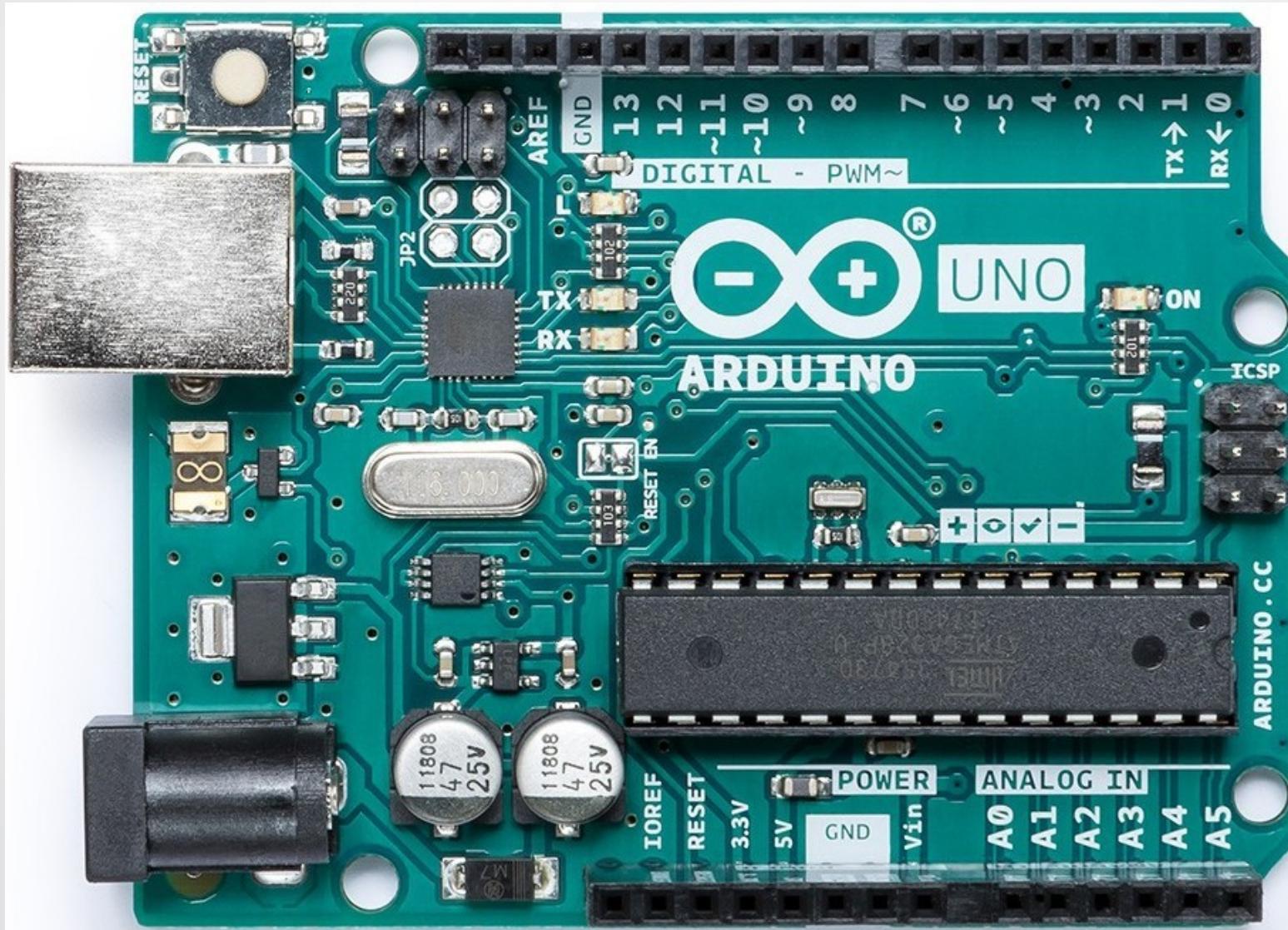
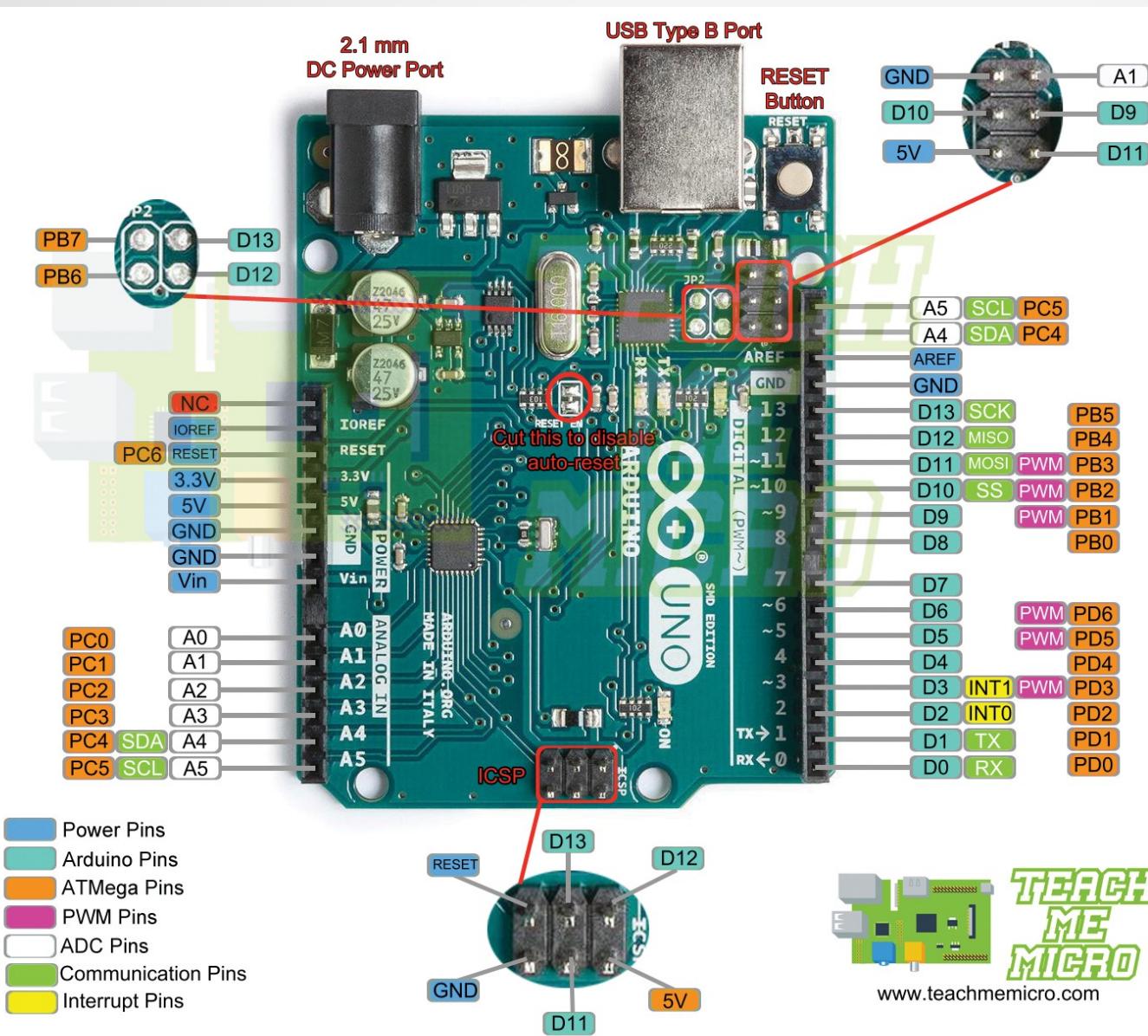


