

BLOQUE 3

Introdución.

Para u derradeiro día, deixamos un par de cousiñas, 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:

- Pulsadores
- LCD
- Ultrasons

Reto guiado 3.1. O LCD

RETO 3.1

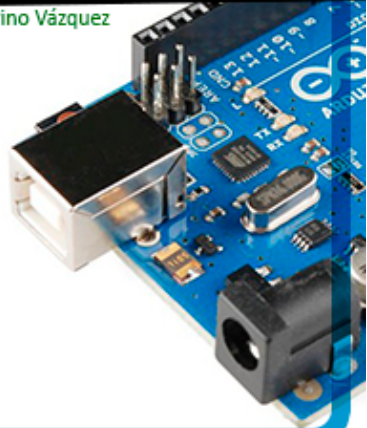
Realiza a montaxe dun circuito que emita a seguinte mensaxe a través dun monitor LCD:

"Curso Arduino CFR Lugo".

Resólveo en THINKERCAD ou na túa placa.

CURSO DE ARDUINO

José Pino Vázquez



O circuito I

O difícil deste reto é a montaxe do circuíto, neste caso o LCD precisará de moitas conexións para que consigamos que amose un determinado contido. E por iso que o circuíto o amosaremos a pasos...

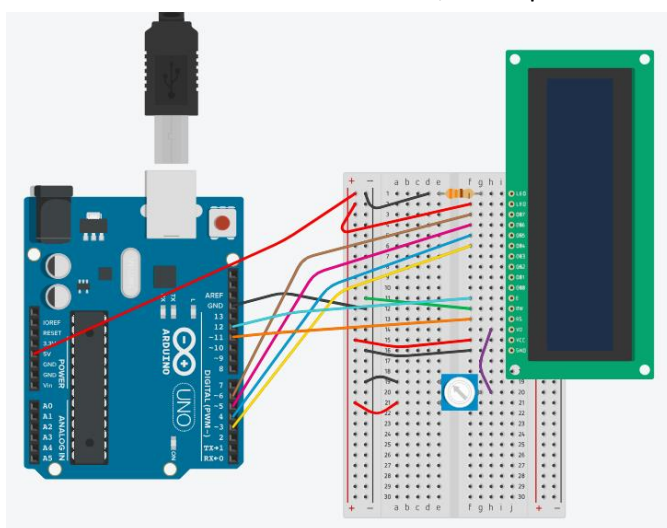
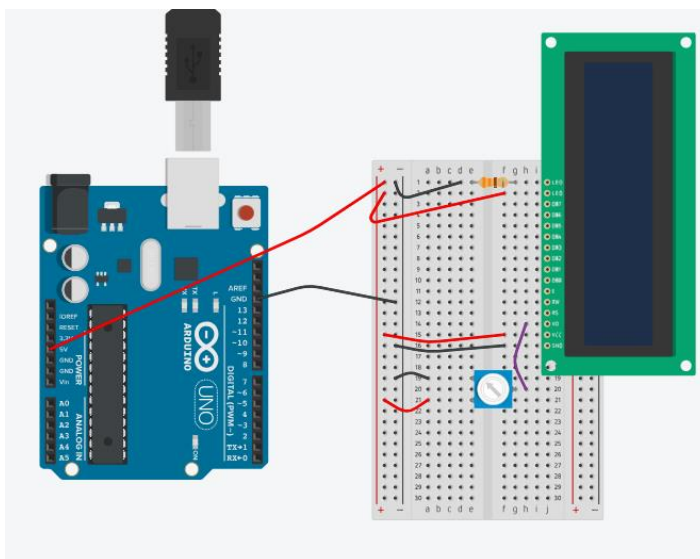
O primeiro paso será alimentar o LCD mediante as conexións a GND e 5v, date de conta que leva dúas alimentacións, unha para a luz de fondo e outra para os cadros dos números. Unha das liñas GND precisará unha resistencia de 330Ω (o ideal sería algo menor, pero apañaremos...), de tódolos xeitos podes atopar manuais nas que non a usan, brillando máis o LCD pero acortando a súa vida.

Por outro lado montaremos un potenciómetro que tamén o conectaremos a GND e 5V. De Momento non conectaremos a saída do valor de lectura do potenciómetro a ningures... debería quedar algo coma a imaxe do lado.

