



INTRODUCCIÓN A ARDUINO II

Basado en los tutoriales de Bitwise Ar

Descripción breve

Basándome de los tutoriales de Bitwise Ar he confeccionado este tutorial para iniciarse desde 0. Explicaciones muy detalladas y montaje paso a paso.



Pere Manel Verdugo Zamora [Dirección de correo electrónico]

Contenido

Capítulo 1 Introducción Arduino	4
Capítulo 2 Instalación del IDE de Arduino y mi primer programa	6
Monitor Serie	13
Capítulo 3 Entradas/Salidas digitales con Pulsador y LED	
Sentencia if	20
else	22
Resistencia pull-down y pull-up	23
for	24
while	26
switch	27
Capítulo 4 PWM con LED y Potenciómetro para Brillo/Intensidad	
map	
Capítulo 5 HC- SR04 Sensor Ultrasónico	
Capítulo 6 Servomotor (conexión, modelos, ajustes para uso óptimo)	
Capítulo 7 Temperatura y Humedad ambiente con DHT22 o DHT11	
Capítulo 8 Módulo Relé con optoacoplador (relay Electromecánico)	
Capítulo 9 Teclado Keypad 4x4 y simple control de acceso con clave	54
Capítulo 10 Módulo LCD 1602A con librería LiquidCrystal	58
Capítulo 11 PIR Sensor de movimiento Infrarrojo Pasivo HC-SR501	62
Capítulo 12 LED RGB y módulo KY-016 Ky-009 (ánodo/cátodo común)	65
Capítulo 13 Sensor de Sonido KY-038 y Ky-037 (Modo digital)	71
Capítulo 14 Sensor Crepuscular pro fotorresistencia (LDR)	74
Capítulo 15 Sensor de Orientación (por interruptor de mercurio)	77
Capítulo 16 Sensor de Vibración KY-002 (y aplicaciones)	
Capítulo 17 Módulo Joystick analógico KY-023	
Capítulo 18 Sensor de Campo Magnético modelos KY (por efecto Hall)	
Capítulo 19 Interrupciones externas (pruebas con KY-010)	
Capítulo 20 Codificador rotatorio KY-040 (rotay encoder)	92
Capítulo 21 LM35 Sensor analógico de temperatura + LCD 1602A	
Capítulo 22 Bluetooth HC-05 Introducción y comandos AT	
Capítulo 23 Bluetooth HC-05 (App en teléfono Android + LEDs + PWM)	
Capítulo 24 Receptor Infrarrojo KY-022 + LED RGB (Lector Código)	
Capítulo 25 Sensor de línea KY-033 (seguidor de línea)	
Capítulo 26 Sensor de gas natural MQ-5 (y combustibles)	
Capítulo 27 Neopixel LED RGB inteligente WS2812 (y titas)	126

Capítulo 29 Librería Stepper con 28BYJ-48(PAP unipolar) y ULN2003	132
Capítulo 30 PRO MINI y conversores USB-TTL PL2303 CH340 CP2102	
Capítulo 31 LCD 12C adaptador e instalación de librerías específicas	142
Capítulo 32 BMP280 Presión atmosférica, Temperatura, Altitud	148
Capítulo 33 Lector tarjeta MicroSD bus SPI (y data-logger DHT11)	154
Capítulo 34 Reloj de Tiempo Real (RTC) DS3231 bus 12C	161
Capítulo 35 EEPROM memoria incorporada No Volátil teoría / práctica	168
Capítulo 36 RFID RC522 kit lector, tarjeta, llavero 13.56 Mhz	173
Capítulo 37 Matriz LED 8x8 con MAX7219 y LedControl	182
Capítulo 38 NANO modelos FT232RL y CH340G (con enlaces a drivers)	203
Capítulo 39 OLED 128x64 12C bicolor basado en SDD1306 por bus 12C	208
Capítulo 40 Sensor de agua (wáter sensor) con esquemático	215
Capítulo 41 Buzzers active y pasivo (zumbadores) tonos y melodías	219
Capítulo 28 Paso a Paso (unipolar) 28BYJ-48 con Driver ULN2003	230
Capítulo 42 TM1637 módulo 4 dígitos y 7 segmentos con dos puntos	238
Capítulo 43 Ritmo cardiaco (pulso) con KY-039 por infrarrojo	244
Capítulo 44 Fuente de protoboard Arduino y electrónica MB102	247
Capítulo 45 Niveles lógicos y Niveles de señal 5 – 3,3 V	250
Capítulo 46 LM2596 Conversor DC-DC reductor de Salida Variable	256
Capítulo 47MPU6050 acelerómetro y giróscopo Teoría y Practica	
Capítulo 48 TTP223 Sensor Táctil Capacitivo Digital Touch sensor	274
Capítulo 49 74Hc595 Registro de desplazamiento (Shift register)	279
Capítulo 50 Actualizar el IDE a la última versión estable (2022)	
Capítulo 51 IDE 2.0 Instalación y Completo análisis	292
Capítulo 52 Comunicación entre dos placas de Arduino	
Capítulo 53 i2c entre dos Arduino	303
Capítulo 54 Serial de datos entre 2 tarjetas Arduino en Tinkercad	307

Este tutorial está vasado en los videotutoriales de bitwise Ar, al comienzo de cada capítulo podrás acceder al tutorial y después encontrarás un resumen del mismo.

La introducción a la programación está realizado basándome en los tutoriales de Johann Perez E.





Capítulo 1.- Introducción Arduino

De la misma que nuestros sentidos llegan al cerebro para ser procesados y actuar en consecuencia la placa Arduino también los tiene.



Información de los sensores.



- Luz
- Sonido
- Orientación
- Temperatura
- Humedad
- ...
- LEDMotoresServosPantallas LCD
 - Bluetooth

Pare este tutorial vamos a trabajar con Arduino UNO.

Su facilidad en la programación lo hace asequible a cualquier persona.



Que iremos aprendiendo progresivamente y con varios ejemplos.

Está siendo cada vez más utilizada en el sistema educativo, para iniciar a los estudiantes en el mundo de la programación y la robótica.



Capítulo 2.- Instalación del IDE de Arduino y mi primer programa

Vamos a instalar el programa que necesitaremos para programar nuestro Arduino, accederemos al siguiente enlace:



HARDWARE	SOFTWARE	CLOUD	DOCUMENTATION -	COMMUNITY -	BLOG	ABOUT

Seleccionaremos Software.

Downloads	
OWNER Arduino IDE 2.0.0 The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger. For more details, please refer to the Arduino IDE 2.0 documentation. Nightly builds with the latest bugfixes are available through the section below. SOURCE CODE The Arduino IDE 2.0 is open source and its source code is hosted on	DOWNLOAD OPTIONS Windows Win 10 and newer, 64 bits Windows MSI installer Windows ZIP file Linux AppImage 64 bits (X86-64) Linux ZIP file 64 bits (X86-64) macOS 10.14: "Mojave" or newer, 64 bits

En nuestro caso yo como tengo Windows descargaré el que he seleccionado.



Si quieres puedes aportar una donación a la fundación si es lo que deseas, si no es así selecciona JUST DOWLOAD.



Ahora tienes que realizar el proceso de instalación, haciendo doble clic.

🥯 Instalación de Arduino IDE - 🗆 🗙
Acuerdo de licencia Por favor revise el acuerdo de licencia antes de instalar Arduino IDE.
Presione Avanzar Página para ver el resto del acuerdo.
Ferms of Service
The Arduino software is provided to you "as is" and we make no express or implied warranties whatsoever with respect to its functionality, operability, or use, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or infringement. We expressly disclaim any liability whatsoever for any direct, indirect, consequential, incidental or special damages, including, without limitation, lost revenues, lost profits, losses resulting from business interruption or loss of data, regardless of the form of action or legal theory under which the liability may be asserted, even if advised of the possibility or likelihood of such damages.
Si acepta todas las condiciones del acuerdo, seleccione Acepto para continuar. Debe aceptar el acuerdo para instalar Arduino IDE.
Arduino IDE 2,0,0
<u>A</u> cepto Cancelar

Hacemos clic en Acepto.

🔤 Instalación de Arduino IDE	_		×
Elegir opciones de instalación			_
¿Para quién se instalará esta aplicación?			$\overline{\circ}$
Elige si deseas que este software esté disponible para todos los usuar	ios o solo	para ti.	
O Cualquiera que utilice este ordenador (todos los usuarios)			
Cualquera que duice este ordenador (todos los usuarios)			
◯ Solo para mí. (pmver)			
Instalación nueva para todos los usuarios (se pedirán los credenciales administrador)	del		
Arduino IDE 2.0.0			
< <u>A</u> trás 😜 Sigu	uiente >	Can	celar

Aquí podemos seleccionar si solo lo utilizaré yo a todos los usuarios de este ordenador, yo he seleccionado cualquiera que utilice este ordenador, seguido del botón siguiente.

🤓 Instalación de Arduino IDE	_		×
Elegir lugar de instalación Elija la carpeta para instalar Arduino IDE.			∞
El programa de instalación instalará Arduino IDE en la siguiente carpeta. carpeta diferente, presione Examinar y seleccione otra carpeta. Presion la instalación.	Para ins e Instala	italar en ar para in	una iiciar
Carpeta de Destino C:\Program Files\Arduino IDE	<u>E</u> xam	inar	
Arduino IDE 2.0,0	lar	Cano	elar

Podemos cambiar la ubicación de donde queremos instalar el programa, lo dejaremos donde pone por defecto seguido del botón Instalar.

Empieza el proceso de instalación.



Activamos al casilla Ejecutar Arduino IDE. Seguido del botón Terminar.



Seleccionaremos INSTALL ALL

sketch_oct20a	🗟 sketch_oct20a Arduino IDE 2.0.0 – 🗆 X							
File Edit Sketch	Tools Help							
I I I I I I I I I I I I I I I I I I I	Select Board 👻		\checkmark	.∕©.				
sketch_o 1 2 3 4 5 6 7 9 10	<pre>t20a ino void setup() { // put your setup code here, to run once: } void loop() { // put your main code here, to run repeatedly: } </pre>							
Output			1	≣ 6				
Destan arduin Replax Desin: Platfo Desin: Confi Platfo	gamo paquetes ning platform arduino:avr@1.8.3 with arduino:avr@1.8.5 talando arduino:avr@1.8.3 uninstalled talando arduino:avr@c.207.3.0-atmel3.6.1-arduino7, la herramienta ya no es requerida stalando arduino:avrdude@6.3.0-arduino17, la herramienta ya no es requerida talando arduino:arduino014@1.3.0, la herramienta ya no es requerida gurando plataforma. orm arduino:ard@1.8.5 installed							
	Ln 1, Col 1 UTF-8 × No	board selec	ted 🗘					

Ya estamos listos para empezar a programar.

Del menú File vamos a seleccionar Preferences:

Preferences	Settings Network	×
Sketchbook location:		
c:\Users\pmver\OneDrive\	Documentos\Arduino	BROWSE
Show files inside Sketcl	les	
Editor font size:	14	
Interface scale:	✓ Automatic 100 %	
Theme:	Light (Arduino) 🗸	
Language:	English (Reload required)	
Show verbose output durin	g 🗌 compile 🗌 upload	
Compiler warnings	None 🗸	
O Verify code after upload		
Auto save		
Editor Quick Suggestion		
Additional boards manager	URLS.	B
		CANCEL

Lo podemos cambiar a español.

Si del menú Herramientas seleccionamos Puerto.

🔤 sketch_oct20a Arduino IDE 2.0.0 —		×
Archivo Edit Sketch Herramientas Help		
Auto Formato Ctrl + T	\checkmark	·Q··
Archivo de programa		
Sketch_oct20 Gestionar bibliotecas Ctrl + Mayús + I		
1 VO Monitor Serie Ctrl + Mayús + M		
B Plotter Serie		
4) placa		
5 Puerto Serial ports		
7 Obtener información de la placa COM3 (Arduino Uno)		
8 COM20		
9 } Actualizator de rimmwere de virito / virinina coM19		
Commission Commis		
Grabar Bootloader COM16		
COM17		
COM15		
Descargando el índice: library index ison gz		
Lin. 10. col. 1 UTF-8 × Ninguna placa selecc	ionada.	(1

Podremos observa si ha detectado nuestra placa Arduino.

🔤 sk	🥯 sketch_oct20a Arduino IDE 2.0.0				
Archiv	vo Edit Ske	tch Herramientas Help			
) 🔿 🚱	Seleccionar Placa 👻			
P	sketch_o	ct20a.ino			
	1	<pre>void setup() {</pre>			
_	2	<pre>// put your setup code here, to run once:</pre>			
1_)	3				
	4	}			
Inh	5				
	6	<pre>void loop() {</pre>			
	7	<pre>// put your main code here, to run repeatedly:</pre>			
	8				
~	9	}			
\sim	10				
Q					

Al empezar con un nuevo proyecto ya muestra la siguiente estructura.

void setup() {

}

Este bloque solo se ejecutará el al principio y solo una vez.

void loop() {

}

A continuación se ejecutará esta parte del código, que se reproducirá en modo de bucle sin parar, hasta que desconectemos la placa.

	📨 sketch_oct20a Arduino IDE 2.0.0				
	Archivo	Edit	Sketch Herramientas Help		
		•	Seleccionar Placa -		
	Ph	sketc	_oct20a.ino		
		1	<pre>void setup() {</pre>		
	_	2	<pre>pinMode(13, OUTPUT); // Para decirle que pin vamos a usar</pre>		
	t_)	З	// En este caso lo vamos a utilizar como salida.		
		4	}		
	Mb	5			
	ШИ	6	<pre>void loop() {</pre>		
		7	<pre>digitalWrite(13, HIGH); // Ahora al pin 13 le enviamos un valor alto.</pre>		
		8	<pre>delay(1000); // Ahora hacemos un proceso de espera de 1 segundo.</pre>		
	200	9	<pre>digitalWrite(13, LOW); // Ahora al pin 13 le enviamos un valor bajo.</pre>		
		10	<pre>delay(1000); // Otro proceso de espera de 1 segundo.</pre>		
	Q	11	}		
1		12			

Todo texto que agregamos después de // son comentarios que no afectan a la programación pero si ayuda al programador lo que realiza parte del código de programación.

Con el botón remarcado vamos a verificar que el programa está bien.



Tienes que tener seleccionada el tipo de placa Arduino con la que estás trabajando.



Ya podemos subir el programa a nuestra placa de Arduino.

Ahora si todo ha ido bien podremos ver como el LED conectado el pin 13 se enciende y se apaga con intervalos de 1 segundo, ya hemos hecho nuestro primer programa.

Monitor Serie

Este primer ejercicio vamos introducir nuestro nombre, y a partir de este nos salude.