Display LCD 16x2

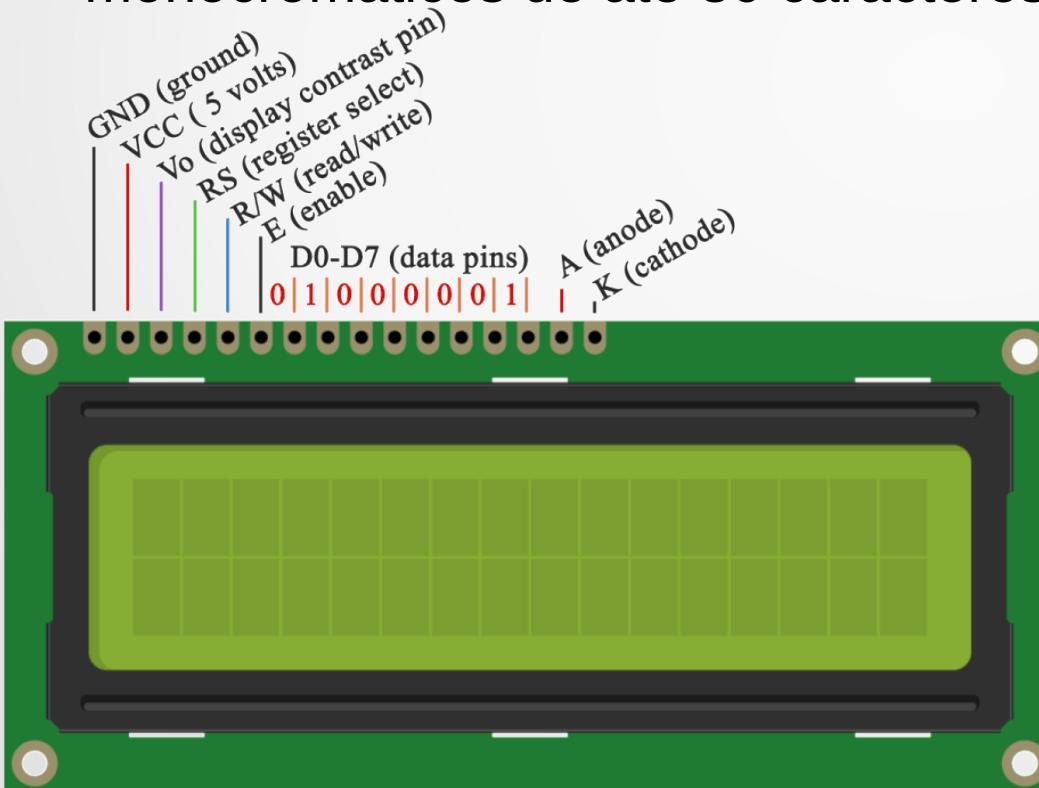


Display LCD 16x2 – pinout Arduino



Display LCD 16x2

- Podemos encontrar displays LCD en varios tamaños: 16x2, 20x4, 20x2, 40x2. Nós imos empregar o de 2 liñas e 16 caracteres por liña.
- Podémonos comunicar con estos displays mediante un controlador, habitualmente o HITACHI HD44780, que traballa con LCDs monocromáticos de até 80 caracteres.