Podes probar a montaxe conectado a placa Arduino ó ordenador ou ben a unha pila, se xiras o potenciómetro verás que este fai aparecer ou non os cadros onde irían as letras, como podes ver o potenciómetro controlará o contraste.

O circuíto II

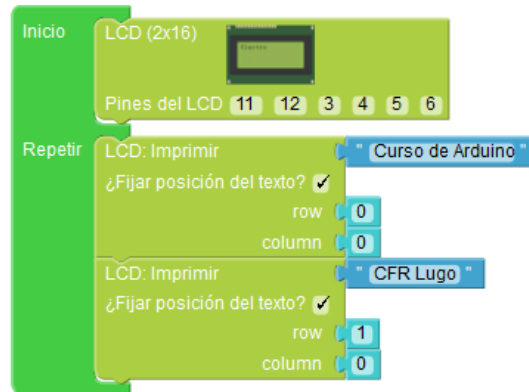
Xa temos o sistema medio montado, tan só faltaría enviarlle a información a amosar o LCD, para iso imos tirar unhas conexións dende a nosa placa contra os pins da pantalla, as conexións que faltan serían as seguintes.



LCD	RS	RW	E	D4	D5	D6	D7
Arduino	11	GND	12	3	4	5	6

O código

As conexións de pines que elaboramos están acorde ao bloque que usa visualino, en cambio se usamos o ide de arduino, os exemplos que trae non coinciden os pines cos que indica o visualino, polo que terías que modificar os pines do LCD:



O código correspondente tamén será sinxelo de seguir:

```
#include <Wire.h>
#include <LiquidCrystal.h>

/** Global variables */
LiquidCrystal lcd(11, 12, 3, 4, 5, 6);

/** Function declaration */

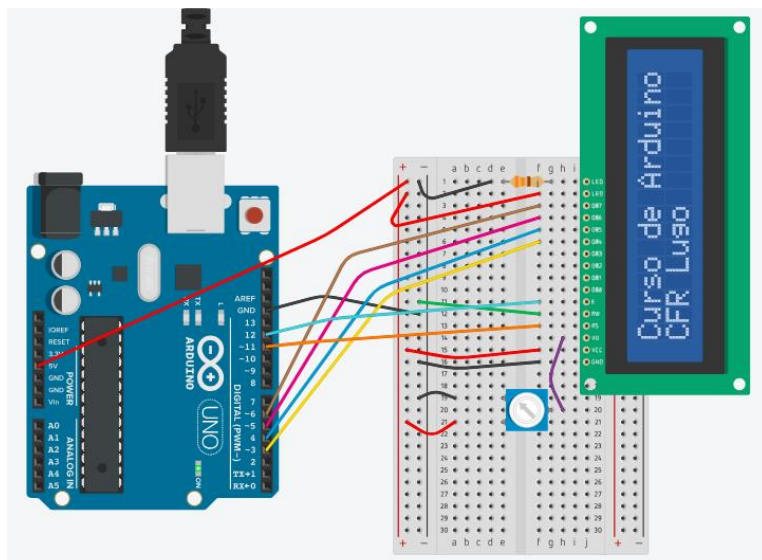
void setup()
{
  lcd.begin(16, 2);
  lcd.clear();
}

void loop()
{
  lcd.setCursor(0,0);
  lcd.print("Curso de Arduino");
  lcd.setCursor(0,1);
  lcd.print("CFR Lugo");
}

/** Function definition */
```

Funcionará??...

Podes facer a simulación tamén no TINKERCAD, proba a axustar o potenciómetro para definir o contraste correctamente.



Reto guiado 3.2. Sensor ultrasons

RETO 3.2

Modifica o reto anterior, amplíalo conectando un sensor de distancia na zona libre da protoboard. Mosa os valores do sensor a través do monitor LCD.

Resólveo en THINKERCAD ou na túa placa.