🔤 ske	🧧 sketch_nov2a Arduino IDE 2.0.1 – 🗆 🗙					
Archivo	Editar SI	xetch Herramientas Ayuda				
	€	🖞 Arduino Uno 👻		৵	·Q··	
Ph	sketch_n	ov2a.ino				
	1	String Nombre;				
	2					
t_)	3	<pre>void setup() {</pre>				
	4	Serial.begin(9600);				
Irfk	5	<pre>Serial.println("Dime tu nombre: ");</pre>				
ШИ	6	<pre>while (Serial.available() == 0){</pre>				
	7					
	8	}				
2,4	9	Nombre = Serial.readString();				
\sim	10	Serial.print("Hola, ");				
Q	11	<pre>Serial.print(Nombre);</pre>				
	12	<pre>Serial.println(" espero que pases un buen dia.");</pre>				
	13	}				
	14					
	15	<pre>void loop() {</pre>				
	16					
	17	}				
	18					

Introduce tu nombre.



En el siguiente capítulo vamos a preguntar por nuestro nombre y nos diga el número de caracteres que tiene.



Vamos a realizar otro proyecto en que nos preguntará por nuestro nombre y tiene que imprimir cada letra en una línea distinta.



Ahora vamos a realizar un programa que nos pregunte por un número y a continuación imprima desde el número 1 hasta el que hemos introducido.





Vamos a realizar el siguiente circuito:



Si escribimos en el monitor serie Rojo, AMARILLO o VERDE que se encienda el led correspondiente y apague los demás leds.

Este será el código:

1 #define ROJO 4
2 #define AMARILLO 3
3 #define VERDE 2
4 String COLOR;

```
5 void setup()
6 {
7
    pinMode(ROJO, OUTPUT);
8
    pinMode (AMARILLO, OUTPUT);
    pinMode(VERDE, OUTPUT);
Serial.begin(9600);
9
10
11
    digitalWrite(ROJO, LOW);
12
    digitalWrite(AMARILLO, LOW);
13
     digitalWrite(VERDE, LOW);
14
     Serial.println("Dime un color ROJO, AMARILLO o VERDE:");
15 }
16
17 void loop()
18 {
19
    while(Serial.available() == 0) {
20
21
     }
22
    COLOR = Serial.readString();
23
    if (COLOR == "ROJO") {
24
       digitalWrite(ROJO, HIGH);
25
       digitalWrite(AMARILLO, LOW);
26
       digitalWrite(VERDE, LOW);
27
     }
    if (COLOR == "AMARILLO") {
28
29
      digitalWrite(ROJO, LOW);
30
       digitalWrite(AMARILLO, HIGH);
31
      digitalWrite(VERDE, LOW);
32
     }
33
    if (COLOR == "VERDE") {
34
       digitalWrite(ROJO, LOW);
35
       digitalWrite(AMARILLO, LOW);
36
       digitalWrite(VERDE, HIGH);
37
     -}
38 }
```



Capítulo 3.- Entradas/Salidas digitales con Pulsador y LED

Vamos a escribir el siguiente código:

🔤 sket	🧧 sketch_oct20a Arduino IDE 2.0.0 – 🗆 🗙					
Archivo	Edit Ske	tch Herramientas Help				
	€ 🗧	Seleccionar Placa 👻		\checkmark	. O .	
P	sketch_o	ct20a.ino				
	1	<pre>void setup() {</pre>				
	2	<pre>pinMode(2, INPUT);</pre>				
1-	3	<pre>pinMode(3, OUTPUT);</pre>				
	4	}				
Ink	5					
	6	void loop() {				
	7	if (digitalRead(2) == HIGH){				
	8	digitalWrite(3, HIGH);				
~	9	}				
	10	else{				
Q	11	digitalWrite(3, LOW);				
	12	}				
	13	}				

El pin 2 lo configuramos como pin de entrada.

El pin 2 lo configuramos como pin de salida.

Cuando empieza el bucle.

Si el pin 2 recibe un nivel alto, pues enciende el led que esta en el pin 3

Sino apaga el led que está en el pin 3.

Ahora cuando presiones el interruptor observarás como el led se enciende y cuando lo sueltas el led se apaga.

Ahora sin modificar el circuito vamos a realizar otro proyecto, que tal si al presionar el interruptor, si el led está apagado este se encienda y de lo contrario si este está encendido que se apague.



Ahora sin modificar el circuito queremos que al presionar el pulsador el led se encienda durante 5 segundo y se vuelva a apagar.

💿 sket	sketch_oct20a Arduino IDE 2.0.0				
Archivo	Edit Sketch Herramientas Help				
\bigcirc	→ 🔂 🖞 Arduino Uno 👻				
P	sketch_oct20a.ino				
	<pre>1 int PULSADOR = 2;</pre>				
	2 int LED = 3;				

```
1월)
             int ESTADO = LOW;
        3
        4
        5
            void setup() {
llh
        6
             pinMode(PULSADOR, INPUT);
        7
             pinMode(LED, OUTPUT);
÷
        8
             }
        9
       10
             void loop() {
               while(digitalRead(PULSADOR) == LOW){
       11
               // Mientras no presionemos el pulsador que no haga nada
       12
       13
               }
              digitalWrite(LED, HIGH); // Enciende el Led
       14
              delay(5000);
                                      // Espera 5 segundos
       15
              digitalWrite(LED, LOW); // Apaga el Led
       16
       17
               while(digitalRead(PULSADOR) == HIGH){
               // Este bucle evita los famosos llamados rebotes.
       18
               }
       19
       20
```

Sentencia if



Partiendo de este circuito, la primera práctica consiste en que si voltaje que circula es mayor de 4 voltios el led se tiene que encender y si no es así que el led se apague.

```
1 int Poten = A0;
 2 int Led = 12;
 3 float VLectura;
   float V;
 4
 5
 6 void setup()
7
   -{
     pinMode(Led, OUTPUT);
8
 Q,
     Serial.begin(9600);
10 }
11
12 void loop()
13
   -{
14
     VLectura = analogRead(A0);
15
     V = VLectura * 5.0 /1023.0;
16
     Serial.println(V);
17
     if (V > 4.0) {
18
       digitalWrite(Led, HIGH);
19
     }
20
     if (V <= 4.0) {
21
       digitalWrite(Led, LOW);
22
     }
23
     delay(100);
24
   }
```

En la siguiente práctica queremos que nuestro led se encienda cunado se encuentre entre 2 voltios y 4 voltios, de lo contrario que el led se apague.

```
1 int Poten = A0;
2 int Led = 12;
3 float VLectura;
4 float V;
5
6 void setup()
7
   -{
8
    pinMode(Led, OUTPUT);
9
     Serial.begin(9600);
10 }
11
12 void loop()
13 {
14
     VLectura = analogRead(A0);
     V = VLectura * 5.0 /1023.0;
15
16
     Serial.println(V);
17
     if (V >=2.0 && V<= 4.0) {
18
       digitalWrite(Led, HIGH);
19
     }
20
     if (V<2.0 || V>4.0){
       digitalWrite(Led, LOW);
21
22
     }
23
     delay(100);
24 }
```

En esta ultima práctica vamos a modificar el diseño.

else

Partiendo del siguiente circuito:



Queremos que si el voltaje está entre 0.0 a 3.0 voltios que se encienda el led verde, si se encuentra entre 3.1 a 4.0 que se encienda el led amarillo y si se encuentra entre 4.1 a 5.0 que se encienda el led rojo, los demás led que no cumplan dicha condición han de permanecer apagados.

```
int Poten = A0;
int LedRojo = 12;
int LedAmarillo = 11;
int LedVerde = 10;
float VLectura;
float V;
void setup()
{
    pinMode(LedRojo, OUTPUT);
    pinMode(LedAmarillo, OUTPUT);
    pinMode(LedVerde, OUTPUT);
    Serial.begin(9600);
}
```

```
void loop()
{
VLectura = analogRead(A0);
V = VLectura * 5.0 /1023.0;
Serial.println(V);
if(V<=3.0){
 digitalWrite(LedVerde, HIGH);
 }else{
  digitalWrite(LedAmarillo, LOW);
 digitalWrite(LedRojo, LOW);
}
if (V >=3.1 && V<= 4.0){
  digitalWrite(LedAmarillo, HIGH);
 }else{
  digitalWrite(LedVerde, LOW);
  digitalWrite(LedRojo, LOW);
}
```

```
initialWrite(LedKojo, LOW),
}
if (V>=4.1){
    digitalWrite(LedRojo, HIGH);
}else{
    digitalWrite(LedVerde, LOW);
    digitalWrite(LedAmarillo, LOW);
}
delay(100);
```

```
}
```

Resistencia pull-down y pull-up









for <u>Práctica:</u>

Realiza el siguiente circuito:



Ahora vamos a escribir el siguiente código:

```
1
   int TIEMPO = 200;
2
   void setup()
3
   -{
4
     for (int i = 2; i < 7; i++)
5
     - {
6
           pinMode(i, OUTPUT);
7
     }
8
   }
9
10 void loop()
11 {
12
    for (int i = 2; i < 7; i++)
13
14
    digitalWrite(i, HIGH);
15
    delay(TIEMPO); // Wait for 1000 millisecond(s)
16
     digitalWrite(i, LOW);
17
     delay(TIEMPO); // Wait for 1000 millisecond(s)
18
19
     for (int i = 5; i > 2; i--)
20
21
     digitalWrite(i, HIGH);
22
     delay(TIEMPO); // Wait for 1000 millisecond(s)
23
     digitalWrite(i, LOW);
24
     delay(TIEMPO); // Wait for 1000 millisecond(s)
25
     }
26 }
```

Para entender bien este programa busca información sobre el ciclo for.

while



Una vez realizado el correspondiente circuito, queremos que cuando el potenciómetro de una señal superior a 1000 el led empiece a parpadear, y si estamos por debajo de este valor el led deja de parpadear.

```
1
   int Poten = A0;
2 int led = 3;
3 int ValorLeido;
4
5 void setup()
6
   {
7
     pinMode(led, OUTPUT);
8
     Serial.begin(9600);
9 }
10
11 void loop()
12
   {
13
     ValorLeido = analogRead(Poten);
    while (ValorLeido > 1000) {
14
15
       Serial.println(ValorLeido);
16
       analogWrite(led, HIGH);
17
       delay(200);
18
       analogWrite(led, LOW);
19
       delay(200);
20
       ValorLeido = analogRead(Poten);
21
     }
22
     analogWrite(led, LOW);
23 }
```

switch

En este ejemplo vamos a utilizar la sentencia switch que es como un conjunto de if pero solo aceptar números enteros para este capítulo vamos a crear un circuito con 5 leds y estos los vamos a numerar de izquierda a derecha con los números 1, 2, 3, 4 y 5, por mediación del monitor serial le vamos a decir que número de led queremos encender, si introducimos un número mayor de 5 que nos diga que este led no existe.



Vamos a escribir el siguiente código:

```
int led1 = 6;
int led2 = 5;
int led3 = 4;
int led4 = 3;
int led5 = 2;
int NumLed;
void setup()
{
  for (int i=2 ; i < 7; i++){
    pinMode(i, OUTPUT);
    }
    Serial.begin(9600);
}
```

void loop()

```
{
  Serial.println("Introduce el numero de led a encender");
  while (Serial.available() == 0){
```

}

```
NumLed = Serial.parseInt();
switch(NumLed){
 case 1:
      digitalWrite(led1, HIGH);
      digitalWrite(led2, LOW);
      digitalWrite(led3, LOW);
       digitalWrite(led4, LOW);
      digitalWrite(led5, LOW);
      break:
 case 2:
      digitalWrite(led1, LOW);
      digitalWrite(led2, HIGH);
      digitalWrite(led3, LOW);
      digitalWrite(led4, LOW);
      digitalWrite(led5, LOW);
      break;
 case 3:
      digitalWrite(led1, LOW);
      digitalWrite(led2, LOW);
       digitalWrite(led3, HIGH);
      digitalWrite(led4, LOW);
      digitalWrite(led5, LOW);
      break;
 case 4:
      digitalWrite(led1, LOW);
      digitalWrite(led2, LOW);
      digitalWrite(led3, LOW);
      digitalWrite(led4, HIGH);
      digitalWrite(led5, LOW);
      break;
 case 5:
      digitalWrite(led1, LOW);
      digitalWrite(led2, LOW);
      digitalWrite(led3, LOW);
      digitalWrite(led4, LOW);
      digitalWrite(led5, HIGH);
      break;
 default:
      digitalWrite(led1, LOW);
      digitalWrite(led2, LOW);
      digitalWrite(led3, LOW);
      digitalWrite(led4, LOW);
```

```
digitalWrite(led5, LOW);
```

Serial.println("El numero de led introducido no es correcto");

} }

Vamos a contar en binario, para ello vamos a mostrar la siguiente tabla:

	8	4	3	1	_		_	
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	
2	0	0	1	0	0	0		0
3	0	0	1	1	0	0		
4	0	1	0	0	0		0	0
5	0	1	0	1	0		0	
6	0	1	1	0	0			0
7	0	1	1	1	0			
8	1	0	0	0		0	0	0
9	1	0	0	1		0	0	
10	1	0	1	0		0		0
11	1	0	1	1		0		
12	1	1	0	0			0	0
13	1	1	0	1			0	
14	1	1	1	0				0
15	1	1	1	1				

Vamos a diseñar el siguiente circuito:



```
Este será el código:
```

```
int led1 = 9;
int led2 = 10;
int led3 = 11;
int led4 = 12;
int contador = 0;
int valor = 0;
void setup()
{
 pinMode(led1, OUTPUT);
 pinMode(led2, OUTPUT);
 pinMode(led3, OUTPUT);
 pinMode(led4, OUTPUT);
}
void loop()
{
 while (contador <= 15){
  valor = contador;
  if (valor \geq 8){
   digitalWrite(led4, HIGH);
   valor = valor - 8;
  }else{
   digitalWrite(led4, LOW);
  }
  if (valor >= 4){
   digitalWrite(led3, HIGH);
   valor = valor - 4;
  }else{
   digitalWrite(led3, LOW);
  }
  if (valor \geq 2)
   digitalWrite(led2, HIGH);
   valor = valor - 2;
  }else{
   digitalWrite(led2, LOW);
  }
  if (valor \geq 1)
   digitalWrite(led1, HIGH);
   valor = valor - 1;
  }else{
   digitalWrite(led1, LOW);
  }
  contador ++;
        delay(1000); // Wait for 1000 millisecond(s)
  }
 contador = 0;
}
```



Antes de continuar con el proyecto vamos a explicar los pines digitales PWM $\tilde{~}$

En un pin digital solo hay dos valores.





Ponemos un valor bajo, esperamos por un tiempo de 20 milisegundos pasamos de a un valor alto durante 20 milisegundos y así sucesivamente.

En el gráfico de la izquierda se muestra que tiene un valor bajo por 20 ms y a continuación un valor alto por 20 ms., que se irá repitiendo.