- A cor de fondo será azul ou verde e dispoñen dun pin de entrada analóxico, co que se pode regular o contraste mediante un potenciómetro (Vo).

Display LCD 16x2 Datasheet

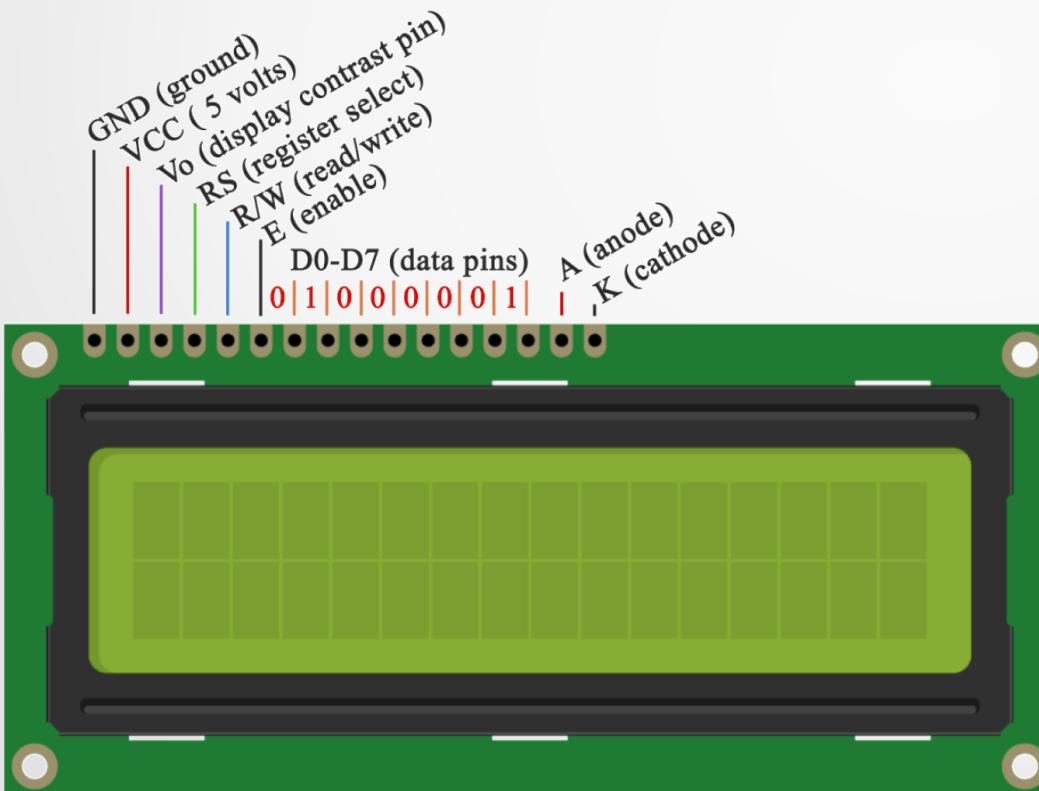
Display LCD 16x2

- O display LCD 16x2 dispón de 16 pins, tal como se indica a seguir:

- GND, VCC: alimentación
- Vo: regula o contraste do display (GND: máximo)
- RS: selecciona en que parte da memoria (rexistro) escribir os datos
- R/W: selecciona modo de lectura ou de escritura (GND: escritura)

- E: habilita a escritura nos rexistros
- D0 a D7: pins de datos, para escribir nun rexistro ou ler do mesmo
- A, K: Bklt+ e Bklt- encenden ou apagan a retroalimentación (+5V, GND)

- Os displays LCD compatibles co controlador Hitachi poden ser usados en dous modos 4-bits ou 8-bits.
- No primeiro caso requírense 7 pins I/O de Arduino, mentres que o segundo require 11 pins.
- Para mostrar texto por pantalla, pódese facer case todo no mod 4-bits.
- Nós imos empregar os pins D4, D5, D6 e D7.

