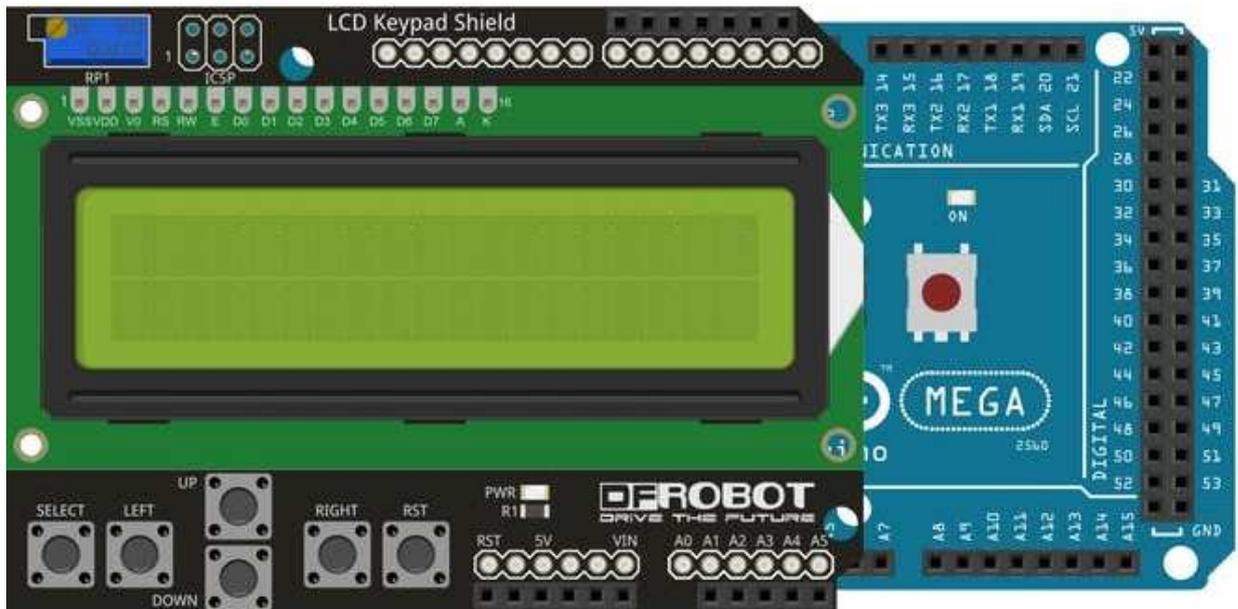


LCD KEYPAD SHIELD

Aunque podemos usar el display LCD, como hemos explicado en otro documento, resultan evidentes los inconvenientes que entrañan el cableado, las soldaduras, etc. Afortunadamente disponemos de un escudo que se inserta directamente sobre la tarjeta Arduino y que integra, además del display LCD de 2 filas y 16 columnas, 5 pulsadores, lo que nos facilita la realización de otros programas. El contraste del LCD se controla manualmente manipulando el pequeño potenciómetro situado en la esquina superior izquierda de la imagen.



Si en lugar que montar nuestro escudo sobre una placa Arduino UNO, que queda totalmente cubierta, lo hacemos sobre una placa Arduino MEGA, disponemos de gran cantidad de pines al descubierto para conectar otros elementos.



Para el manejo de este escudo también utilizaremos la librería LiquidCrystal que ya viene incluida en el entorno integrado de Arduino y que dispone de varias funciones que ya conocemos.

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

Los pines utilizados por el escudo vienen determinados, por lo que no podemos cambiarlos. Tendremos que incluir en nuestro programa la orden:

```
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
```

Donde "lcd" es el nombre que le hemos dado a nuestro display.

En el típico ejemplo de escribir el mensaje Hola, mundo, ya quedan claras las principales funciones de la librería.

```
#include <LiquidCrystal.h> //incluye la librería LiquidCrystal
LiquidCrystal lcd(8, 9, 4, 5, 6, 7); //Definimos los pines usados

void setup() {
  lcd.begin(16, 2); //indicamos número de columnas y filas del display
  lcd.clear(); //limpiamos la pantalla
  lcd.setCursor(0,1); //Situamos el cursor en la primera columna y segunda fila
  lcd.print("Hola, mundo"); //Escribimos un mensaje
}

void loop() {
}
```

Cuando queremos imprimir el valor de variables se utiliza la función .print(), igual que con las cadenas de texto. Por ejemplo, si tenemos una variable que se llama temperatura, tan solo hay que ejecutar:

```
lcd.print(temperatura);
```

Si queremos imprimir cadenas de texto y variables mezcladas, podemos hacerlo utilizando variables de tipo String.