El Período de la señal representado con la T es igual a 40 milisegundos.

Esto es igual a la Frecuencia que se representa con la letra f, que se mide en segundos.

f = 1/T = 1/0,04 seg.

f = 25 Hz (Herts)



Hertz = ciclos por Segundo.

El tiempo donde la señal está activa en tal periodo.

En este ejemplo es de un 50% que se define con la letra D.



Aquí tenemos tres ejemplos donde le primero de la izquierda D = 20% de señal activa, el del centro un 50% y el de la derecha un 80%.

Si sabemos que la señal alta son 5 voltios en el gráfico de la izquierda que es de un 20% será un total de 1 voltio.

En el gráfico del centro como vemos que D = 50% será un total de 2.5 voltios.

En cambio en el gráfico de la derecha la señal D = 80% será un total de 4 voltios.



Solo los pines 3, 5, 6, 9, 10 y 11 tienen esta propiedad.

Ahora vamos a ver un ejemplo de código:

```
int LED = 3;
1
2
3
    void setup() {
4
    pinMode(LED, OUTPUT);
5
    }
6
7
    void loop() {
8
     analogWrite(LED, 0-255)
9
0
```

Al hacer una lectura una escritura del pin 3, este podrá tener valor entre 0 y 255, siendo 0, 0 voltios y 255 para 5 Voltios. Si queremos que sea 2,5 voltios tendremos que poner 127.

Ahora con ayuda de la sentencia for vamos a hacer que nuestro led aumente su brillo y lo disminuya.

🤓 sketch_oct20a Arduino IDE 2.0.0					
Archivo	Edit Ske	tch Herramientas Help			
	€ 🗧	Arduino Uno 👻			
	sketch_o	ct20a.ino			
	1	int LED = 3;			
	2	int BRILLO;			
1-)	3				
	4	<pre>void setup() {</pre>			
Ith	5	<pre>pinMode(LED, OUTPUT);</pre>			
	6	}			
	7				
	8	<pre>void loop() {</pre>			
~	9	<pre>for(BRILLO = 0; BRILLO < 256; BRILLO++){</pre>			
	10	analogWrite(LED, BRILLO);			
Q	11	delay(10);			
	12	}			
	13	<pre>for(BRILLO = 255; BRILLO >=0; BRILLO){</pre>			
	14	analogWrite(LED, BRILLO);			
	15	delay(10);			
	16	}			
	17	}			

El bucle for hace que la variable BRILLO empiece asumiendo el valor de 0, (LED apagado) en la siguiente vuelta el 1 y así sucesivamente hasta llevar al valor 255 (LED totalmente encendido).

Con lo cual el LED asume los valores 0, 1, 2..., 255 que irá aumentando el brillo.

El siguiente bucle empieza con BRILLO igual a 255 y termina con el valor O decrementado cada ve 1.

Con lo cual el LED asume los valores 255, 254, 253,..., 0 de este modo el LED disminuirá su brillo hasta apagarse totalmente.

🥯 sketch_oct20a Arduino IDE 2.0.0								
Archivo Edit Sketch Herramientas Help								
	€ 🕞	🖞 Arduino Uno 👻						
En l	sketch_oo	xt20a.ino						
	1	<pre>int LED = 3;</pre>						
	2	int BRILLO;						
12	3	int POT = 0;						
	4							
mk	5	<pre>void setup() {</pre>						
ШИ	6	pinMode(LED, OUTPUT);						
	7	// Las entradas analógicas no requieren inicialización						
	8	ł						
27	9							
	10	void loop() {						
Q	11	BRILLO = analogRead(POT) / 4;						
	12	analogWrite(LED, BRILLO);						
	13	}						

Ahora verás que girando el potenciómetro modificarás el brillo del led.


```
1 int pin = 6;
2
 3 void setup()
4 {
 5
     pinMode(pin, OUTPUT);
 6 }
 7
 8 void loop()
9 {
10
    analogWrite(pin, 0);
    delay(100);
analogWrite(pin, 125);
delay(100);
11
12
13
14
    analogWrite(pin, 255);
15
     delay(100);
16 }
```



map

con esta función vamos a adaptar valore que leemos en la salida analógica AO que son valores entre O y 1023 a valores de entre O y 255 que son los valores que puede escribir en un pin digital en PWM.



```
1 int analog = A0;
 2 int led = 3;
 3 int brillo;
 4 int Valor;
 5
 6 void setup()
 7
   {
    pinMode(led, OUTPUT);
 8
9
     Serial.begin(9600);
10 }
11
12 void loop()
13
   {
14
     Valor = analogRead(analog);
15
     brillo = map(Valor, 0, 1023, 0, 255);
     analogWrite(led, brillo);
16
17
     Serial.println(brillo);
18 }
```

Capítulo 5.- HC- SR04 Sensor Ultrasónico



Usa hondas de ultrasonido para detectar la distancia de un objeto, el sonido que emite no lo podemos escuchar ya que tiene una frecuencia de 40 Khz. Prácticamente el doble de la frecuencia máxima que el oído humano puede percibir.

Vamos a realizar el correspondiente circuito:



Este será el código:



Inicializamos la variable TRIG con el valor 10 que es el pin al que está conectado.

La variable ECO con el valor 9 que es el pin al que está conectado.

La variable LED con el pin que está conectado.

Las variables DURACIÓN y DISTANCIA no le asignamos ningún valor, se le asignará durante la ejecución del programa.

En el void setup() le decimos que el pin con el valor de la variable TRIG es de salida. El pin con el valor de la variable ECO es de entrada, así como el pin de la variable LED.

Para mantener una comunicación con la consola tendremos que poner Serial.begin(9600).

Con el bucle void loop() con digitalWrite(TRIG, HIGH) estamos mandando al pin 10 una corriente alta.

Esperamos 1 milisegundo.

Ahora con digitalWrite(TRIG, LOW) estamos mandando al pin 10 una corriente baja.

A la variable DURACIÓN le pasamos la siguiente formula pulseIn(ECO, HIGH), le asignamos el tiempo que ha recibido la señal.

Con la variable DISTANCIA y con la siguiente formula DURACION / 58.2 (Este valor suministrado por el fabricante) le asignaremos la distancia en cms. del sensor a un objeto.

Con delay(200) hacemos una demora de 200 milisegundos.

Ejecutaremos la consola para poder ver los valores.

🔤 sketch_oct20a Arduino IDE 2.0.0 — 🗆 🗙						
Archivo Edit Sketch Herramientas Help						
Arduino Uno -		√	۰Q۰			
sketch_oct20a.ino						
1 int TRIG = 10:						

Si muestra algún valor negativo, es normal indica que el sensor está fuera de rango, porque el objeto está muy lejos o por que el objeto no recibe el rebote ultrasónico.

Ahora queremos que el LED se encienda cuando detecte una distancia mayor de 0 y menor de 20 cms.



```
24 else {

25 digitalWrite(LED, LOW);

26 }

27 delay(200);

28 }
```

Ahora cuando lo ejecutes ves acercando un objeto y alejándolo.

&& es la condición que compara más de una condición y se han de cumplir todas.

Ahora queremos que el Led realice un parpadeo y cuanto mas cerca esté el objeto el parpadeo será más rápido.



Vamos a modificar el circuito.



Capítulo 6.- Servomotor (conexión, modelos, ajustes para uso óptimo)





Permite un movimiento de 0 a 180 grados.

Vamos a diseñar el siguiente circuito:



Vamos a escribir el código:

🔤 Capitulo6 Arduino IDE 2.0.0							
Archivo	Edit Sket	tch Herramientas Help					
\bigcirc	€ 🕞	Seleccionar Placa -					
Ph	Capitulo6	ino					
	1	<pre>#include <servo.h></servo.h></pre>					
	2						
1-)	3	Servo servo1;					
	4						
Inh	5	<pre>int PINSERVO = 2;</pre>					
	6	int PULSOMIN = 1000;					
	7	int PULSOMAX = 2000;					
	8						
~	9	<pre>void setup() {</pre>					
	10	<pre>servo1.attach(PINSERVO, PULSOMIN, PULSOMAX);</pre>					
Q	11	}					
	12						

```
12
13 void loop() {
14 servo1.write(0);
15 delay(5000);
16 servo1.wsrite(180);
17 delay(5000);
18 }
```

Para poder manejar el servo tenemos que incluir la siguiente librería.

#include <Servo.h>

Creamos un objeto con esta librería: Servo servo1;

Definimos la variable PINSERVO = 2 en el pin que está conectado el servo.

Definimos las variables PULSOMIN = 1000 y PULSOMAX = 2000, valores en milisegundos.

En el void setup() configuramos el servo1.attach(3, 1000, 2000), pero en lugar de los valores están sus variables.

Ahora en el bucle void loop() Para moverse a la posición de 0 grados servo1.write(0), esperamos 5 segundos, nos movemos a la posición de 180 grados y de nuevo esperamos 5 segundos, y se irá repitiendo.

Como vemos que no realiza un giro de 180 grados hemos modificado los siguientes parámetros:

6 int PULSOMIN = 500; 7 int PULSOMAX = 2500;

Esto lo hemos realizado progresivamente.

Ahora vamos a modificar el programa para poderlo controlar con el Potenciómetro.

🔤 sket	🔤 sketch_oct20a Arduino IDE 2.0.0								
Archivo	Edit Ske	tch Herramientas Help							
	€ 🗧	🜵 Arduino Uno 👻							
Ph	sketch_o	ct20a.ino							
	1	<pre>#include <servo.h></servo.h></pre>							
	2								
1 <u>-</u>)	3	Servo servo1;							
	4								
Infly	5	<pre>int PINSERVO = 2;</pre>							
	6	<pre>int PULSOMIN = 500;</pre>							
	7	<pre>int PULSOMAX = 2500;</pre>							
	8	int VALORPOT;							
~	9	int ANGULO;							
	10	<pre>int POT = 0;</pre>							
Q	11								
	12	<pre>void setup() {</pre>							
	13	<pre>servo1.attach(PINSERVO, PULSOMIN, PULSOMAX);</pre>							
	14	// Las entradas analóginas no requieren de iniciliazación.							
	15	}							
	16								
	17	void loop() {							
	18	VALORPOT = analogRead(POT);							
	19	ANGULO = $map(VALORPOT, 0, 1023, 0, 180);$							
	20	<pre>servol.write(ANGULO);</pre>							
	21	delay(20);							
	22	}							

Hemos realizado las siguientes modificaciones:

Hemos definido las variables VALORPOT, ANGULO sin asignarle ningún valor.

La variable POT a 0 ya que haremos referencia al pin analógico A0.

En el void loop() a la variable VALORPOT le asignamos el valor que lee desde el pin A0.

El valor que puede obtener es de 0 a 1023 y lo queremos transformar a un valor de 0 a 180 para esto se utiliza la función map, y se lo asignamos a la variable ANGULO.

Con Servo1.write(ANGULO) desplazamos al servo a la posición de la variable ANGULO.

Hace una espera de 20 milisegundo y volvemos a empezar el bucle.

Capítulo 7.- Temperatura y Humedad ambiente con DHT22 o



Vamos a realizar el circuito:



Cuando instalamos este programa no incluye las librerías del sensor HDT11.





Buscaremos por DHT11.

Hacemos clic en el botón instalar.

Ahora tenemos que buscar Adadruit Unified Sensor y también instlarlo.



Ahora la tenemos que incluir.

Vamos al código:

```
1
     #include <TinyDHT.h>
 2
     #include <Adafruit_Sensor.h>
 3
 4
    int SENSOR = 2;
 5
     int TEMPERATURA;
 6
     int HUMEDAD;
 7
 8
     DHT dht(SENSOR, DHT11);
9
10
11
     void setup() {
       Serial.begin(9600);
12
13
       dht.begin();
14
15
     }
16
```

```
17
     void loop() {
18
       TEMPERATURA = dht.readTemperature();
19
       HUMEDAD = dht.readHumidity();
       Serial.print("Temperatura: ");
20
       Serial.print(TEMPERATURA);
21
       Serial.print(" Humedad: ");
22
       Serial.println(HUMEDAD);
23
24
     }
```

Si ahora miramos a la consola:



Nos muestra la temperatura y la humedad.

Salida Monito	Serie ×	
Mensaje (Ctrl +	nter para enviar el mensaje a 'Arduino Uno' en 'COM3')	
Temperatura:	4 Humedad: 95	
Temperatura:	5 Humedad: 95	
Temperatura:	5 Humedad: 95	
Temperatura:	5 Humedad: 95	



Diagrama eléctrico / Esquemático (simplificado)





Vamos a diseñar el circuito:



Ahora vamos a programarlo:



ſ	nık	5	}
Ŀ	ШИ	6	
Ŀ		7	<pre>void loop() {</pre>
Ŀ		8	<pre>digitalWrite(RELE, LOW);</pre>
Ŀ	202	9	delay(5000);
Ŀ		10	<pre>digitalWrite(RELE, HIGH);</pre>
Ŀ	Q	11	delay(5000);
L		12	}

Podemos escuchar el ruido que hace la bobina al abrir y cerrar el circuito.

Posibles usos:





Capítulo 9.- Teclado Keypad 4x4 y simple control de acceso con clave



Este tipo de distribución se les denomina matricial.

Vamos a conectarlo:



Vamos a programar:

Para evitar el problema de rebotes vamos a descargar unas librerías.

Seleccionaremos Sketch de este Incluir biblioteca y de este Gestionar bibliotecas.

Buscaremos por keypad.

Keypad de Mark Stanley, Alexander Brevig As of version 3.0 it now supports mulitple keypresses. This library is based upon the Keypad Tutorial. It was created to promote Hardware Abstraction. It improves readability of the code by hiding the pinMode and digitalRead calls for the user. Keypad is a library for using matrix style keypads with the Arduino. Más información

Tenemos que instalar este.



En este proyecto vamos a trabajar con Array[].