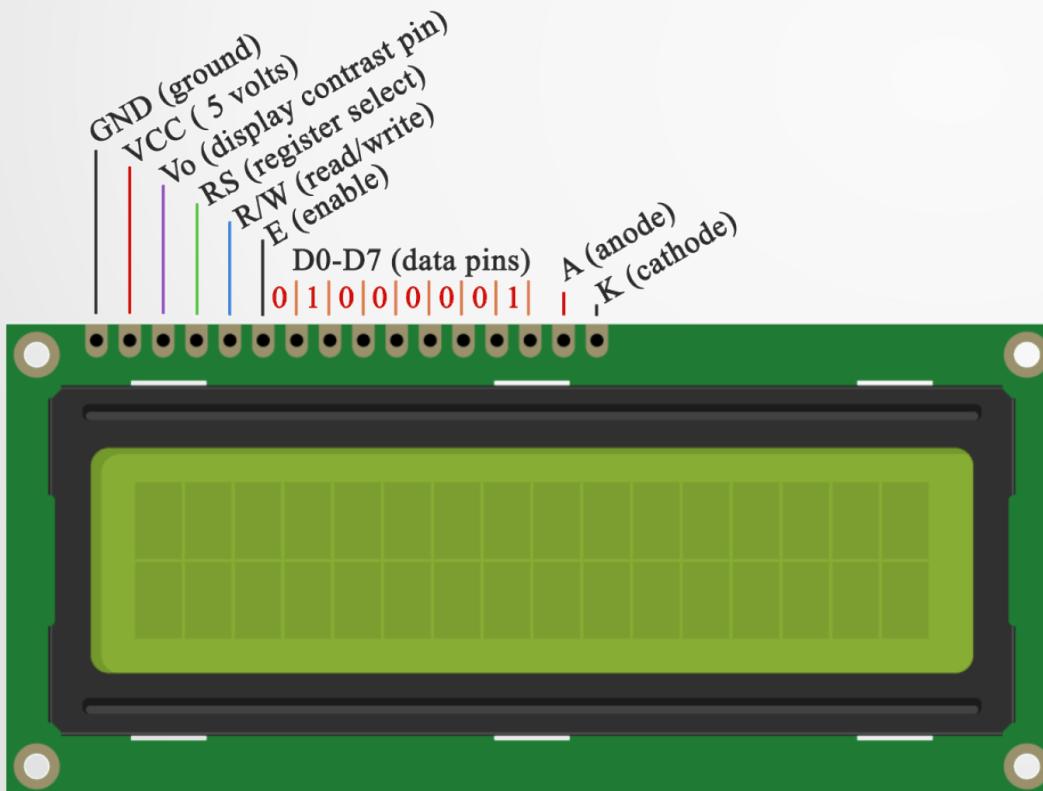


Display LCD 16x2 Datasheet

Display LCD 16x2

- Para escribir texto no display LCD, é recomendable emplegar a librería LiquidCrystal.h, que forma parte das librerías estándard de Arduino.
- A ligazón á documentación desta librería é:

<https://www.arduino.cc/en/Reference/LiquidCrystal>



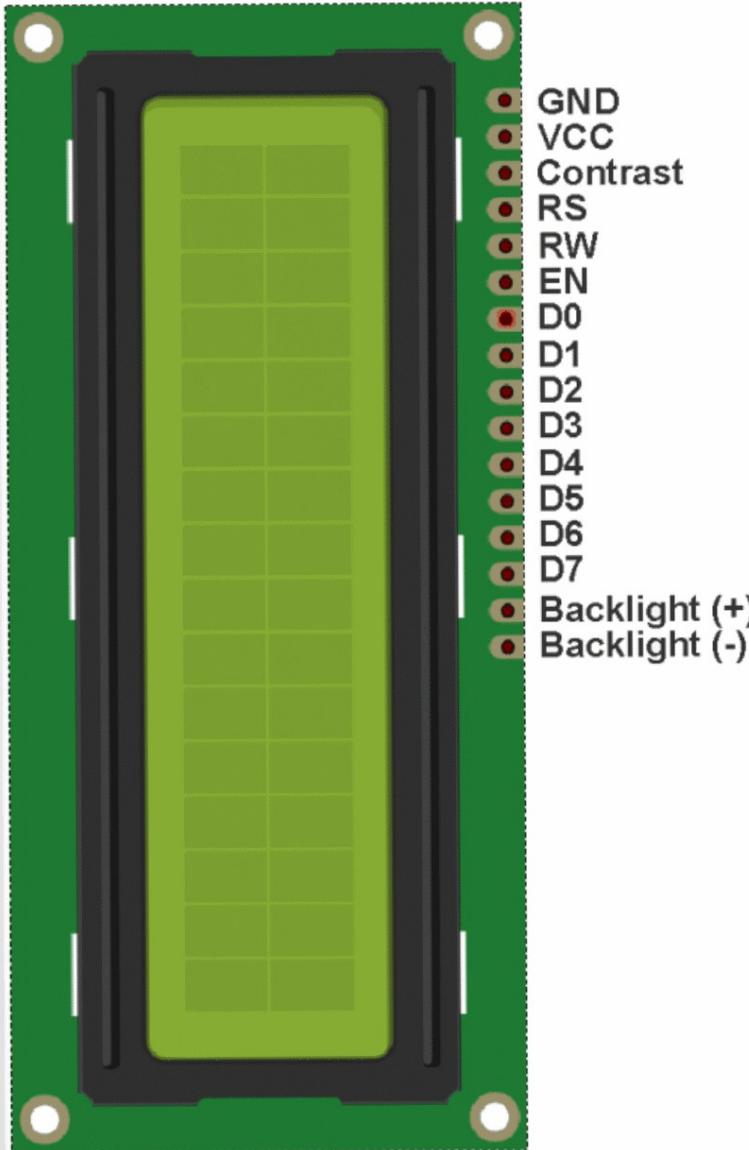
- Proporciona funcións moi útiles para o control da LCD:
 - begin()
 - clear()
 - setCursor()
 - write() e print()
 - scrollDisplayLeft() e scrollDisplayRigth()
 - etc.

Display LCD 16x2

- Imos empregar as más usadas, para aprender a usar a librería.
- Está desenvolvida empregando POO, polo que para facer uso da mesma temos que declarar un obxecto que representará ao noso display.
- O obxecto declarado proporciona acceso ás funcións (métodos).
- P.ex. se declaramos un obxecto co nome 'lcd', para facer que imprima unha mensaxe no display, facemos uso do procedemento print():

```
lcd.print('Esta é a mensaxe');
```

Display LCD 16x2



- As diferentes formas de declarar un obxecto están na figura da seguinte diapositiva.
- No caso de empregar a opción (b) ou (d), temos que asegurarnos de poñer RW a 0V.
- En calquera caso, imos manter a cousa simple e empregar só 4 pins de datos e a opción (a).

