



Arduino Inicial

Ing. Juan C. Abdala

Clase 4



Objetivo de esta clase

- Aprender sobre los Rele.
- Aprender sobre el Shield LCD con botones DFR_LCD_Keypad
- Aprender cómo se crea una librería.



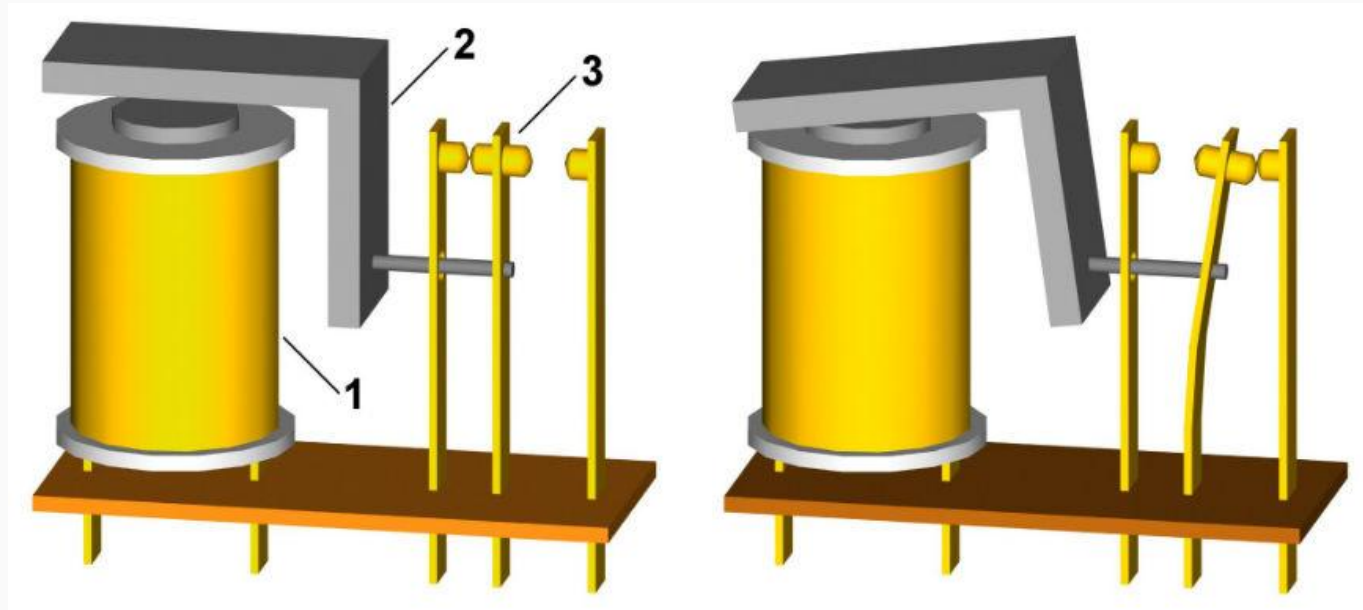
Rele

Rele

- Un relé es un dispositivo electromecánico que funciona como interruptor controlado por un circuito eléctrico en el que por medio de una bobina o electroimán, se acciona un juego de contactos que permiten abrir o cerrar otros circuitos eléctricos.
- Es capaz de controlar un circuito de salida de mayor potencia que el de entrada.



Rele por dentro



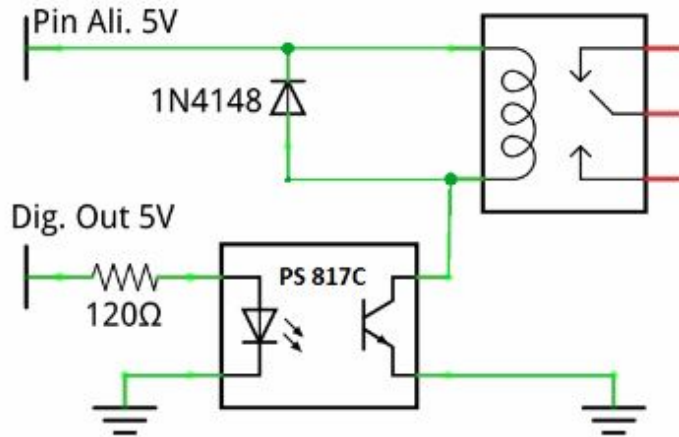
Rele Arduino



- Tan simple como un led, pero se activa por bajo y apaga por alto.
- Podemos controlar Luces, o electrodomésticos.
- Nuestro periférico tiene 2 Relés.
- Soportan hasta 220v a 10 A
- OJO tenemos que saber que corriente necesita nuestra CARGA.