Si queremos modificar un valor de la Array.

```
1
     #include <Key.h>
2
     #include <Keypad.h>
3
4
     const byte FILAS = 4;
5
     const byte COLUMNAS = 4;
6
     char keys[FILAS][COLUMNAS] = {
       {'1','2','3','A'},
{'4','5','6','B'},
7
8
     {'7','8','9'.'C'},
{'*','0','#','D'},
9
10
11
     };
12
     byte pinesFilas[FILAS] = {9,8,7,6};
13
     byte pinesColumnas[COLUMNAS] = {5,4,3,2,};
14
15
     keypad teclado = keypad(makekeymap(keys), pinesFilas, pinesColumnas, FILAS, COLUMNAS);
```

Cada vez que utilicemos este tipo de teclado este código habrá que escribirlo.

```
#include <Key.h>
#include <Keypad.h>
const byte FILAS = 4;
const byte COLUMNAS = 4;
char keys[FILAS][COLUMNAS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  { '*', '0', '#', 'D' },
};
byte pinesFilas[FILAS] = {9,8,7,6};
byte pinesColumnas[COLUMNAS] = {5,4,3,2,};
Keypad teclado = Keypad(makeKeymap(keys), pinesFilas, pinesColumnas,
FILAS, COLUMNAS);
char TECLA;
char CLAVE[7];
char CLAVE MAESTRA[7] = "123456";
byte INDICE = 0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  TECLA = teclado.getKey();
  if (TECLA) {
    CLAVE[INDICE] = TECLA;
    INDICE++;
    Serial.print(TECLA);
```

```
}
if (INDICE == 6){
    if(!strcmp(CLAVE, CLAVE_MAESTRA))
    Serial.println(" Correcta");
    else
    Serial.println(" Incorrecta");
    INDICE = 0;
    }
}
```

Este será el resultado en la consola:

Salida	Monitor Serie ×
Mensaje	e (Ctrl + Enter para enviar el mensaje a 'Arduino Uno' en 'COM3')
102456	Correcto
654321	Incorrecta



Capítulo 10.- Módulo LCD 1602A con librería LiquidCrystal

La ubicación de los caracteres empiezan 0,0 y termina 1,15.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1																

Las conexiones que están seleccionadas de color azul no las vamos a utilizar.

El pin 3 irá conectado a un Potenciómetro, para controlar el brillo.

Podemos soldar estos pines.



Este será el circuito:



Vamos a programar:

🥯 sketch_oct21a Arduino IDE 2.0.0						
Archivo	Edit Sket	ch Herramientas Help				
	€ 🕞	🖞 Arduino Uno 👻				
Ph	sketch_oc	t21a.ino				
	1	<pre>#include <liquidcrystal.h></liquidcrystal.h></pre>				
_	2					
1_)	3	LiquidCrystal <pre>lcd(7,6,5,4,3,2);</pre>				
	4					
Mh	5	<pre>void setup() {</pre>				
ШИ	6	<pre>lcd.begin(16,2);</pre>				
	7	}				
	8					
200	9	<pre>void loop() {</pre>				
	10	<pre>lcd.setCursor(0,0);</pre>				
Q	11	<pre>lcd.print("Hola, han pasado");</pre>				
	12	<pre>lcd.setCursor(0,1);</pre>				
	13	<pre>lcd.print(millis() / 1000);</pre>				
	14	<pre>lcd.print(" seg.");</pre>				
	15	}				

Incluimos la librería LiquidCristal.h.

En la línea 3 a que pines están conectados.

En la línea 6 decimos que dimensiones tiene el LCD.

Desde la línea 10 hasta la 14 trabajamos con la posición del cursor e imprimir texto.



Ahora queremos que el texto parpadee a modo de llamar la atención.

💿 ske	🥯 sketch_oct21a Arduino IDE 2.0.0						
Archivo	Edit Sket	ch Herramientas Help					
Ø	€ 🔊	🜵 Arduino Uno 👻					
	sketch_oc	t21a.ino					
	1	<pre>#include <liquidcrystal.h></liquidcrystal.h></pre>					
5	2						
	3	LiquidCrystal <pre>lcd(7,6,5,4,3,2);</pre>					
D-0.	5	<pre>void setup() {</pre>					
	6	<pre>lcd.begin(16,2);</pre>					
	7	}					
l ₽	8						
	9	Void 100p() {					
	10	<pre>lcd.setCursor(0,0); lcd.print("Alapma active !");</pre>					
X	12	<pre>lcd_display():</pre>					
	13	delay(500):					
	14	<pre>lcd.noDisplay();</pre>					
	15	delay(500);					
	16	}					

Otra modificación es mostrar el cursor, es útil cuando quiere demostrar que está esperando que se ingrese algún dato.

💿 ske	sketch_oct21a Arduino IDE 2.0.0						
Archivo	Edit Ske	tch Herramientas Help					
	€ 🕞	🖞 Arduino Uno 👻					
P	sketch_o	ct21a.ino					
	1	<pre>#include <liquidcrystal.h></liquidcrystal.h></pre>					
-	2						
1	3	LiquidCrystal <pre>lcd(7,6,5,4,3,2);</pre>					
	4	<pre>void setup() {</pre>					
	6	lcd.begin(16,2);					
	7	}					
da ∣	8						
	9	void loop() {					
0	10	<pre>lcd.print("Ingrese clave:"):</pre>					
\sim	12	<pre>lcd.setCursor(0,1);</pre>					
	13	<pre>lcd.cursor();</pre>					
	14	<pre>delay(500);</pre>					
	15	<pre>lcd.noCursor();</pre>					
	16	delay(500);					
	17	}					

Cuando queremos utilizar un texto que tenga más de 16 caracteres.

💿 sket	🤓 sketch_oct21a Arduino IDE 2.0.0						
Archivo	Edit Sket	ch Herramientas Help					
	€ 🕞	🖞 Arduino Uno 👻					
P	sketch_oc	t21a.ino					
	1	<pre>#include <liquidcrystal.h></liquidcrystal.h></pre>					
	2						
	3	LiquidCrystal <pre>lcd(7,6,5,4,3,2);</pre>					
	4						
Irfk	5	<pre>void setup() {</pre>					
ши	6	<pre>lcd.begin(16,2);</pre>					
	7	}					
	8						
200	9	<pre>void loop() {</pre>					
	10	<pre>lcd.setCursor(2,0);</pre>					
Q	11	<pre>lcd.print("Prueba scroll");</pre>					
	12	<pre>lcd.scrollDisplayLeft();</pre>					
	13	delay(1000);					
	14	}					

Capítulo 11.- PIR Sensor de movimiento Infrarrojo Pasivo HC-SR501 Ver video



Este es el circuito:



Vamos a programar:

应 Rele	🔤 Rele Arduino IDE 2.0.0							
Archivo	Edit Sket	ch Herramientas Help						
	€ 🔊	Arduino Uno 👻						
Ph	Rele.ino							
	1	int PIR = 2;						
	2	<pre>int RELE = 3;</pre>						
t_)	3	<pre>int ESTADO = 0;</pre>						
	4							
Mk	5	<pre>void setup() {</pre>						
	6	<pre>pinMode(PIR, INPUT);</pre>						
	7	<pre>pinMode(RELE, OUTPUT);</pre>						
	8	delay(20000);						
~	9	}						
	10							
Q	11							
	12	<pre>void loop() {</pre>						
	13	ESTADO = digitalRead(PIR);						
	14	if (ESTADO == HIGH) {						
	15	<pre>digitalWrite(RELE, LOW);</pre>						
	16	delay(4000);						
	17	} else {						
	18	<pre>digitalWrite(RELE, HIGH);</pre>						
	19	}						
	20	}						



Cuando detecta movimiento se enciende una luz.



Cuando detecta movimiento suena una alarma.



Cátodo común en los tres casos:



Este es el diseño:



Vamos al código:

```
1
     int LEDROJO = 9;
 2
     int LEDVERDE = 10;
 3
     int LEDAZUL = 11;
 4
 5
 6
     void setup() {
7
       pinMode(LEDROJO, OUTPUT);
8
       pinMode(LEDVERDE, OUTPUT);
9
       pinMode(LEDAZUL, OUTPUT);
10
     }
11
     void loop() {
12
13
       analogWrite(LEDROJO, 255);
14
       analogWrite(LEDVERDE, 0);
       analogWrite(LEDAZUL, 0);
15
16
```

Nos muestra el LED RGB de color rojo.

```
void loop() {
    analogWrite(LEDROJO, 0);
    analogWrite(LEDVERDE, 255);
    analogWrite(LEDAZUL, 0);
}
```

Nos muestra el LED RGB de color verde.

```
void loop() {
    analogWrite(LEDROJO, 0);
    analogWrite(LEDVERDE, 0);
    analogWrite(LEDAZUL, 255);
}
```

Nos muestra el LED RGB de color azul.

🤓 sketch_oct21a Arduino IDE 2.0.0					
Archivo Edit Sketch Herramientas Help					
	→ ↓ Arduino Uno				
Ph	sketch_oct21a.ino				
	<pre>1 int LEDROJO = 9;</pre>				
	<pre>2 int LEDVERDE = 10;</pre>				
t_)	<pre>3 int LEDAZUL = 11;</pre>				
	4				
Elk	5				
	<pre>6 void setup() {</pre>				
	<pre>7 pinMode(LEDROJO, OUTPUT);</pre>				
	<pre>8 pinMode(LEDVERDE, OUTPUT);</pre>				
200	<pre>9 pinMode(LEDAZUL, OUTPUT);</pre>				
	10 }				
Q	11				
	12 void loop() {				
	<pre>13 analogWrite(LEDROJO, 255);</pre>				
	<pre>14 analogWrite(LEDVERDE, 0);</pre>				
	<pre>15 analogWrite(LEDAZUL, 0);</pre>				
	16 delay(2000);				
	1/ analogwrite(LEDKOJO, 0);				
	<pre>18 analogwrite(LEDVERDE, 255); 40 analogwrite(LEDVERDE, 255);</pre>				
	19 analogwrite(LEDAZOL, 0);				
	20 ueray(2000), 21 analogWpite(LEDROJO θ):				
	22 analogwrite(LEDVERDE 0)				
	23 analogwrite(LEDA7UL 255)				
	24 delay(2000):				
	25 }				

Cada 2 segundos va cambiando de color.



En un editor de imágenes podemos jugar con todos los colores.

	Rojo	Verde	Azul
Amarillo	255	255	0
Rosa	255	0	255
Morado	178	102	255
Naranja	255	128	0

13	<pre>analogWrite(LEDROJO, 255);</pre>
14	analogWrite(LEDVERDE, 255);
15	<pre>analogWrite(LEDAZUL, 0);</pre>
16	delay(2000);
17	<pre>analogWrite(LEDROJO, 255);</pre>
18	<pre>analogWrite(LEDVERDE, 0);</pre>
19	<pre>analogWrite(LEDAZUL, 255);</pre>
20	delay(2000);
21	<pre>analogWrite(LEDROJO, 178);</pre>
22	<pre>analogWrite(LEDVERDE, 102);</pre>
23	<pre>analogWrite(LEDAZUL, 255);</pre>
24	delay(2000);
25	analogWrite(LEDROJO, 255);
26	<pre>analogWrite(LEDVERDE, 128);</pre>
27	analogWrite(LEDAZUL, 0);
28	delay(2000);

Ahora vamos a crear un función para ir creando colores.



De este modo evitamos código repetitivo. Te propongo la siguiente práctica:



Este es el código:

🥯 sketch_oct21a Arduino IDE 2.0.0					
Archivo	Edit Ske	etch Herramientas Help			
	€ 🕞	🜵 Arduino Uno 👻			
P	sketch_o	ct21a.ino			
	1	<pre>int LEDROJO = 9;</pre>			
	2	<pre>int LEDVERDE = 10;</pre>			
12	3	int LEDAZUL = 11;			
	4	int POR0 = 0;			
Mk	5	int POR1 = 1;			
Ши	6	int POR2 = 2;			
	7	int BRILLO0;			
	8	int BRILLO1;			
-	9	int BRILLO2;			
	10				
1 Q	11	void setup() {			
	12	pinMode(LEDROJO, OUTPUT);			
	13	pinMode(LEDVERDE, OUTPUT);			
	14	pinMode(LEDAZOL, OUTPOT);			
	15	3			
	10	void loop() {			
	10	BRTILOO = analogRead(DORO) / A:			
	10	analogWrite(LEDROIO BRILLOR):			
	20	BRILLO1 = analogRead(POR1) / 4:			
	21	analogWrite(LEDVERDE, BRILLO1):			
	22	BRILLO2 = analogRead(POR2) / 4:			
	23	analogWrite(LEDAZUL, BRILLO2);			
	24				
	25	}			

Ahora a buscar los colores que más te gusten.



Vamos realizar el correspondiente circuito.

Este dispositivo detecta el sonido, y cuando lo recibe haremos que se encienda la luz y nuevamente cuando lo vuelva a recibir qué se apague.



Vamos a escribir el código:
sketch_oct22a Arduino IDE 2.0.0		
Archivo	Edit Sket	ch Herramientas Help
	€ 🔊	🖞 Arduino Uno 👻
Ph	sketch_oc	t22a.ino
	1	int MIC = 2;
_	2	<pre>int LED = 3;</pre>
t_)	3	int VALOR;
	4	int ESTADO;
Iffk	5	
111/7	6	<pre>void setup() {</pre>
	7	<pre>pinMode(MIC, INPUT);</pre>
	8	<pre>pinMode(LED, OUTPUT);</pre>
2,5	9	}
	10	
Q	11	<pre>void loop() {</pre>
	12	<pre>VALOR = digitalRead(MIC);</pre>
	13	if (VALOR == HIGH){
	14	ESTADO = digitalRead(LED);
	15	<pre>digitalWrite(LED, !ESTADO);</pre>
	16	delay(500);
	17	}
	18	}

Si a este circuito le ponemos un relé podemos hacer que con el ruido de una palmada se encienda la luz de nuestra habitación.

La salida analogía nos brinda el valor de referencia del preset de sensibilidad. Vamos a realizar el programa para que lea el valor de la salida analógica.

🔤 sketch_oct22b Arduino IDE 2.0.0		
Archivo	Edit Sket	ch Herramientas Help
	€ 🕞	🖞 Arduino Uno 👻
Ph	sketch_oc	t22b.ino
	1	int MIC = 0;
	2	int VALOR;
12	3	<pre>void setup() {</pre>
	4	<pre>Serial.begin(9600);</pre>
nh	5	}
	6	
	7	<pre>void loop() {</pre>
	8	VALOR = analogRead(MIC);
~	9	<pre>Serial.println(VALOR);</pre>
	10	delay(500);
Q	11	}



Para poder enviar la señal a un Preamplificador.

Capítulo 14.- Sensor Crepuscular pro fotorresistencia (LDR)



Vamos a realizar el siguiente circuito:



Led con una resistencia de 220 Ω y el módulo KY-=18 consta de una Fotorresistencia más una resistencia de 10 K Ω .

Si queremos utilizar el equivalente.



La resistencia es de 10 K Ω .