CURSO DE ARDUINO

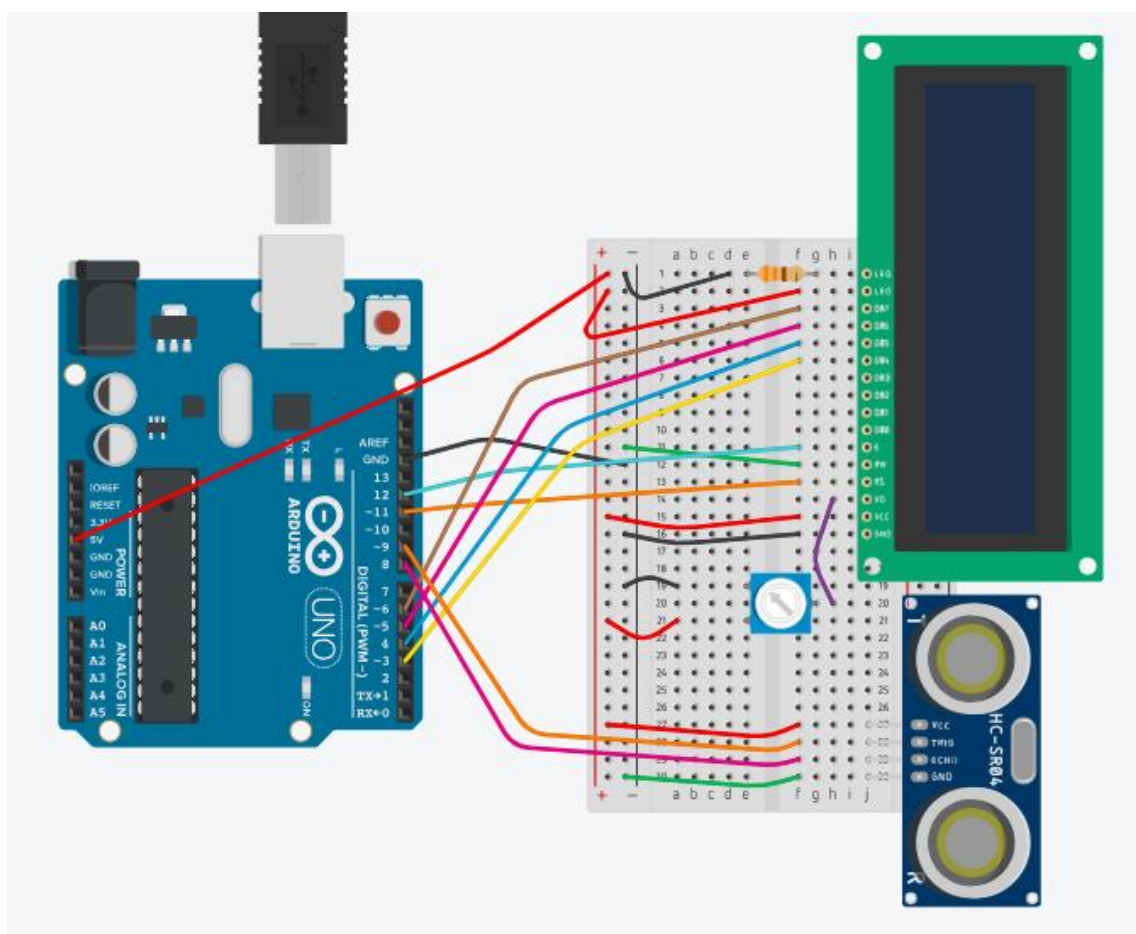
José Pino Vázquez



Seguiremos traballando co circuíto anterior pero imos a engadir o sensor de ultrasons.

O circuíto

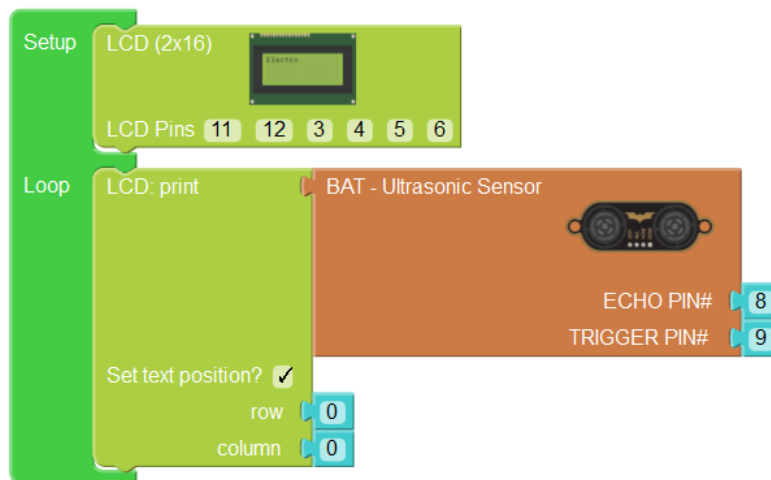
Neste caso non temos usaremos o compoñente de ultrasons de 4 pines (date de conta que hai un de 3, pero no kit da xunta ven o de 4). O circuíto podería quedar así:



Date de conta que, no TINKERCAD quedará así ben presentado, pero montado físicamente non ten moito sentido a posición do sensor ultrasónico, sen dúbida será mellor darlle á volta e prestar atención á nova ubicación do cableado.

O código

O código xérao o visualino utilizando os bloques ZUM



```
1 #include <Wire.h>
2 #include <LiquidCrystal.h>
3
4 /** Global variables */
5 LiquidCrystal lcd(11, 12, 3, 4, 5, 6);
6
7 /** Function declaration */
8 //bqBAT
9 long TP_init(int trigger_pin, int echo_pin);
10 long Distance(int trigger_pin, int echo_pin);
11
12 void setup()
13 {
14     lcd.begin(16, 2);
15     lcd.clear();
16
17
18     pinMode( 8 , INPUT );
19
20     pinMode( 9 , OUTPUT );
21
22 }
23
24
25 void loop()
26 {
27     lcd.setCursor(0,0);
28     lcd.print(Distance(9,8));
29
30 }
31
32
33 /** Function definition */
34 //bqBAT
35 long TP_init(int trigger_pin, int echo_pin)
36 {
```

```

37 digitalWrite(trigger_pin, LOW);
38 delayMicroseconds(2);
39 digitalWrite(trigger_pin, HIGH);
40 delayMicroseconds(10);
41 digitalWrite(trigger_pin, LOW);
42 long microseconds = pulseIn(echo_pin ,HIGH);
43 return microseconds;
44 }
45 long Distance(int trigger_pin, int echo_pin)
46 {
47     long microseconds = TP_init(trigger_pin, echo_pin);
48     long distance;
49     distance = microseconds/29/2;
50     if (distance == 0){
51         distance = 999;
52     }
53     return distance;
54 }

```

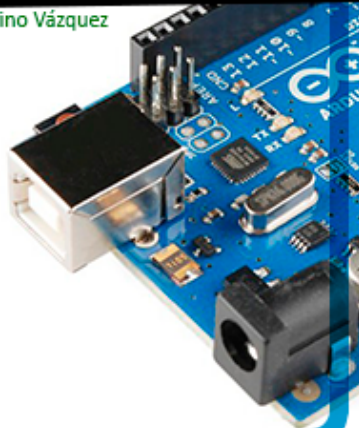
Reto guiado 3.3. Blink led con Scratch na Raspberry

RETO 3.3

Utilizando a linguaxe Scratch e os portos de comunicación GPIO, realiza a programación da placa Raspberry de xeito que fagamos o mesmo que no reto 1.1 (blink led) pero dende a propia placa.