Display LCD 16x2

```
#include <LiquidCrystal.h>

//Chamamos 'lcd' ao obxecto que representa ao LCD

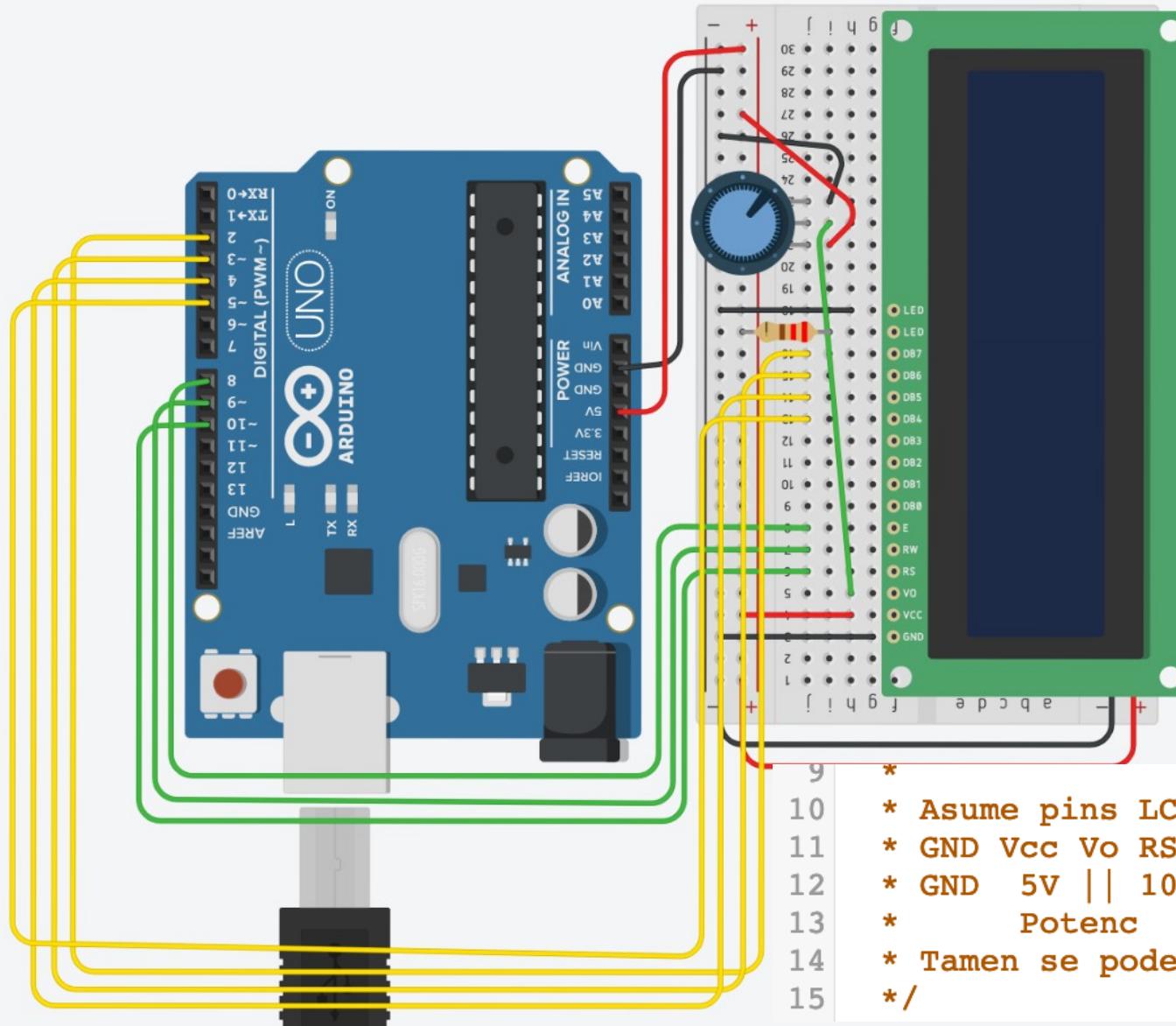
//(a) Usamos os pins RS, E e D4 a D7
//    Neste caso RW debe conectarse a GND de Arduino
LiquidCrystal lcd(rs, enable, d4, d5, d6, d7)

//(b) Usamos os pins RS, RW, E e mais D4 a D7
LiquidCrystal lcd(rs, rw, enable, d4, d5, d6, d7)

//(c) Usamos RS, E e todos os pins de datos
//    Igual que antes: RW --> GND de Arduino
LiquidCrystal lcd(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)

//(d) Usamos RS, RW, E e todos os pins de datos
LiquidCrystal lcd(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)
```

Display LCD 16x2



* Asume pins LCD --> Arduino:
* GND Vcc Vo RS RW E D0 -- D7 A K
* GND 5V || 10 9 8 ----5432 | GND
* Potenc Resist
* Tamen se pode prescindir de RW
*/