Vamos a programar:

🧧 sketch_oct22a Arduino IDE 2.0.0		
Archivo	Edit Ske	tch Herramientas Help
Ø	∂ 🕑	🕂 Arduino Uno 👻
	sketch_o	ct22a.ino
	1	<pre>int SENSOR = 0;</pre>
	2	<pre>int LED = 3;</pre>
12	3	int VALOR;
	4	
Infly	5	<pre>void setup() {</pre>
	6	<pre>pinMode(LED, OUTPUT);</pre>
	7	// Entradas analóginas no requieren inicialización.
	8	}
	9	
	10	Void 100p() {
L C	11	VALOR = analogRead(SENSOR);
	12	if (VALOR < 600){
	13	digitalWrite(LED, LOW);
	14	delay(1000);
	15	}else{
	16	<pre>digitalWrite(LED, HIGH);</pre>
	17	}
	18	}

El valor de la condición lo tendrás que ajustar a tus necesidades y de la iluminación de donde estés trabajando.

sketch_oct22a Arduino IDE 2.0.0		
Archivo	Edit Sket	tch Herramientas Help
	€ 🕞	Arduino Uno 👻
P	sketch_oo	t22a.ino
	1	<pre>int SENSOR = 0;</pre>
	2	<pre>int LED = 3;</pre>
12	3	int VALOR;
	4	int PWM;
Mh	5	
	6	<pre>void setup() {</pre>
	7	<pre>pinMode(LED, OUTPUT);</pre>
	8	// Entradas analóginas no requieren inicialización.
~	9	}
	10	
Q	11	<pre>void loop() {</pre>
· ·	12	<pre>VALOR = analogRead(SENSOR);</pre>
	13	PWM = map(VALOR, 0, 1023, 0, 255);
	14	analogWrite(LED, 255-PWM);
	15	}

La iluminación del LED es inversamente proporcional a la luz que recibe el Sensor Crepuscular.

Si queremos que la iluminación del LED sea proporcional a la luz que recibe el Sensor Crepuscular, modificaríamos la línea 14.





Capítulo 15.- Sensor de Orientación (por interruptor de mercurio).

Este será el circuito de conexión:



Vamos a escribir el código:

1	<pre>int SENSOR = 2;</pre>
2	int LED = 3;
3	int ESTADO;
4	

5	<pre>void setup() {</pre>
6	<pre>pinMode(SENSOR, INPUT);</pre>
7	<pre>pinMode(LED, OUTPUT);</pre>
8	}
9	
10	<pre>void loop() {</pre>
11	ESTADO = digitalRead(SENSOR);
12	<pre>digitalWrite(LED, ESTADO);</pre>
13	delay(100);
14	}



A continuación vamos a realizar el correspondiente circuito.



Vamos a escribir el siguiente código:

```
int SENSOR DERECHO = 2;
 1
 2
     int LED_DERECHO = 3;
 3
 4
    int SENSOR IZQUIERDO = 4;
 5
     int LED_IZQUIERDO = 5;
 6
 7
    int ESTADO_DERECHO;
 8
     int ESTADO_IZQUIERDO;
9
10
    void setup() {
11
      pinMode(SENSOR_DERECHO, INPUT);
      pinMode(SENSOR_IZQUIERDO, INPUT);
12
      pinMode(LED_DERECHO, OUTPUT);
13
     pinMode(LED_IZQUIERDO, OUTPUT);
14
15
     }
16
17
    void loop() {
      ESTADO_DERECHO = digitalRead(SENSOR_DERECHO);
18
19
      digitalWrite(LED_DERECHO, ESTADO_DERECHO);
20
      delay(100);
21
22
      ESTADO_IZQUIERDO = digitalRead(SENSOR_IZQUIERDO);
      digitalWrite(LED_IZQUIERDO, ESTADO_IZQUIERDO);
23
24
     delay(100);
25
     }
```



Capítulo 16.- Sensor de Vibración KY-002 (y aplicaciones)

Vamos a realizar el siguiente circuito:



Vamos a programar:

1	<pre>int SENSOR = 2;</pre>
2	<pre>int LED = 3;</pre>
3	int ESTADO;
4	

```
void setup() {
 5
       pinMode(SENSOR, INPUT);
 6
 7
       pinMode(LED, OUTPUT);
 8
     }
 9
10
     void loop() {
       ESTADO = digitalRead(SENSOR);
11
12
       if (ESTADO == LOW){
13
        digitalWrite(LED, HIGH);
14
        delay(500);
       }
15
       digitalWrite(LED, LOW);
16
17
      ł
```



Capítulo 17.- Módulo Joystick analógico KY-023





Vamos a programar:

Hay que tener en cuenta que cuando el joystick está en el centro dejamos un margen que se muestra en el cuadrado de la siguiente imagen.



```
int X;
int Y;
int LED_IZQUIERDO = 3;
int LED_DERECHO = 5;
int LED_ABAJO = 6;
int LED_ARRIBA = 9;
int PULSADOR = 10;
int LED_SW = 11;
int SW;
void setup() {
  pinMode(LED IZQUIERDO, OUTPUT);
  pinMode(LED_DERECHO, OUTPUT);
  pinMode(LED_ABAJO, OUTPUT);
  pinMode(LED_ARRIBA, OUTPUT);
  pinMode(LED_SW, OUTPUT);
  pinMode(PULSADOR, INPUT);
  // Entradas analógicas no requiere inicialización
}
void loop() {
 X = analogRead(A0);
  Y = analogRead(A1);
  SW = digitalRead(PULSADOR);
  if (X >= 0 && X <= 480){
    analogWrite(LED_IZQUIERDO, map(X, 0, 480, 255, 0));
  }else {
    analogWrite(LED_IZQUIERDO, 0);
  }
  if (X >= 520 && X <= 1023){
    analogWrite(LED_DERECHO, map(X, 520, 1023, 0, 255));
  }else {
    analogWrite(LED_DERECHO, 0);
  }
```

```
if (Y >= 0 && Y <= 480){
    analogWrite(LED_ABAJO, map(Y, 0, 480, 255, 0));
}else {
    analogWrite(LED_ABAJO, 0);
}
if (Y >= 520 && Y <= 1023){
    analogWrite(LED_ARRIBA, map(Y, 520, 1023, 0, 255));
}else {
    analogWrite(LED_ARRIBA, 0);
}
digitalWrite(LED_SW, !SW);
}</pre>
```

Capítulo 18.- Sensor de Campo Magnético modelos KY (por efecto Hall) Ver video





Vamos a realizar el circuito:



Vamos a escribir el código:

💿 sket	🔤 sketch_oct23a Arduino IDE 2.0.0		
Archivo	Edit Sket	ch Herramientas Help	
	€ 🕞	Arduino Uno 👻	
Ph	sketch_oc	t23a.ino	
	1	int SENSOR;	
	2		
1_)	3	<pre>void setup() {</pre>	
	4	<pre>Serial.begin(9600);</pre>	
Inh	5		
	6	}	
	7		
	8	<pre>void loop() {</pre>	
204	9	<pre>SENSOR = analogRead(A0);</pre>	
	10	<pre>Serial.println(SENSOR);</pre>	
Q	11	delay(500);	
	12	}	

Vamos a modificar el programa:

1	int SENSOR;
2	<pre>void setup() {</pre>
3	<pre>Serial.begin(9600);</pre>
4	}

```
5
     void loop() {
 6
       SENSOR = analogRead(A0);
 7
 8
       if(SENSOR < 450 || SENSOR > 600){
         Serial.print("Campo detrectado: ");
9
10
       }
11
       Serial.println(SENSOR);
       delay(500);
12
13
```

Este será el resultado:

Salida Monitor Serie ×	Salida Monitor Serie \times
Message (Enter to send message	Message (Enter to send message
Campo detrectado: 370	- Campo detrectado: 702
Campo detrectado: 377	Campo detrectado: 703
Campo detrectado: 381	Campo detrectado: 705
Campo detrectado: 383	Campo detrectado: 705
Campo detrectado: 385	Campo detrectado: 707
Campo detrectado: 385	Campo detrectado: 707
Campo detrectado: 385	Campo detrectado: 708

Tanto si el imán lo pasamos por el polo positivo o negativo se sale de los márgenes que hemos especificado en programa.

Capítulo 19.- Interrupciones externas (pruebas con KY-010)

Ver video

Interrup	ociones externas	s en Arduino	
	MODELO	PINES INTERRUPCIONES	118
	Uno, Nano, Mini y otras basadas en 328	2, 3	
	Mega, Mega2560 y MegaADK	2, 3, 18, 19, 20, 21	0 2
And the second s	Micro, Leonardo y otras basadas en 32u4	0, 1, 2, 3, 7	
	Zero	todos los pines digitales excepto el 4	
1 2 2 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1	Due	todos los pines digitales	
attachInterrupt(di	gitalPinToInterrupt	(pin), ISR, modo);	
	pin: 2 ò 3 en Ardu	ino UNO	
	ISR: función a lla se detecte la	mar cuando interrupción	
	modo: •LOW // •RISING // •CHANGE // •FALLING // •HIGH* //	nivel bajo flanco de subida cambio de nivel flanco de bajada nivel alto HIGH	
			ALLING
* unicamente disponible para mod	etos Due, Zero y MKR1000		
		CHANGE	



Vamos a diseñar el circuito:



Vamos a programar:

🥯 sketch_oct23a Arduino IDE 2.0.0			
Archivo	Edit Ske	tch Herramientas Help	
\bigcirc	€ 🗧	🖞 Arduino Uno 👻	
	sketch_o	ct23a.ino	
	1	volatile int CONTADOR = 0; // Variable global	
	2	<pre>int ANTERIOR = 0;</pre>	
1_)	3		
	4	<pre>void setup() {</pre>	
Iffk	5	Serial.begin(9600);	
	6	<pre>attachInterrupt(digitalPinToInterrupt(2), sensor, RISING);</pre>	
	7	}	
a −	8		
	9	void loop() {	
	10	1+ (ANTERIOR != CONTADOR){	
X	11	Serial.printin(CONTADOR);	
	12	ANTERIOR = CONTADOR;	
	13		
	14	3	
	15	void sensor(){	
	17	CONTADOR++:	
	18	}	



Ahora vamos a modificar el código para que cuando vayamos realizando interrupciones el parpadeo del LED 13 vaya más rápido.

🔤 sketch_oct23a Arduino IDE 2.0.0			
Archivo	Edit Ske	tch Herramientas Help	
	⇒ 🔊	v ⁴ Arduino Uno	
Ph	sketch_o	ct23a.ino	
	1	volatile int CONTADOR = 1000; // Variable global	
	2	<pre>int ANTERIOR = 0;</pre>	
1-)	3		
	4	<pre>void setup() {</pre>	
Mk	5	<pre>pinMode(13, OUTPUT);</pre>	
	6	<pre>attachInterrupt(digitalPinToInterrupt(2), sensor, RISING);</pre>	
	7	}	
	8		
~	9	void loop() {	
	10	digitalWrite(13, HIGH);	
Q	11	delay(CONTADOR);	
	12	digitalWrite(13, LOW);	
	13	delay(CONTADOR);	
	14	1+ (CONTADOR < 100){	
	15	CONTADOR = 1000;	
	10	}	
	1/		
	10	3	
	20	void senson(){	
	20	CONTADOR = CONTADOR - 100	
	22	}	

Capítulo 20.- Codificador rotatorio KY-040 (rotay encoder)























Vamos a realizar el circuito.



```
unsigned long: variable sin signo
de 32 bits.
(0 a 4.294.967.295)
millis(): tiempo transcurrido desde
▷ el inicio del programa
en milisegundos.
static: modificador para indicar
que una variable debe
mantener su valor entre
llamados de la función.
```

```
min(x, y): calcula el mínimo de
    dos números.
    (permite establecer el
    límite superior)
max(x, y): calcula el máximo de
    dos números.
    (permite establecer el
    límite inferior)
```

Vamos a realizar la programación:

```
int A = 2; //DT
int B = 4; //CLK
int ANTERIOR = 50;
volatile int POSICION = 50;
void setup() {
 pinMode(A, INPUT);
 pinMode(B, INPUT);
 Serial.begin(9600);
 attachInterrupt(digitalPinToInterrupt(A), encoder, LOW);
 Serial.println("Listo");
}
void loop() {
  if (POSICION != ANTERIOR){
    Serial.println(POSICION);
    ANTERIOR = POSICION;
  }
}
void encoder(){
  static unsigned long ultimaInterrupcion = 0;
  unsigned long tiempoInterrupcion = millis();
  if(tiempoInterrupcion - ultimaInterrupcion > 5){
    if (digitalRead(B) == HIGH)
    {
      POSICION++;
    }
    else {
      POSICION--;
    }
     POSICION = min(100, max(0, POSICION));
     ultimaInterrupcion = tiempoInterrupcion;
    }
}
```

Capítulo 21.- LM35 Sensor analógico de temperatura + LCD 1602A



Vamos a realizar el correspondiente circuito:



Esta será el código:

```
int SENSOR;
 1
     float TEMPERATURA;
 2
 3
     void setup() {
 4
     Serial.begin(9600);
 5
 6
     }
 7
 8
    void loop() {
     SENSOR = analogRead(A0);
9
     TEMPERATURA = ((SENSOR * 5000.0)/1023)/10;
10
     Serial.println(TEMPERATURA, 1); //El 1 le decimos que solo queremos 1 decimal.
11
     delay(1000);
12
13
     }
```

Como resultado:

Salida M	∕lonitor Serie ×
Message (Enter to send message to 'Arduino Uno' on 'COM3')
21.1	
21.2	
21.2	
21.4	
21.5	

Para obtener un valor más preciso vamos a realizar 5 lecturas y el promedio se mostrará en el monitor serial.

```
1
     int SENSOR;
 2
     float TEMPERATURA;
     float SUMA;
 3
 4
     void setup() {
 5
     Serial.begin(9600);
 6
     }
 7
 8
     void loop() {
 9
      SENSOR = analogRead(A0);
      SUMA = 0;
10
11
      for (int i=0; i<5; i++){</pre>
        TEMPERATURA = ((SENSOR * 5000.0)/1023)/10;
12
        SUMA = SUMA + TEMPERATURA;
13
14
       delay(500);
15
      }
      Serial.println(SUMA/5.0, 1); //El 1 le decimos que solo queremos 1 decimal.
16
17
     }
```

Este será el resultado:

Salida	Monitor Serie \times
Messag	e (Enter to send message
24.2	
24.2	
24.2	

Vamos a realizar el siguiente circuito:



Este será el código:

```
#include <LiquidCrystal.h>
 1
 2
     LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
 З
     int SENSOR;
 4
     float TEMPERATURA;
 5
     float SUMA;
 6
     void setup() {
 7
       lcd.begin(16,2);
8
      Serial.begin(9600);
 9
     3
10
11
     void loop() {
      SENSOR = analogRead(A0);
12
13
      SUMA = 0;
      for (int i=0; i<5; i++){</pre>
14
15
        TEMPERATURA = ((SENSOR * 5000.0)/1023)/10;
16
        SUMA = SUMA + TEMPERATURA;
17
        delay(500);
18
      }
19
      lcd.setCursor(0,0);
20
      lcd.print("Temp: ");
21
      lcd.print(SUMA/5.0, 1);
22
      lcd.print(" C");
23
     }
```

Si dispones de dos sensores de temperatura DS18B20 te propongo el siguiente ejercicio. Tendrás que importar las siguientes librerías:

> OneWire de Jim Studt, Tom Pollard, Robin James, Glenn Trewitt, Jason Dangel, Guillermo Lovato, Paul Stoffregen, Scott Roberts, Bertrik Sikken, Mark Tillotson, Ken Butcher, Roger Clark, Love Nystrom Access 1-wire temperature sensors, memory and other chips.