Ejemplo: Programa que lee un nivel de luminosidad de una LDR e imprime su valor en el display con intervalos de 1 segundo.

```
#include <LiquidCrystal.h> //incluye la librería LiquidCrystal
String cadena1 = "Luminosidad ";
int lumin=0;
LiquidCrystal lcd(8, 9, 4, 5, 6, 7); //Definimos los pines usados

void setup() {
  lcd.begin(16, 2); //indicamos número de columnas y filas del display
}

void loop() {
  lumin=analogRead(A8); //Leemos una LDR en pin analógico A8
  lcd.clear(); //limpiamos la pantalla
  lcd.setCursor(0,0); //Situamos el cursor en la primera columna y primera fila
  lcd.print(cadena1+lumin); //Escribimos mensaje seguido de variable
  delay(1000); //Refrescamos dato medido cada segundo
}
```

El uso de los pulsadores

Vamos ahora a aprender a utilizar los 5 pulsadores del escudo (hay un sexto pulsador, el situado más a la derecha que es un Reset de la placa).

El funcionamiento de los pulsadores del escudo se fundamenta en que éstos forman parte de un divisor de tensión de modo que podemos leer unos valores de tensión bien diferenciados en el pin analógico A0 dependiendo del pulsador pulsado.

Concretamente, los valores teóricos son:

Botón pulsado	Lectura en pin A0
RIGHT	0
UP	145
DOWN	330
LEFT	505
SELECT	740
Ninguno	1023

No obstante, las resistencias tienen tolerancias de fabricación, por lo que los valores reales variarán. Lo mejor es medirlos en nuestra placa y trabajar en un rango. Por ejemplo:

Botón pulsado	Lectura en pin A0
RIGHT	0 a 50
UP	50 a 250
DOWN	250 a 450
LEFT	450 a 650
SELECT	650 a 850
Ninguno	> 850

Por tanto, debemos incluir en nuestros programas una función como la función lee_botones() que sigue a continuación, que devuelve un valor entero entre 0 y 5 en función del botón pulsado. Para no tener que memorizar números definimos unas constantes.

```
//Constantes para identificar las teclas
#define btnRIGHT    0
#define btnUP       1
#define btnDOWN     2
#define btnLEFT     3
#define btnSELECT   4
#define btnNINGUNO  5

int lee_botones_LCD(){
    int lectura = analogRead(0);
    if(lectura>850) return btnNINGUNO;
    if(lectura<50) return btnRIGHT;
    if(lectura<250) return btnUP;
    if(lectura<450) return btnDOWN;
    if(lectura<650) return btnLEFT;
    if(lectura<850) return btnSELECT;
    return btnNINGUNO;
}
```

Ejemplo: el siguiente programa lee el botón que se pulsa y lo indica en la pantalla del display. Aparece el nombre del botón mientras se mantiene pulsado y aparece la palabra NINGUNO cuando se deja de pulsar.

```
#include <LiquidCrystal.h>

//Constantes para identificar las teclas
#define btnRIGHT    0
#define btnUP      1
#define btnDOWN    2
#define btnLEFT    3
#define btnSELECT  4
#define btnNINGUNO 5

int boton_pulsado = 5; //Variable para leer el botón pulsado
LiquidCrystal lcd(8, 9, 4, 5, 6, 7); //Indico pines usado por LCD

void setup() {
  lcd.begin(16,2);           //LCD de 16 columnas y 2 filas
  lcd.clear();              //Borrar la pantalla
  lcd.setCursor(0,0);       //Llevar el cursor al inicio
  lcd.print("Prueba de teclas:");
}

void loop() {
  lcd.setCursor(0,1); // Lleva el cursor al inicio de la segunda línea
  boton_pulsado = lee_botones_LCD(); // Lee el botón pulsado
  if(boton_pulsado==btnRIGHT) lcd.print("DERECHO ");
  else if(boton_pulsado==btnLEFT) lcd.print("IZQUIERDO ");
  else if(boton_pulsado==btnUP) lcd.print("ARRIBA ");
  else if(boton_pulsado==btnDOWN) lcd.print("ABAJO ");
  else if(boton_pulsado==btnSELECT) lcd.print("SELECCION");
  else if(boton_pulsado==btnNINGUNO) lcd.print("NINGUNO ");
}

int lee_botones_LCD() {
  int lectura = analogRead(0);
  if(lectura>850) return btnNINGUNO;
  if(lectura<50) return btnRIGHT;
  if(lectura<250) return btnUP;
  if(lectura<450) return btnDOWN;
  if(lectura<650) return btnLEFT;
  if(lectura<850) return btnSELECT;
  return btnNINGUNO;
}
```