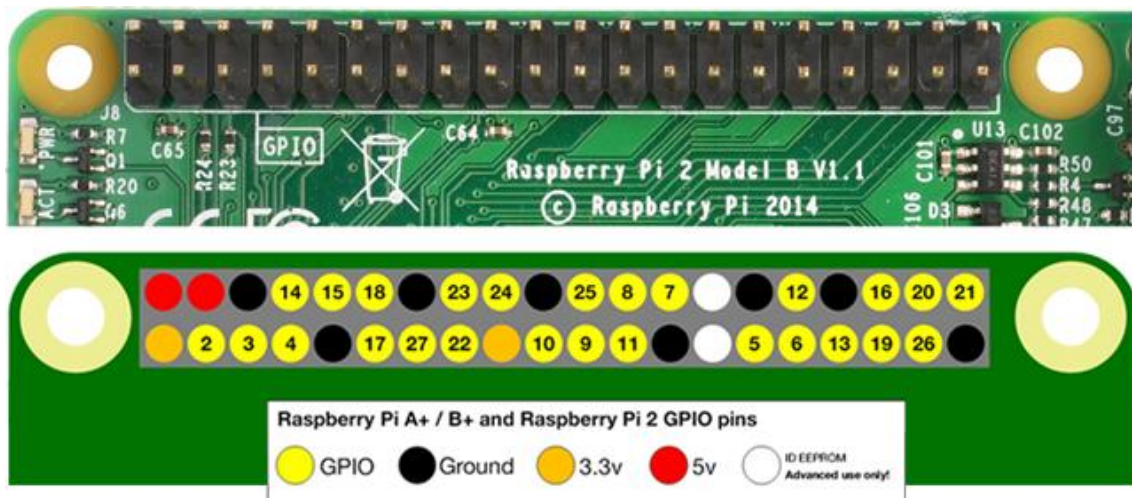
CURSO DE ARDUINO

José Pino Vázquez

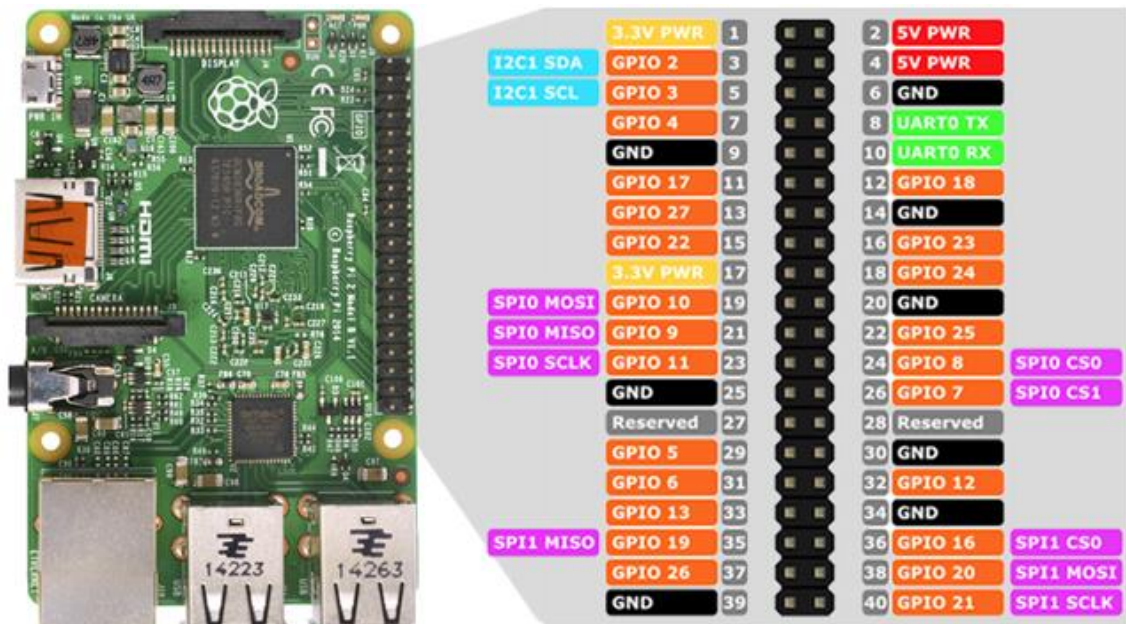


Neste caso usaremos a Raspberry para conseguir o mesmo que fixemos co arduino. Neste caso imos seguir o curso da web <https://www.programoergosum.com/cursos-online/raspberry-pi/>

Usaremos os pines GPIO, que podemos ver a continuación:



Existen 2 xeitos de numerar os pins da Raspberry Pi, en modo GPIO ou en modo BCM. Nomenclatura GPIO e BCM



O código:

Neste caso usaremos os controis de scratch, o inicio do programa faremos que sexa ao clicar sobre a bandeira verde.

Lembra que para ter os bloques de GPIO teremos que engadir os bloques na opción “More Blocks”



O circuíto:

O circuito é realmente simples, tão só teremos que conectar o pin de saída ao GPIO 4 (ou o 7 na nomenclatura BCM), e realizar a montagem.