DallasTemperature de Miles Burton <miles@mnetcs.com>, Tim Newsome <nuisance@casualhacker.ne Guil Barros <gfbarros@bappos.com>, Rob Tillaart <rob.tillaart@gmail.com> Supports DS18B20, DS18S20, DS1822, DS1820 Arduino Library for Dallas Temperature ICs

Si desconectar el monitor lcd de propongo que hagas el siguiente circuito.

Podrás observar que ahora está conectado el 9 PWM.

Queremos que nos muestre la lectura alternando los dos sensores con un intervalo de 5 segundos.



Este será el código:

1	<pre>#include <dallastemperature.h></dallastemperature.h></pre>
2	<pre>#include <onewire.h></onewire.h></pre>
3	<pre>#include <liquidcrystal.h></liquidcrystal.h></pre>
4	
5	LiquidCrystal <pre>lcd(7, 6, 5, 4, 3, 2);</pre>
6	
7	<pre>const int pinDatosDQ = 9;</pre>
8	float TEMPERATURA;
9	float SUMA;
10	OneWire oneWireObjeto(pinDatosDQ);
11	DallasTemperature <pre>sensorDS18B20(&oneWireObjeto);</pre>
12	

```
13
     void setup() {
14
      lcd.begin(16,2);
15
      Serial.begin(9600);
      sensorDS18B20.begin();
16
17
     }
18
19
     void loop() {
20
      sensorDS18B20.requestTemperatures();
      lcd.setCursor(0,0);
21
      lcd.print("Temp 1: ");
22
23
      lcd.print(sensorDS18B20.getTempCByIndex(0));
24
      lcd.print(" C");
25
      delay(5000);
26
      lcd.clear();
27
      lcd.setCursor(0,0);
      lcd.print("Temp 2: ");
28
29
      lcd.print(sensorDS18B20.getTempCByIndex(1));
30
      lcd.print(" C");
31
      delay(5000);
32
      lcd.clear();
33
```

Proyecto final:



Capítulo 22.- Bluetooth HC-05 Introducción y comandos AT



AT+PSWD Contraseña (PIN)	
AT+UART Parámetros de comunicación	
AT+ROLE Rol del dispositivo: 0: esclavo (slave) 1: maestro (master)	and the second sec
AT+ORGL Restaura a valores de fábrica	
AT+RESET Vuelve a modo usuario	



Vamos a la programación:

🧧 sketch_oct24a Arduino IDE 2.0.0			
Archivo	Edit Ske	tch Herramientas Help	
	€	Arduino Uno 👻	
Ph	sketch_o	ct24a.ino	
	1	<pre>#include <softwareserial.h></softwareserial.h></pre>	
	2		
1_)	3	SoftwareSerial miBT(10, 11);	
	4		
ITIN	5	<pre>void setup() {</pre>	
	6	Serial.begin(9600);	
	7	<pre>Serial.println("Listo");</pre>	
	8	miBT.begin(38400);	
200	9	}	
	10		
Q	11	<pre>void loop() {</pre>	
	12	<pre>if (miBT.available())</pre>	// Lee BT y envía a Arduino
	13	<pre>Serial.write(miBT.read());</pre>	
	14		
	15	if (Serial.available())	
	16	<pre>miBT.write(Serial.read());</pre>	// Lee Arduino y envía a BT
	17	}	

Observamos que al cargar el programa el Bluetooth parpadea de forma rápida.

Para entrar en modo de configuración vamos a realizar los siguientes pasos.

Desconectamos el cable que alimenta con 5 voltios al Bluetooth.

Observarás que el módulo se apaga.

Volvemos a conectar los 5 voltios mientras mantenemos pulsado el botón de la placa del Bluetooth, durante unos 5 segundos, ahora observamos que el parpadeo es más lento.

Ahora vamos a abrir el monitor serial de Arduino.



En la parte derecha de la consola tenemos que tener seleccionado NL & CR.

Enviamos el mensaje AT seguido de Ctrol + Intro, si nos contesta con Ok es que está listo.

Vamos a preguntar por el nombre del Bluetooth.

AT+NAME? y nos contestará :



Ahora vamos a preguntar por la contraseña: AT+PSWD?



Ahora vamos a consultar su rool: AT+ROLE?



O significa que está como esclavo.

Ahora vamos a preguntar AT+UART?



Vamos a restaurar los valores de fábrica: AT+ORGL

OK	Ì
	J

Vamos a cambiar el nombre del Bluetooth: AT+NAME=PERE

ſ	OK	l
Π.,		

Ahora vamos a cambiarle la contraseña: AT+PSWD=1234.

Vamos a configurarlo con el rool esclavo: AT+ROLE=0.

Vamos a establecer la comunicación a: AT+UART=38400,0,0

Vamos a salir del modo configuración: AT+RESET, ahora ya vuelve a parpadear rápidamente.

A continuación vamos a vincular el Bluetooth a nuestro dispositivo móvil.

Nos pedirá el pin que le asignamos en la configuración.
Capítulo 23.- Bluetooth HC-05 (App en teléfono Android + LEDs + PWM)



Vamos a realizar el siguiente circuito:



Ahora en tu dispositivo Android tendrás que instalarte la siguiente aplicación:



Arduino bluetooth controller Vamos a rescribir el siguiente código:

```
🔤 sketch_oct24a | Arduino IDE 2.0.0
Archivo Edit Sketch Herramientas Help
       ÷
                  Arduino Uno
                                             •
       sketch_oct24a.ino
               #include <SoftwareSerial.h>
          1
          2
          3
              SoftwareSerial miBT(10, 11);
          4
          5
              char DATO = 0;
          6
              int LEDROJO = 2;
          7
              int LEDVERDE = 3;
          8
 ₽
          9
              void setup() {
                 miBT.begin(9600);
         10
         11
                 pinMode(LEDROJO, OUTPUT);
         12
                 pinMode(LEDVERDE, OUTPUT);
         13
               3
         14
         15
               void loop() {
         16
                 if (miBT.available()) {
                   DATO = miBT.read();
         17
                   if(DATO == '1')
         18
         19
                   digitalWrite(LEDROJO, HIGH);
         20
                   if(DATO == '2')
         21
                   digitalWrite(LEDROJO, LOW);
         22
         23
                   if(DATO == '3')
         24
                   digitalWrite(LEDVERDE, HIGH);
         25
         26
                   if(DATO == '4')
         27
                   digitalWrite(LEDVERDE, LOW);
         28
         29
         30
                 }
         31
         32
               }
```

Lo cargamos en el Arduino.

Ejecutamos la aplicación que instalamos con anterioridad.

Ahora vamos a modificar el programa, para que el led verde que está en el pin 3[~] cada vez que enviemos en número 2 se incremente su brillo en 10 y cada vez que enviemos un 3 decremente su brillo en 3.

```
BuzzerPasivo | Arduino IDE 2.0.0
Archivo Edit Sketch Herramientas Help
       →
                  ÷
                    Arduino Uno
                                              ÷
       BuzzerPasivo.ino
               #include <SoftwareSerial.h>
          1
          2
          3
              SoftwareSerial miBT(10, 11);
          4
          5
             char DATO = 0;
          6
               int LEDROJO = 2;
          7
              int LEDVERDE = 3;
          8
               int BRILLO = 0;
 ÷
          9
         10 \vee void setup(){
                 miBT.begin(9600);
         11
                 pinMode(LEDROJO, OUTPUT);
         12
         13
                 pinMode(LEDVERDE, OUTPUT);
         14
               3
         15
         16 \vee void loop(){
                 if(miBT.available()){
         17 🗸
                   DATO = miBT.read();
         18
                   if (DATO == '2'){
         19 \sim
         20
                     BRILLO = BRILLO + 10;
         21 \lor
                      if (BRILLO > 255){
         22
                       BRILLO = 255;
         23
                      }
                     analogWrite(LEDVERDE, BRILLO);
         24
         25
                   3
                   if (DATO == '3'){
         26 \sim
                      BRILLO = BRILLO - 10;
         27
         28 \sim
                      if (BRILLO < 0){
                       BRILLO = 0;
         29
         _{30} \sim
         31
                      analogWrite(LEDVERDE, BRILLO);
         32
         33
                 }
          34
               }
```

Capítulo 24.- Receptor Infrarrojo KY-022 + LED RGB (Lector Código)





Vamos a diseñar el circuito:

RESET			
IC			
	Rx = Ardu no"		
		N	
		 W.	
Módulo KY-022	·····		
Módulo KY-022		9 8 8	• • • • • • • •
Módulo KY-022			· · · · · ·
Módulo KY-022		• * * * * *	S
Módulo KY-022			•
Módulo KY-022			
Módulo KY-022			
Módulo KY-022			

Ahora vamos a programar, el objetivo es mostrar en el monitor serial el valor del botón que seleccionamos en el mando a distancia.

IRremoteESP8266 de
David Conran, Sebastien
Warin, Mark Szabo, Ken
Shirriff
This library enables you to
send and receive infra-red
signals on an ESP8266 or
an ESP32.
Send and receive infrared
signals with multiple
protocols (ESP8266/ESP32)
Más información

Primero vamos a descargar la correspondiente librería.

ControlRemoto Arduino IDE 2.0.0								
Archivo	Edit Ske	tch Herramientas Help						
Ø	⇒ 🕑	🖞 Arduino Uno 👻						
Ph	ControlR	emoto.ino						
	1	<pre>#include <irremote.hpp></irremote.hpp></pre>						
_	2							
t_)	3	<pre>int SENSOR = 11;</pre>						
	4	<pre>IRrecv irrecv(SENSOR);</pre>						
Ու	5	decode_results codigo;						
ШИ	6							
	7	<pre>void setup() {</pre>						
	8	Serial.begin(9600);						
2,04	<pre>9 irrecv.enableIRIn();</pre>							
	10							
Q	11	}						
	12							
	13	<pre>void loop() {</pre>						
	14	if(irrecv.decode(&codigo)){						
	15	<pre>Serial.println(codigo.value, HEX);</pre>						
	16	<pre>irrecv.resume();</pre>						
	17	}						
	18	delay(100);						
	19	}						

Decimal (base 10): 0,1,2,3,4,5,6,7,8,9 Hexadecimal (base 16): 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F Ejemplo:

```
12 en Decimal: 12 (2 dígitos)
12 en Hexadecimal: C (1 dígito)
```

0	FF6897
1	FF30CF
2	FF18E7
3	FF7A85
4	FF10EF
5	FF38C7
6	FF5AA5
7	FF42BD
8	FF4AB5
9	FF52AD



Valores que retorna en hexadecimal cuando presionamos desde el mando a distancia los valores del 0 al 1, en tu caso los valores pueden ser otros. Si te aparece el valor FFFFFFF significa que estás manteniendo pulsado el mismo botón durante un tiempo prolongado.

Vamos a realizar el correspondiente circuito:



Vamos a programar:

🔤 Cor	🔤 ControlRemoto Arduino IDE 2.0.0							
Archivo	Archivo Edit Sketch Herramientas Help							
\bigcirc	€ 🕞	🖞 Arduino Uno 👻						
	ControlRe	emoto.ino						
	1	<pre>#include <irremote.hpp></irremote.hpp></pre>						
	2	#define Boton_1 0xFF30CF						
12	3	#define Boton_2 0xFF18E7						
	4	#define Boton_3 0xFF7A85						
Infly	5	int SENSOR = 11;						
ШИ	6	IRrecv irrecv(SENSOR);						
	7	decode_results codigo;						
	8	<pre>int LEDROJO = 2;</pre>						
~	9	<pre>int LEDVERDE = 3;</pre>						
	10	<pre>int LEDAZUL = 4;</pre>						
Q	11 \vee	<pre>void setup() {</pre>						
	12	<pre>Serial.begin(9600);</pre>						
	13	<pre>irrecv.enableIRIn();</pre>						
	• 14	<pre>pinMode(LEDROJO, OUTPUT);</pre>						
	15	<pre>pinMode(LEDVERDE, OUTPUT);</pre>						
	16	<pre>pinMode(LEDAZUL, OUTPUT);</pre>						

```
17
     }
18
19 \vee void loop() {
20 🗸
       if(irrecv.decode(&codigo)){
         Serial.println(codigo.value, HEX);
21
22
         if(codigo.value == Boton_1)
         digitalWrite(LEDROJO, !digitalRead(LEDROJO));
23
         if(codigo.value == Boton_2)
24
         digitalWrite(LEDVERDE, !digitalRead(LEDVERDE));
25
26
         if(codigo.value == Boton_3)
         digitalWrite(LEDAZUL, !digitalRead(LEDAZUL));
27
        irrecv.resume();
28
29
       }
30
       delay(500);
31
     3
```

Capítulo 25.- Sensor de línea KY-033 (seguidor de línea)



Vamos a diseñar el circuito:



Vamos a programar:

🥯 sketch_oct25a Arduino IDE 2.0.0					
Archivo	Edit Sket	ch Herramientas Help			
Ø	€ 🔊	🖞 Arduino Uno 👻			
	sketch_oc	t25a.ino			
	1	<pre>int SENSOR = 2;</pre>			
	• 2	int VALOR;			
1	3	int ANTERIOR = 1;			
	4				
Inh	5	void setup() {			
	6	<pre>Serial.begin(9600);</pre>			
	7	pinMode(SENSOR, INPUT);			
÷	8	}			
	10	void loop() (
	10	VALOR = digitalRead(SENSOR):			
\sim	12	if (VALOR != ANTERIOR){			
	13	if (VALOR == HIGH)			
	14	<pre>Serial.println("Linea");</pre>			
	15	else			
	16	<pre>Serial.println("Fuera");</pre>			
	17	ANTERIOR = VALOR;			
	18	}			
	19	delay(500);			
	20	}			

En una cartulina de color blanco, tienes que agregar unas tiras de color negro, como si fuera la carretera.



Este será el resultado:

Salida	Monitor Serie ×
Mensaje	(Ctrl + Enter para enviar el mensaje a 'Arduino Uno' en 'COM3')
Fuera	
Linea	
Fuera	
Linea	
Fuera	
Linea	
Fuera	

Cuando está en la zona blanca dice Fuera y cuando está en la zona negra dice línea.

Un ejemplo para el uso es el seguimiento de un línea.



Como solo tiene dos ruedas una de las ruedas se para y al moverse solo una es cuando nuestro vehículo gira.



Como el sensor no sabe si la línea negra está a su derecha o izquierda existe una solución:



Lo ideal es trabajar con 2 módulos:



Este módulo también se puede ubicar en un sitio fijo y cuando otro objeto se mueva peda detectar su ubicación exacta.



