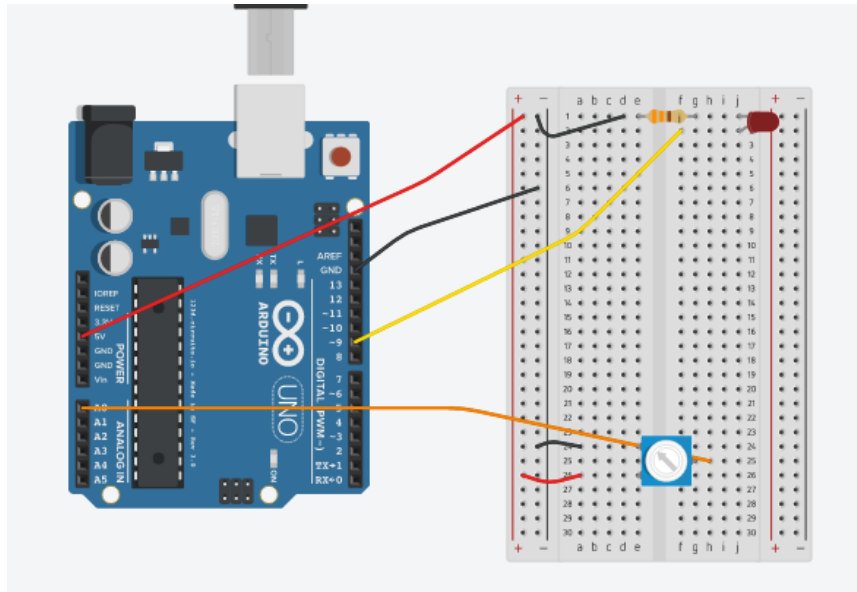
CURSO DE ARDUINO

José Pino Vázquez



Dun xeito rápido imos a construír un reto para probar as saídas PWM, este reto consiste en que un lede se ilumine cunha intensidade proporcional ao valor dun potenciómetro. O circuíto sería o seguinte:

Date de conta que o único especial do circuíto e que o pin que controla a iluminación do lede e un PWD, podes velo pola pequena marca que hai dentro do número ~9.



O código

O código é realmente sinxelo, neste caso temos que transportar o valor que lemos en `analogRead(0)`, cunha escala de 0 a 1023, a un valor que nos permita en `analogWrite(9)`, que irá de 0 a 255.

O resto é ler a variable e amosar o valor...

O código fonte é tan ou igual máis sinxelo que o código de bloques, quedaría algo como así:



```

/** Global variables */
/** Function declaration */

void setup()
{
    pinMode(0, INPUT);
    pinMode(9, OUTPUT);
}

void loop()
{
    int intensidad=analogRead(0);
    analogWrite(9,map(intensidad,0,1024,0,255));
}

/** Function definition */

```