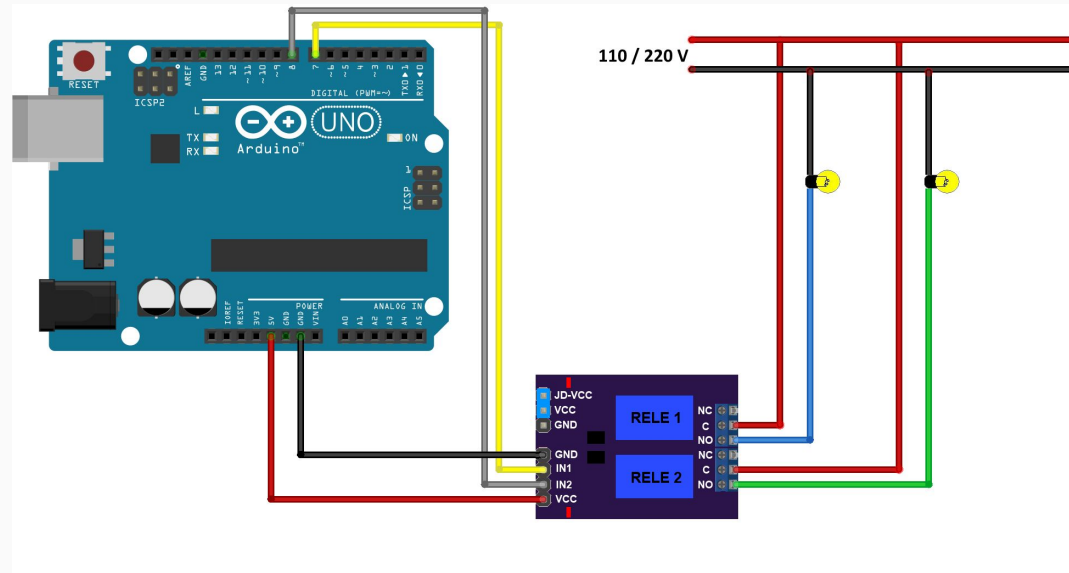
Periferico por Dentro



- Nuestros Relés necesitan unos 80 ma para que las bobinas muevan la placa.
- Nuestro arduino no entrega mas de 40 ma en el pin digital.
- Se adapta con un optoacoplador o un Transistor.
- NO NOS IMPORTA VIENE TODO ARMADO!!



Esquematico Rele

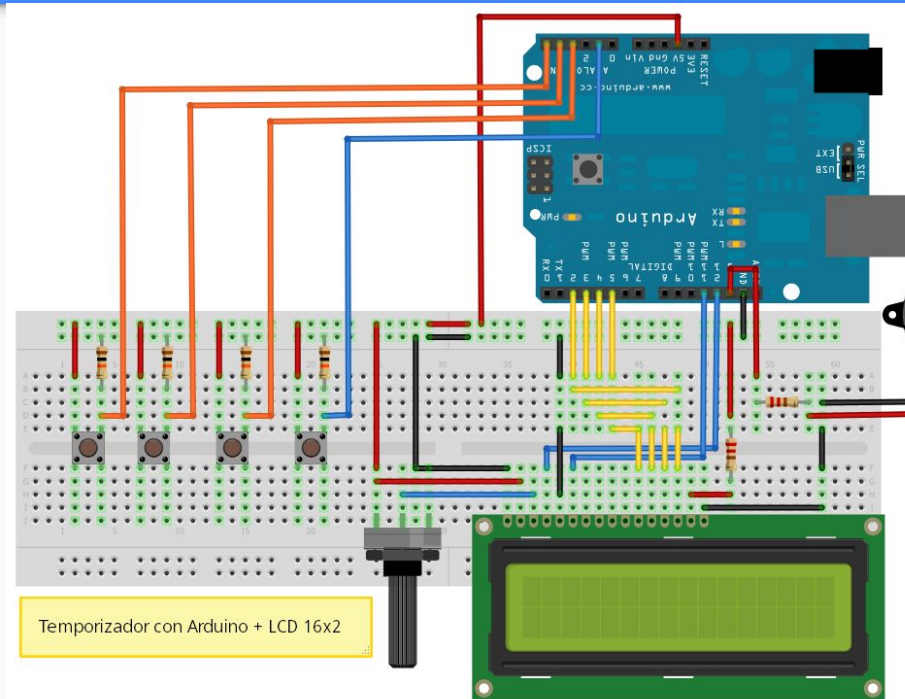


LCD SHIELD

SHIELD LCD DFR_LCD_Keypad



Sería parecido a:

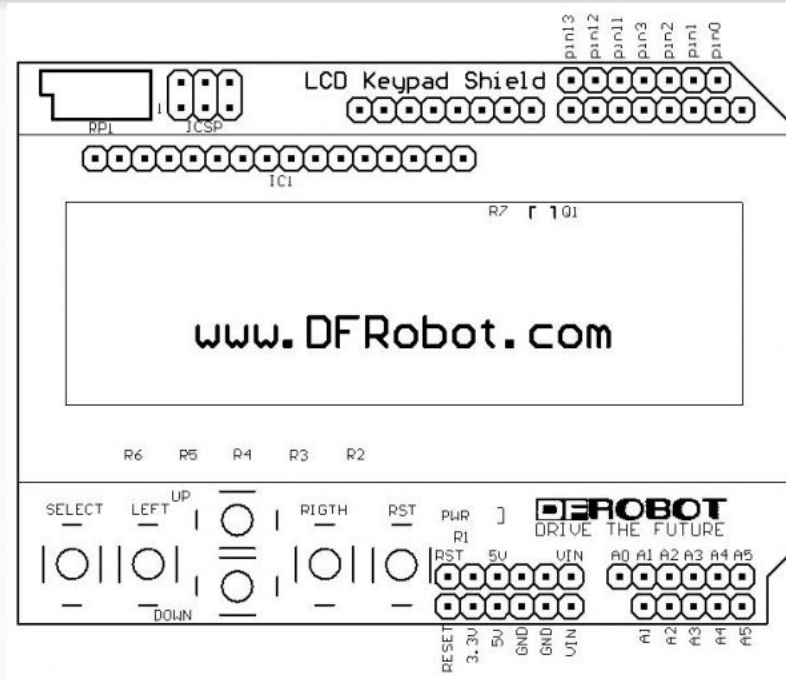


Pines ocupados

Pin	Function
Analog 0	Button (select, up, right, down and left)
Digital 4	DB4
Digital 5	DB5
Digital 6	DB6
Digital 7	DB7
Digital 8	RS (Data or Signal Display Selection)
Digital 9	Enable
Digital 10	Backlit Control



Pinout del SHIELD



inicialización.

Creamos el objeto:

```
DFR_Keypad keypad(16, 2, keyPin, 3);
```

Después solo lo usamos.



Funciones

//define dónde va a empezar a escribir

keypad.setCursor(columna, fila);

//imprime en pantalla

keypad.print("palabra");

//limpia la pantalla

keypad.clear()



Nuestra Libreria

Como podemos escalar

Debemos crear nuestras librerías.

En la misma carpeta que el sketch creamos **2 archivos**.

Un HEADER y el Código Fuente



Simple

Carpeta Libreria

- Libreria.h
- Libreria.cpp
- Carpeta examples



HEADER libreria .h

- Este archivo es donde vamos a declarar los métodos y atributos necesarios para la librería.
- extern evita que llame instancias en el código.
- Puede no ponerse y crear un constructor

En el h debe tener.

```
#ifndef __UTILIDADES__  
#define __UTILIDADES__  
class UtilidadesClass  
{  
    public:  
        static void inicio();  
        static int funcion(int parametro);  
  
};  
extern UtilidadesClass utilidades;  
#endif
```



FUENTE libreria.cpp

En este archivo está la lógica de la librería aquí escribiremos el código.

Aquí deben estar declarados todos los métodos del archivo *Libreria.h* .

```
#include "utilidades.h"

void UtilidadesClass::inicio(void)
{
    //Algo que debe poner en el setup
}

int UtilidadesClass::funcion(int parametro)
{
    //La funcion que quiera realizar
    return algunParametro;
}
```



En mi proyecto

En mi proyecto solo debo llamar:

```
#include "utilidades.h"
```

y usarla con:

```
util.inicio();
```

```
dato = util.funcion(10);
```



Descansemos 15 min

15:00



Practico

Vamos al Practico 4:

Material: <https://github.com/jcabdala/ArduinolnicialUNC>



Consultas:



abdalajc@gmail.com



@toniabdala

Google Group: arduino-unc-septiembre@googlegroups.com