Capítulo 26.- Sensor de gas natural MQ-5 (y combustibles)



Este será el circuito:



Vamos a programar:

🥯 sketch_oct24a Arduino IDE 2.0.0							
Archivo Edit Sketch Herramientas Help							
	€ 🕞	Arduino Uno 👻					
Ph	sketch_oc	t24a.ino					
	1	int VALOR;					
	2						
1_)	3	<pre>void setup() {</pre>					
	<pre>4 Serial.begin(9600);</pre>						
Mh	5	}					
ШИ	6						
	7	<pre>void loop() {</pre>					
	8	VALOR = analogRead(A0);					
200	9	<pre>Serial.println(VALOR);</pre>					
	10	<pre>delay(500);</pre>					
Q	11	}					

Al ejecutar le hemos aproximado en encendedor de gas solo saliendo el gas cerca del sensor.

Mensaje (Ctrl + Enter para enviar el mensaje a 'Arduino Uno' en 'COM3')	
258	
249	
242	
235	
229	
224	
218	
214	
A*A	

```
Proceso de "curado" del sensor
24 hs: recomendación del fabricante
para resultados precisos
6 hs: mi recomendación para
buenos resultados
15 min: resultados muy aproximados
(solo para impacientes)
```



Del menú Herramientas podemos seleccionar Plotter Serie para mostrar un gráfico con los valores:



🥯 sketch_oct24a Arduino IDE 2.0.0					
Archivo	Edit Sket	ch Herramientas Help			
	€ 🕞	∲ Arduino Uno 👻			
P -1	sketch_oo	t24a.ino			
	1	int VALOR:			
	2	int BUZZER = 2;			
f_)	3	int LED = 13;			
	4	<pre>void setup() {</pre>			
D-D-	5	Serial.begin(9600);			
ША	6	<pre>pinMode(LED, OUTPUT);</pre>			
	7	<pre>pinMode(BUZZER, OUTPUT);</pre>			
	8	}			
Đ.	9				
	10	<pre>void loop() {</pre>			
Q	11	VALOR = analogRead(A0);			
	12	if (VALOR > 400){			
	13	<pre>digitalWrite(BUZZER, HIGH);</pre>			
	14	<pre>digitalWrite(LED, HIGH);</pre>			
	15	}else {			
	16	<pre>digitalWrite(BUZZER, LOW);</pre>			
	17	<pre>digitalWrite(LED, LOW);</pre>			
	18	}			
	19	<pre>Serial.println(VALOR);</pre>			
	20	delay(500);			
	21	3			





Un ejemplo de esta utilidad consiste que cuando detecte gas un relé ponga en marcha un ventilador.

Pare evitar posibles chispas que pueda realizar el relé este debe de permanecer lejos de donde se produce la fuga de gas.

Capítulo 27.- Neopixel LED RGB inteligente WS2812 (y titas)





Este es el circuito:



Este será el código:

```
1 #include <Adafruit_NeoPixel.h>
2 Adafruit_NeoPixel tira = Adafruit_NeoPixel(8, 2, NEO_GRB + NEO_KHZ800);
3 
4 void setup() {
5 tira.begin();
6 tira.show();
7 }
```

Como podrás observar se enciende el primer led con el color azul.

Vamos a modificar el código:

```
#include <Adafruit_NeoPixel.h>
 1
 2
     Adafruit_NeoPixel tira = Adafruit_NeoPixel(8, 2, NEO_GRB + NEO_KHZ800);
 3
 4
     void setup() {
 5
       tira.begin();
 6
       tira.show();
7
     3
8
9
     void loop() {
10
       tira.setBrightness(20);
                                       //Brillo global para toda la tira.
11
       for(int i = 0; i<8; i++){</pre>
12
         tira.setPixelColor(i,0,0,255); //El prier valor es el número de led,
13
                                          //Seguido de los colores R,G,B en
14
                                          //este caso saldrá el color azul.
15
         tira.show();
16
         delay(500);
17
       3
```

Cuando lo ejecutes encenderá un led detrás de otro hasta encender todos los leds.

Al agregar la función clear() cada vez que se encienden todos los led, estos se apagarán para volver a repetir el ciclo.

Ahora realiza las siguientes modificaciones:

```
9 \checkmark void loop() {
       tira.setBrightness(20);
                                        //Brillo global para toda la tira.
10
        for(int i = 0; i<8; i++){</pre>
11
12
          tira.setPixelColor(i,0,0,255); //El prier valor es el número de led,
                                           //Seguido de los colores R,G,B en
13
14
                                           //este caso saldrá el color azul.
15
          tira.show();
          delay(500);
16
17 \sim
        for(int i = 7; i>=0; i--){
18 \sim
19
          tira.setPixelColor(i,255,0,0); //El prier valor es el número de led,
20
                                           //Seguido de los colores R,G,B en
                                           //este caso saldrá el color rojo.
21
22
          tira.show();
23
          delay(500);
24
        }
```

Se ponen de color azul empezando desde el primero hasta el último y luego de color rojo desde el último hasta el primero.

Vamos a agregar las siguientes líneas:

```
void loop() {
 9
10
       tira.setBrightness(20);
                                        //Brillo global para toda la tira.
       for(int i = 0; i<8; i++){</pre>
11
         tira.setPixelColor(i,0,0,255); //El prier valor es el número de led,
12
13
                                          //Seguido de los colores R,G,B en
                                          //este caso saldrá el color azul.
14
15
         tira.show();
16
         delay(500);
         tira.setPixelColor(i,0,0,0);
17
18
         tira.show();
19
         for(int i = 7; i>=0; i--){
20
21
         tira.setPixelColor(i,255,0,0); //El prier valor es el número de led,
22
                                          //Seguido de los colores R,G,B en
23
                                          //este caso saldrá el color rojo.
24
         tira.show();
25
         delay(500);
         tira.setPixelColor(i,0,0,0);
26
27
         tira.show();
28
```

Ahora las luces se encienden y se apagan antes de encenderse el siguiente led.



Matriz de 8x8



0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Matriz de 8x8



0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Capítulo 29.- Librería Stepper con 28BYJ-48(PAP unipolar) y ULN2003



Utilizaremos el mismos circuito:



Paso	А	В	С	D
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

Vamos a programar:

🔤 sket	tch_oct25b	Arduino IDE 2.0.0			
Archivo	Edit Ske	etch Herramientas Help			
Ø	€ 🕞	Arduino Uno 👻			
Ph	sketch_oct25b.ino				
	1	<pre>#include <stepper.h></stepper.h></pre>			
	2				
12	3	<pre>Stepper motor1(2048, 8, 10, 9, 11); // Pasos completos</pre>			
	4				
Mk	5	<pre>void setup() {</pre>			
	6	<pre>motor1.setSpeed(2); // En RPM (valores de 1, 2, o 3 para 28BYJ-48)</pre>			
	7				
÷	8	}			
	9	unid long() (
	10	void 100p() {			
X	11	dolow(2000):			
	12	ueray(2000),			
	13	}			

Comentario de la línea 6.



Hemos elegido en el programa 2 RPM.

Y se queremos realizar un giro en sentido contrario.

```
10 \void loop() {
11 motor1.step(512); // Cantidad de pasos dirección horario
12 delay(2000);
13 motor1.step(-512); // Cantidad de pasos dirección antihorario
14 delay(2000);
15 }
```

Hemos modificado el código del programa anterior, ahora el motor gira en sentido horario un cuarto de vuelta, se detiene 2 segundos, a continuación gira en sentido antihorario un cuarto de vuelta y se detiene de nuevo 2 segundos.

En el siguiente proyecto queremos asignarle el número de vueltas desde la consola.



Ejecutamos el programa y activamos el monitor en serie.

Salida Monitor Serie	×
Mensaje (Ctrl + Enter pa	ra enviar el mensaje a 'Arduino Uno' en 'COM3')
512	

Hemos introducido el valor 512 que nos lo muestra y el motor empieza a actuar.

Si ponemos -512 el motor gira en un cuarto de vuelta en sentido contrario.

Observarás que por defecto hay dos led encendidos, esta es la configuración de esta librería.



Para des energizar las bobinas vamos a agregar el siguiente código:

```
void loop() {
    if (Serial.available()){
        int pasos = Serial.parseInt();
        Serial.println(pasos);
        motor1.step(pasos);
        digitalWrite(8, LOW);
        digitalWrite(10, LOW);
        digitalWrite(10, LOW);
        digitalWrite(11, LOW);
    }
}
```

Ahora cuando el motor esté parado las luces no estarán encendidas como se muestra en la imagen anterior.



Capítulo 30.- PRO MINI y conversores USB-TTL PL2303 CH340



Ver video







Arduino Pro Mini







Ya podemos conectar el USB a nuestro ordenador.



Tenemos que seleccionar el tipo de placa y en el puerto que tiene que estar conectado.

Introduciremos el siguiente código para verificar si funciona.

```
void setup() {
1
2
       pinMode(13, OUTPUT);
3
4
     }
5
     void loop() {
6
7
       digitalWrite(13, HIGH);
8
       delay(1000);
9
       digitalWrite(13, LOW);
       delay(1000);
10
11
     }
```

Cuando en la barra del programa del Arduino IDE después de compilar cuando diga subiendo, hemos de presionar el botón RESET de la placa Arduino mini.



Capítulo 31.- LCD 12C adaptador e instalación de librerías

específicas






Vamos al circuito:



Vamos a programar:

Tenemos que descargar la librería del siguiente enlace.

https://web.archive.org/web/20200720122215/https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads/

🔽 fma	alpartida / Ne	ew LiquidCrystal / × +						~	-		×
$\leftarrow \rightarrow$	С 🚺	https://web.archive.org/web	b/20200720122215/https://bitbucket.org/fmalpar	ida/new-liquidcryst	al/downloads/		፼ \$	*	=J [1 🥠) :
Aplic	aciones										
INTERNE	T ARCHIVE	https://bitbucket.org/fmalpartida/	new-liquidcrystal/downloads/				Go SEP JUL	SEP	٢	?	3
NAÂRØ(CK IIIRCUNG	45 captures 27 Jan 2013 - 30 May 2022					2019 2020	2021	▼ About	f his captu	2 are
Q	· <>	Source Commits	Francisco Malpartida / Untitled project /	New LiquidCrysta	I You temp	caught us doing s porarily restricting access.	ome quick main your repositor	ntenan 7 to re	ce, ad-	×	
	jy	Branches	Downloads Tags Branches								
	រែ	Pull requests	Name	Size	Uploaded by	Downloads	Date				
		Pipelines	Download repository	4.9 MB							. 1
		Deployments	NewLiquidCrystal_1.5.1.zip	6.2 MB	Francisco	121037	2019-04-28				- 1
	⊻	Issues			Malpartida						- 1
	\$	Jira issues NEW	NewLiquidCrystal_1.5.0.zip	6.1 MB	Francisco Malpartida	14015	2019-03-23				
	F	Wiki	Newliquidcrystal_1.3.5.zip	2.4 MB	Francisco	150472	2017-11-26				
	₽	Downloads	NewliquidCrystal_1.3.4.zip	6.0 MB	Malpartida Francisco Malpartida	391460	2015-10-24				

Lo descargamos y lo descomprimimos.

Tienes que renombrar la antigua librería y a la nueva copiarla en la carpeta con el mismo nombre que tenía la anterior.



La librería Wire.h ya viene por defecto y la podemos cargar.

Las librerías LCD.h y LiquidCrystal_I2C.h la hemos obtenido de la nueva librería que hemos descargado y sustituida por la anterior.

En la línea 5 creamos un objeto llamado lcd que será de tipo LiquidCrystal_I2C a los que le pasaremos una serie de parámetros como la dirección que es 27 en hexadecimal, además de

indicarle el número de puertos del circuito integrado del adaptador corresponde a los pines E, RW, RS, D4, D5, D6, D7.



En la línea 8 sirve para indicar el número de puertos paralelos que se usará para habilitar la iluminación trasera de la pantalla y que sea positiva.

En la línea 9 enviando un nivel alto encendemos la iluminación posterior.

En la línea 10 le decimos que nuestro modelo en un 16 x 2.

En la línea 11 borra de la pantalla cualquier contenido que pudiera tener.

La parte del void loop() es igual que cuando trabajamos la pantalla sin el adaptador.





Consiste en puentear las soldaduras.



Este ahora sería 0, 1, 1 con la dirección 0x26, hay un total de 8 combinaciones.

Capítulo 32.- BMP280 Presión atmosférica, Temperatura, Altitud









Vamos a soldar los pines.

Vamos a realizar el circuito.



Vamos a programar:

Tenemos que incluir una librería. Buscaremos por BMP280.

Adafruit BMP280 Library de Adafruit Arduino library for BMP280 pressure and altitude sensors. Arduino library for BMP280 sensors. Más información

Antes de continuar vamos a abrir la carpeta donde estás las librerías.

Image: Second					-		× ^ ?
Anclar al Acceso rápido	Mover Copiar a * a *	Nueva elemento ▼ Nueva carpeta	Propiedades	Sele No s	ccionar tod seleccionar i rtir selecció	o nada n	
Portapapeles	Organizar	Nuevo	Abrir	Abrir Sele			
← → · ↑ 🔒 > OneDrive - Personal > Docur	nentos > Arduino > libraries		~	Ō	,⊂ Bus	car en libra	aries
Documentos	Nombre	Fecha de modi	ficación Tipo	Tam	naño		^
Adobe	Adafruit_10DOF	20/10/2022 17:	00 Carpeta de archi	vos			
Arduino	Adafruit_BMP085_Unified	20/10/2022 17:	00 Carpeta de archi	vos			
Audacity	Adafruit_BMP280_Library	08/11/2022 15:	00 Carpeta de archi	vos			
Corel VideoStudio MCE	Adafruit_BusIO	06/11/2022 10:0	09 Carpeta de archi	Carpeta de archivos			
Desumentes digitalizades	Adafruit_EPD	23/10/2022 18:4	48 Carpeta de archi	vos			
	Adafruit_GFX_Library	23/10/2022 18:4	48 Carpeta de archi	vos			
Downloads	Adafruit_HX8357_Library 25/10/2022 10:0		01 Carpeta de archi	vos		_	~
48 elementos 1 elemento seleccionado							III 📼

Adafruit_BMP280_Library

Abrimos dicha carpeta y abrimos el siguiente archivo, con el bloc de notas.