Display LCD 16x2

lcd.16x2.mensaxe

```
/*
 * Presenta por display LCD 16x2
 * unha mensaxe estática e un
 * cursor que encende e apaga.
 *
 * Programa para mostrar o uso das
 * funcións básicas da librería
 * LiquidCristal.h.
 *
 * Asume pins LCD --> Arduino:
 * GND Vcc Vo RS RW E D0 -- D7 A K
 * GND 5V || 10 9 8 ----5432 | GND
 *          Potenc      Resist
 * Tamen se pode prescindir de RW
 */

#include <LiquidCrystal.h>

//LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)
LiquidCrystal lcd(10, 9, 8, 5, 4, 3, 2);

int espera = 500;

void setup() {
    //Inicia LCD e imprime duas linhas de
    //mensaxe estatica
    lcd.begin(16, 2);
    lcd.setCursor(0, 0);
    lcd.print(" LCD preparado");
    lcd.setCursor(5, 1);
    lcd.print("OK :-)");
}

void loop() {
    //Mostra e oculta o cursor alternativamente
    lcd.noCursor();
    delay(espera);
    lcd.cursor();
    delay(espera);
}
```

- O script do exemplo fai o seguinte:
 - (a) declara un obxecto `Lcd()`
 - (b) inicia o obxecto: `Lcd.begin()`
 - (c) sitúa o cursor na primeira liña
 - (d) imprime unha mensaxe
 - (e) sitúa o cursor na segunda liña
 - (f) imprime outra mensaxe
- Finalmente apaga e encende o cursor indefinidamente cun intervalo de 500 ms.

Display LCD 16x2

```
#include <LiquidCrystal.h>

//LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)
LiquidCrystal lcd(10, 9, 8, 5, 4, 3, 2);

int espera = 500;

void setup() {
    //Inicia LCD e imprime duas linhas de
    //mensaxe estatica
    lcd.begin(16, 2);
    lcd.setCursor(0, 0);
    lcd.print(" LCD preparado");
    lcd.setCursor(5, 1);
    lcd.print("OK :-)");
}

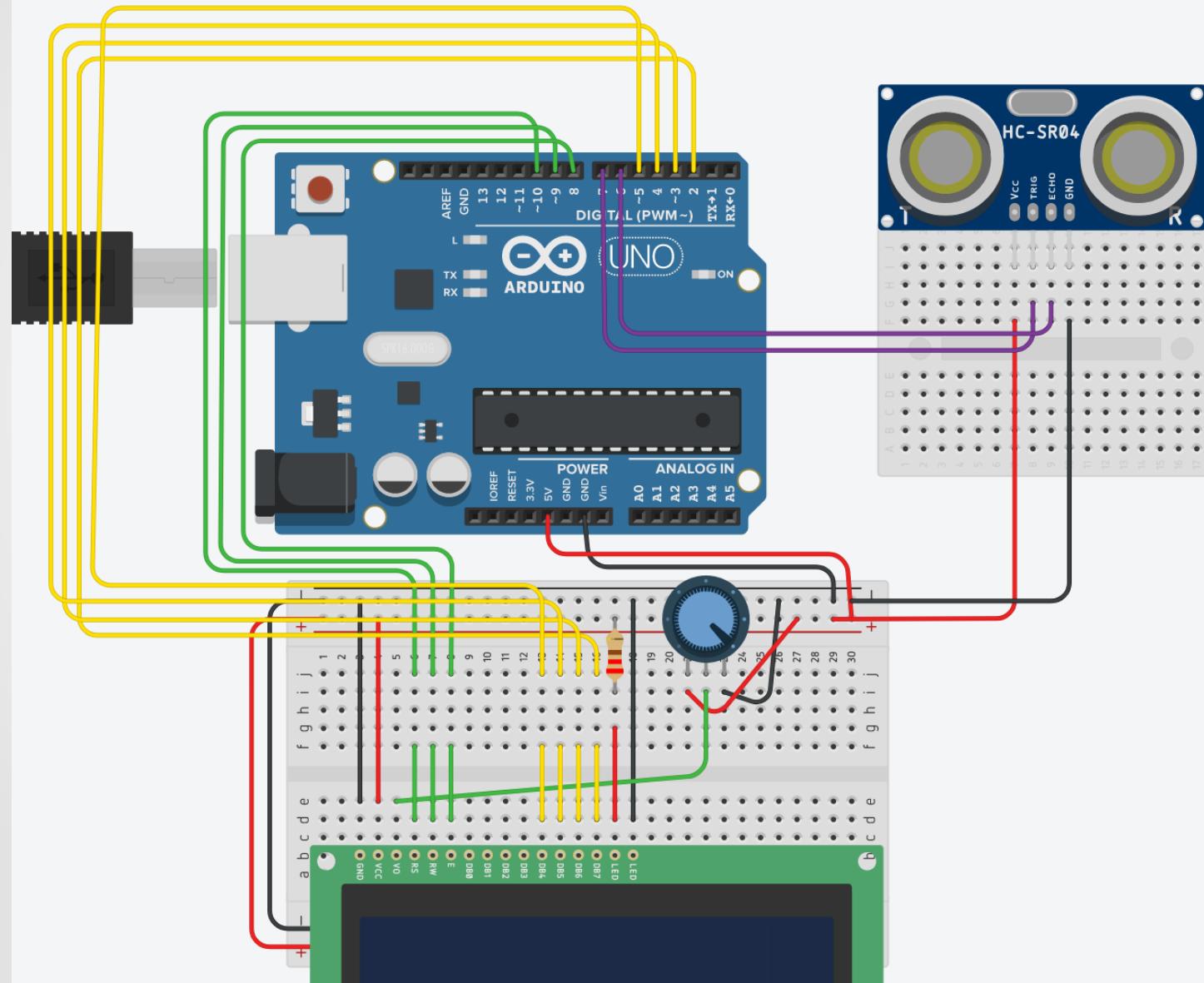
void loop() {
    //Mostra e oculta o cursor alternativamente
    lcd.noCursor();
    delay(espera);
    lcd.cursor();
    delay(espera);
}
```

- Este script exemplifica o uso do display:
 - declarar e iniciar: lcd(), lcd.begin()
 - situar o cursor e imprimir: lcd.setCursor(), lcd.print()
 - mostar e ocultar o cursor: lcd.cursor(), lcd.noCursor()

Display LCD 16x2

- Imos facer outro exemplo, combinando co sensor de ultrasóns.
- Reaproveitando o código do exemplo anterior e o dos exemplos de HC-SR04, imos mostrar por display as lecturas do sensor periodicamente, así como un aviso de límites de tolerancia.
- Empregamos a mesma montaxe do LCD e engadimos o HC-SR04.

Display LCD 16x2



Display LCD 16x2

```
lcd.16x2.sensor.distancia.ultrasons
/*
 * Presenta por display LCD 16x2
 * a distancia a un obxecto que
 * proporciona un sensor ultrasons
 *
 * Asume pins LCD --> Arduino:
 * GND Vcc Vo RS RW E D0 -- D7 A K
 * GND 5V I1 10 9 8 ----5432 I GND
 * Potenc Resist
 * Tamen se pode prescindir de RW
 */

#include <LiquidCrystal.h>

#define TRIGGER 7
#define ECHO 6
//Definimos macro para calcular distancia
#define calcDist(t) t*0.017 //en cm

#define DIST_MIN 10
#define DIST_MAX 150

//Declaramos obxecto LiquidCristal
//LiquidCrystal(rs, rw, enable, d4, d5,
LiquidCrystal lcd(10, 9, 8, 5, 4, 3, 2);

float tempo, distancia;
String mensaxe = "Dist: ";

int espera = 400;
```

```
lcd.16x2.sensor.distancia.ultrasons
int espera = 400;

void setup() {
    //Declaramos pins do sensor ultr
    pinMode(TRIGGER, OUTPUT);
    pinMode(ECHO, INPUT);
    //Inicia LCD e imprime duas linh
    //mensaxe estatica
    lcd.begin(16, 2);
    lcd.setCursor(0, 0);
    lcd.print(" LCD preparado");
    lcd.setCursor(5, 1);
    lcd.print("OK :-)");
    delay(15*espera);
}

void loop() {
    //Enviamos un pulso co HC-SR04
    digitalWrite(TRIGGER, LOW);
    delayMicroseconds(5);
    digitalWrite(TRIGGER, HIGH);
    delayMicroseconds(25);
    digitalWrite(TRIGGER, LOW);
    tempo = pulseIn(ECHO, HIGH);

    //Calculamos a distancia en cm
    distancia = calcDist(tempo);

    //Borra display
    lcd.clear();
```

```
lcd.16x2.sensor.distancia.ultrasons
digitalWrite(TRIGGER, HIGH);
delayMicroseconds(25);
digitalWrite(TRIGGER, LOW);
tempo = pulseIn(ECHO, HIGH);

//Calculamos a distancia en cm
distancia = calcDist(tempo);

//Borra display
lcd.clear();

//Mostra lecturas do sensor
lcd.setCursor(0, 0);
lcd.print(mensaxe + String(distancia) + " cm");
lcd.setCursor(0, 1);
if(distancia <= DIST_MIN) {
    lcd.setCursor(1, 1);
    lcd.print("PRECAUCION!!");
}
if(distancia >= DIST_MAX) {
    lcd.setCursor(1, 1);
    lcd.print("FORA DE RANGO!");
}
//Mostra e oculta o cursor alternativamente
lcd.noCursor();
delay(espera);
lcd.cursor();
delay(espera);}
```

Display LCD 16x2

- No script anterior engadimos a programación relativa ao HC-RS04, de maneira que obtemos o pulso e a partir do tempo de demora calculamos a distancia ao obxecto.
- A seguir, borramos a mensaxe estática do Display e imprimimos a mensaxe da liña superior. Despois a da liña inferior no caso de térense superado os límites.
- Despois móstrase o cursor e pásaxe a unha nova medición.
- O seguinte exemplo é para mostrar as funcíons de scroll e como imprimir texto que se move ao longo do display.

Display LCD 16x2

```
//Borra display
lcd.clear();

//Mostra lecturas do sensor
mFila1 = "Dist: " + String(distancia) + " cm";
lcd.setCursor(0,0);
lcd.print(mFila1);
//Mostra aviso de limites do sensor
if(distancia <= DIST_MIN)
    mFila2 = "PRECAUCION!!";
else if(distancia >= DIST_MAX)
    mFila2 = "FORA DE RANGO!";
else
    mFila2 = "";
lcd.setCursor(0,1);
lcd.print(mFila2);
delay(espera);
//Move o texto a dereita...
for(int i=0; i<16; i++) {
    lcd.scrollDisplayRight();
    delay(espera/3);
}
//...volta a esquerda...
for(int i=0; i<32; i++) {
    lcd.scrollDisplayLeft();
    delay(espera/3);
}
//...e novamente a dereita
for(int i=0; i<16; i++) {
    lcd.scrollDisplayRight();
    delay(espera/3);
}
//Cursor encende e apaga
lcd.noCursor();
delay(espera);
lcd.cursor();
delay(espera);
}
```

- Neste novo exemplo engadimos tres bucles for() antes do apagado e encendido do cursor.
- Nestes bucles desprázaxe o texto á dereita até ocultalo, vólvese desprazar á esquerda até que non se ve e finalmente desprázase á dereita xusto antes de actualizar o novo valor da distancia.
- Para isto empregamos os tres bucles for() e combinación cos métodos lcd.scrollDisplayRight() e lcd.scrollDisplayLeft(), desprazan o texto impreso previamente.

Display LCD 16x2

```
lcd.16x2.sensor.distancia.texto.deslizante
/*
 * Presenta por display LCD 16x2
 * a distancia a un obxecto que
 * proporciona un sensor ultrasons.
 *
 * Fai scrolllo do texto a dereita
 * e esquerada, usando funcions
 * da libreria LiquidCrystal
 *
 * Asume pins LCD --> Arduino:
 * GND Vcc Vo RS RW E D0 -- D7 A K
 * GND 5V 11 10 9 8 ----5432 I GND
 * Potenc Resist
 * Tamen se pode prescindir de RW
 */

#include <LiquidCrystal.h>

#define TRIGGER 7
#define ECHO 6
//Definimos macro para calcular distancia
#define calcDist(t) t*0.017 //en cm

#define DIST_MIN 10
#define DIST_MAX 150

//Declaramos obxecto LiquidCristal
//LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)
LiquidCrystal lcd(10, 9, 8, 5, 4, 3, 2);

float tempo, distancia;
String mFila1 = "", mFila2 = "";

int espera = 400;

void setup() {
    //Declaramos pins do sensor ultrasons
    pinMode(TRIGGER, OUTPUT);
    pinMode(ECHO, INPUT);
    //Inicia LCD e imprime duas linhas de
    //mensaxe estatica
    lcd.begin(16, 2);
```

```
lcd.16x2.sensor.distancia.texto.deslizante
//mensaxe estatica
lcd.begin(16, 2);
lcd.setCursor(0, 0);
lcd.print(" LCD preparado");
lcd.setCursor(5, 1);
lcd.print("OK :-)");
delay(4*espera);

}

void loop() {
    //Enviamos un pulso co HC-SR04
    digitalWrite(TRIGGER, LOW);
    delayMicroseconds(5);
    digitalWrite(TRIGGER, HIGH);
    delayMicroseconds(25);
    digitalWrite(TRIGGER, LOW);
    tempo = pulseIn(ECHO, HIGH);

    //Calculamos a distancia en cm
    distancia = calcDist(tempo);

    //Borra display
    lcd.clear();

    //Mostra lecturas do sensor
    mFila1 = "Dist: " + String(distancia) + " cm";
    lcd.setCursor(0,0);
    lcd.print(mFila1);
    //Mostra aviso de limites do sensor
    if(distancia <= DIST_MIN)
        mFila2 = "PRECAUCION!!";
    else if(distancia >= DIST_MAX)
        mFila2 = "FORA DE RANGO!";
    else
        mFila2 = "";
    lcd.setCursor(0,1);
    lcd.print(mFila2);
    delay(espera);

    //Move o texto a dereita...
    for(int i=0; i<16; i++) {
        lcd.scrollDisplayRight();
        delay(espera/3);
    }
    //...volta a esquerda...
    for(int i=0; i<32; i++) {
        lcd.scrollDisplayLeft();
        delay(espera/3);
    }
    //...e novamente a dereita
    for(int i=0; i<16; i++) {
        lcd.scrollDisplayRight();
        delay(espera/3);
    }
    //Cursor encende e apaga
    lcd.noCursor();
    delay(espera);
    lcd.cursor();
    delay(espera);
```

```
lcd.16x2.sensor.distancia.texto.deslizante
//Calculamos a distancia en cm
distancia = calcDist(tempo);

//Borra display
lcd.clear();

//Mostra lecturas do sensor
mFila1 = "Dist: " + String(distancia) + " cm";
lcd.setCursor(0,0);
lcd.print(mFila1);
//Mostra aviso de limites do sensor
if(distancia <= DIST_MIN)
    mFila2 = "PRECAUCION!!";
else if(distancia >= DIST_MAX)
    mFila2 = "FORA DE RANGO!";
else
    mFila2 = "";
lcd.setCursor(0,1);
lcd.print(mFila2);
delay(espera);

//Move o texto a dereita...
for(int i=0; i<16; i++) {
    lcd.scrollDisplayRight();
    delay(espera/3);
}
//...volta a esquerda...
for(int i=0; i<32; i++) {
    lcd.scrollDisplayLeft();
    delay(espera/3);
}
//...e novamente a dereita
for(int i=0; i<16; i++) {
    lcd.scrollDisplayRight();
    delay(espera/3);
}
//Cursor encende e apaga
lcd.noCursor();
delay(espera);
lcd.cursor();
delay(espera);
```

Display LCD 16x2

- Nesta unidade aprendemos a:
 - cablear un display LCD 16x2,
 - chamar á librería LiquidCrystal.h e declarar obxectos que representan ao display,
 - usar os métodos proporcionados polos obxectos da librería LiquidCrystal.h, nomeadamente:
 - `Lcd.begin()`
 - `Lcd.setCursor()`, `Lcd.print()`, `Lcd.clear()`
 - `Lcd.cursor()`, `Lcd.noCursor()`
 - `Lcd.scrollDisplayRight()`, `Lcd.scrollDisplayLeft()`