C Adafruit_BMP280.h

Adafruit_BMP280.h: Bloc de notas >					
Archivo Edición Formato Ver Ayuda					
<pre>/*! * @file Adafruit_BMP280.h *</pre>			^		
* This is a library for the Adafruit BMP280 Breakout. *					
* Designed specifically to work with the Adafruit BMP280 Breakout. *					
<pre>* Pick one up today in the adafruit shop! *> https://www.adafruit.com/product/2651 *</pre>					
* These sensors use I2C to communicate, 2 pins are required to interface.					
 * Adafruit invests time and resources providing this open source code, * please support Adafruit andopen-source hardware by purchasing products * from Adafruit! * 					
* K.Townsend (Adafruit Industries) *					
* BSD license, all text above must be included in any redistribution */					
#ifndefBMP280_H #defineBMP280_H					
// clang-format off #include <arduino.h></arduino.h>					
<pre>#include <adafruit_sensor.h> #include <adafruit_i2cdevice b=""></adafruit_i2cdevice></adafruit_sensor.h></pre>					
<pre>#include <adafruit_spidevice.h> // clang-format on</adafruit_spidevice.h></pre>					
			~		

Buscaremos la siguiente texto:

Buscar			×
<u>B</u> uscar:	BMP280_ADDRESS		<u>B</u> uscar siguiente
Coincidir	<u>m</u> ayúsculas y minúsculas	Dirección O A <u>r</u> iba	Cancelar
Ajuste au	tomático		

#define BMP280_ADDRESS (0x77) /**< The default I2C address for the sensor. */ Cambiaremos 77 por el 76 y guardamos los cambios. Ya podemos empezar a programar.

1	<pre>#include <adafruit_sensor.h></adafruit_sensor.h></pre>
2	<pre>#include <adafruit_bmp280.h></adafruit_bmp280.h></pre>
3	<pre>#include <wire.h></wire.h></pre>
4	
5	Adafruit_BMP280 bmp;
6	
7	float TEMPERATURA;
8	float PRESION;
9	
10	<pre>void setup() {</pre>
11	<pre>Serial.begin(9600);</pre>
12	<pre>Serial.println("Iniciando:");</pre>
13	<pre>if (!bmp.begin()) {</pre>
14	<pre>Serial.println("BMP280 no encontrado !");</pre>
15	<pre>while(1);</pre>
16	}
17	}
18	
19	<pre>void loop() {</pre>
20	<pre>TEMPERATURA = bmp.readTemperature();</pre>
21	<pre>PRESION = bmp.readPressure()/100;</pre>
22	<pre>Serial.print("Temperatura: ");</pre>
23	<pre>Serial.print(TEMPERATURA);</pre>
24	<pre>Serial.print(" C ");</pre>
25	
26	<pre>Serial.print("Presion: ");</pre>
27	<pre>Serial.print(PRESION);</pre>
28	<pre>Serial.println(" hPa");</pre>
29	
30	delay(5000);
31	}

Este será el resultado:

Salida Monitor Serie × Message (Enter to send message to 'Arduino Uno' on 'COM3') Iniciando: Temperatura: 20.80 C Presion: 990.60 hPa Temperatura: 20.79 C Presion: 990.66 hPa Temperatura: 20.80 C Presion: 990.61 hPa Temperatura: 20.79 C Presion: 990.62 hPa

Ahora queremos mostrar la altitud para ello vamos a agregar varias líneas al código.

```
1
     #include <Adafruit Sensor.h>
 2
     #include <Adafruit BMP280.h>
 3
     #include <Wire.h>
 4
 5
     Adafruit_BMP280 bmp;
 6
 7
     float TEMPERATURA;
     float PRESION, P0;
 8
 9
10
     void setup() {
11
       Serial.begin(9600);
12
13
       Serial.println("Iniciando:");
14
       if ( !bmp.begin() ) {
         Serial.println("BMP280 no encontrado !");
15
16
         while(1);
17
18
       P0 = bmp.readPressure()/100;
19
     ß
20
21
     void loop() {
       TEMPERATURA = bmp.readTemperature();
22
23
       PRESION = bmp.readPressure()/100;
       Serial.print("Temperatura: ");
24
       Serial.print(TEMPERATURA);
25
26
       Serial.print(" C ");
27
28
       Serial.print("Presion: ");
       Serial.print(PRESION);
29
30
       Serial.print(" hPa");
31
32
       Serial.print(" Altititud aprox.: ");
33
       Serial.print(bmp.readAltitude(P0));
       Serial.println(" m. ");
34
35
       delay(5000);
36
37
```

Ahora movemos nuestra placa hacia arriba y veremos como nos está indicando la altitud de la placa con respecto a la posición anterior.

Salida Monitor Serie ×

Message (Enter to send message to 'Arduino Uno' on 'COM3')

```
Temperatura: 20.63 C Presion: 990.62 hPa Altititud aprox.: -0.23 m.
Temperatura: 20.62 C Presion: 990.60 hPa Altititud aprox.: -0.06 m.
Temperatura: 20.62 C Presion: 990.57 hPa Altititud aprox.: 0.15 m.
Temperatura: 20.62 C Presion: 990.58 hPa Altititud aprox.: 0.06 m.
```

Capítulo 33.- Lector tarjeta MicroSD bus SPI (y data-logger DHT11)







Tiene que estar formateado en un formato de FAT o FAT32.

```
Vamos a programar:
```

```
#include <SPI.h>
#include <SD.h>
#define SSpin 10
File archivo;
void setup() {
  Serial.begin(9600);
 Serial.println("Inicializando tarjeta...");
  if (!SD.begin(SSpin)){
    Serial.println("Fallo de inicialización!");
    return; //Salimos de la función
  }
  Serial.println("Inicializacion correcta");
  archivo = SD.open("prueba.txt", FILE_WRITE); // Si no existe lo crea.
  if (archivo) { // Si se ha creado el archivo.
    archivo.println("Probando 1, 2, 3"); //Escribimos el siguiente texto.
    Serial.println("Escribiendo en archivo prueba.txt...");
    archivo.close(); // Cerramos el archivo.
    Serial.println("Escritura correcta");
  } else {
    Serial.println("Error en apertura de prueba.txt");
  }
  archivo = SD.open("prueba.txt"); // Abrimos el archivo en modo lectura.
  if (archivo){ //Si lo hace correctamente
    Serial.println("Contenido de prueba.txt:");
    while (archivo.available()){
      Serial.write(archivo.read()); // Lee carácter a carácter.
    }
    archivo.close(); //Cierra el archivo.
  } else {
    Serial.println("Error en la apertura de prueba.txt");
  }
}
void loop() {
   // Está vacío.
}
```

Este será el resultado en Monitor serie.

```
Salida Monitor Serie 🗙
```

Mensaje (Ctrl + Enter para enviar el mensaje a 'Arduino Uno' en 'COM3')

```
Inicializando tarjeta...
Inicializacion correcta
Escribiendo en archivo prueba.txt...
Escritura correcta
Contenido de prueba.txt:
Probando 1, 2, 3
```

Cerramos el monitor serie y lo volvemos a abrir.



Al ejecutarlo de nuevo agrega una nueva línea al final.

Vamos a agregar el sensor de humedad.



```
Vamos a programar:
```

#include <TinyDHT.h>

```
#include <Adafruit Sensor.h>
#include <SPI.h>
#include <SD.h>
#define SENSOR 4
int TEMPERATURA;
int HUMEDAD;
#define SSpin 10
File archivo;
DHT dht(SENSOR, DHT11);
void setup() {
  Serial.begin(9600);
  Serial.println("Inicializando tarjeta...");
  if (!SD.begin(SSpin)){
    Serial.println("Fallo de inicialización!");
    return;
  }
  Serial.println("Inicializacion correcta");
  archivo = SD.open("datos.txt", FILE_WRITE);
  if (archivo) {
    for (int i = 1; i < 31; i++ ){</pre>
      TEMPERATURA = dht.readTemperature();
      HUMEDAD = dht.readHumidity();
      archivo.print(i);
      archivo.print(",");
      archivo.print(TEMPERATURA);
      archivo.print(",");
      archivo.println(HUMEDAD);
      Serial.print(i);
      Serial.print(",");
      Serial.print(TEMPERATURA);
      Serial.print(",");
      Serial.println(HUMEDAD);
      delay(1000);
    }
    archivo.close();
    Serial.println("Escritura correcta");
  } else {
    Serial.println("Error en apertura de datos.txt");
```

```
}
}
void loop() {
}
```

Este será el resultado del monitor serial:

Salida Monitor Serie ×
Mensaje (Ctrl + Enter para enviar el mensaje a 'Arduino Uno' en 'COM3')
20/20/00
24,25,95
25,25,95
26,25,95
27,25,95
28,25,95
29,25,95
30,25,95
Escritura correcta

Ahora vamos a leer el fichero en el ordenador.

DATC 🧾	S.TXT: Blo	c de	-		×
<u>A</u> rchivo	<u>E</u> dición	F <u>o</u> rmato	<u>V</u> er	<u>A</u> yuda	
1,25,9	5				^
2,25,9	5				
3,25,9	5				
4,25,9	5				
5,25,9	5				
6,25,9	5				
7,25,9	5				
8,25,9	5				
9,25,9	5				
10,25,9	95				
11,25,9	95				
12,25,	95 55				
17,25,	90 05				
15 25 (35				
16 25 (95				
17.25	95				
18.25.	95				
19.25.	95				
20,25,	95				
21,25,9	95				
22,25,	95				
23,25,9	95				
24,25,9	95				
25,25,	95				
26,25,9	95				
27,25,9	95				
28,25,	95				
29,25,	95				
30,25,9	95				~

Capítulo 34.- Reloj de Tiempo Real (RTC) DS3231 bus 12C





Módulo ONO O ONO O O O O	Reloj de Tiemp DS3231	po Real (F	Panasonic CR 2032	
E MOS E XSK	705.0	Diámetro Espesor	CR2032 20 mm. 3.2 mm.	CR2025 20 mm. 2.5 mm.
Módulo Módulo Mantien Oscilad Error d Batería Salida Salida Salida Salida	e Reloj de Tiem DS3231 e fecha y horario (año, lor interno compensando p le ±2 ppm (unos pocos min d de respaldo mantiene fe directa de oscilador in programable de 1 Hz, 1 H conexión mediante interfa	mes, día, hora, por temperatura nutos al año) echa y horario p terno de 32.768 Khz, 4 Khz, 8 Kh az I2C	RTC) , minutos, segun (TCXO) por al menos 5 a Khz (32 K) nz (SQW)	ndos) años apen-drain

Vamos a crear el circuito:



Vamos a programar, para ello deberemos descargar una librería.

Buscaremos por rtclib.

```
RTClib de Adafruit
                            Works with DS1307.
                            DS3231, PCF8523,
                            PCF8563 on multiple
                            architectures
                            A fork of Jeelab's fantastic
                            RTC library
                            Más información
#include <RTClib.h>
#include <Wire.h>
RTC DS3231 rtc;
void setup() {
  Serial.begin(9600);
  if (! rtc.begin()){
    Serial.println("Modulo no encontrado!");
    while(1);
  }
  //rtc.adjust(DataTimer(2022, 10, 26, 13, 44, 0));
  rtc.adjust(DateTime(__DATE__, __TIME__));
}
void loop() {
  DateTime fecha = rtc.now();
  Serial.print(fecha.day());
  Serial.print("/");
  Serial.print(fecha.month());
  Serial.print("/");
  Serial.print(fecha.year());
  Serial.print(" ");
  Serial.print(fecha.hour());
  Serial.print(":");
  Serial.print(fecha.minute());
  Serial.print(":");
  Serial.println(fecha.second());
  delay(1000);
}
```

Este será el resultado en el monitor serie.

```
      Salida
      Monitor Serie ×

      Mensaje (Ctrl + Enter para enviar el mensaje a 'Arduino Uno' en 'COM3')

      26/10/2022
      13:58:28

      26/10/2022
      13:58:29

      26/10/2022
      13:58:30

      26/10/2022
      13:58:31

      26/10/2022
      13:58:31

      26/10/2022
      13:58:32

      26/10/2022
      13:58:33

      26/10/2022
      13:58:33

      26/10/2022
      13:58:34

      26/10/2022
      13:58:35
```

Comentamos la línea donde se actualiza la hora:

```
//rtc.adjust(DateTime(__DATE__, __TIME__));
```

Vamos a subir de nuevo el programa.



Los datos de la fecha y la hora se mantienen, gracias a su batería de respaldo.

Vamos a realizar un programa que haga la función de alarma y en determinado momento aparezca un mensaje en el monitor serial.

```
#include <RTClib.h>
#include <Wire.h>
RTC_DS3231 rtc;
bool alarma = true;
```

```
void setup() {
  Serial.begin(9600);
  if (! rtc.begin()){
    Serial.println("Modulo no encontrado!");
    while(1);
  }
}
void loop() {
  DateTime fecha = rtc.now();
  if (fecha.hour()== 14 && fecha.minute() == 34){
    if (alarma == true){
      Serial.println("Alarma");
      alarma = false;
    }
  }
  Serial.print(fecha.day());
  Serial.print("/");
  Serial.print(fecha.month());
  Serial.print("/");
  Serial.print(fecha.year());
  Serial.print(" ");
  Serial.print(fecha.hour());
  Serial.print(":");
  Serial.print(fecha.minute());
  Serial.print(":");
  Serial.println(fecha.second());
  delay(1000);
  if (fecha.hour() == 2 && fecha.minute() == 0)
  alarma = true; //reactiva alarma
}
```

```
Salida Monitor Serie x

Mensaje (Ctrl + Enter para enviar el mensaje

26/10/2022 14:33:54

26/10/2022 14:33:55

26/10/2022 14:33:56

26/10/2022 14:33:57

26/10/2022 14:33:58

26/10/2022 14:33:59

Alarma

26/10/2022 14:34:0

26/10/2022 14:34:0
```

Ahora vamos a agregar a nuestro circuito un relé con el fin que desde una hora de inicio hasta una hora final se active el relé para que nos de iluminación durante una hora.



Vamos a programar, yo he realizado este ciclo se realice en un minuto.

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

```
RTC_DS3231 rtc;
bool evento_inicio = true;
bool evento_fin = true;
# define RELE 3
void setup() {
  Serial.begin(9600);
  pinMode(RELE, OUTPUT);
  if (! rtc.begin()){
    Serial.println("Modulo no encontrado!");
    while(1);
  }
}
void loop() {
  DateTime fecha = rtc.now();
  if (fecha.hour()== 15 && fecha.minute() == 7){
    if (evento_inicio == true){
      digitalWrite(RELE, HIGH);
      Serial.println("Rele encendido");
      evento_inicio = false;
```

```
}
}
 if (fecha.hour()== 15 && fecha.minute() == 8){
   if (evento_fin == true){
     digitalWrite(RELE, LOW);
     Serial.println("Rele apagadp");
     evento_fin = false;
   }
 }
 Serial.print(fecha.day());
 Serial.print("/");
 Serial.print(fecha.month());
 Serial.print("/");
 Serial.print(fecha.year());
 Serial.print(" ");
 Serial.print(fecha.hour());
 Serial.print(":");
 Serial.print(fecha.minute());
 Serial.print(":");
 Serial.println(fecha.second());
 delay(1000);
 if (fecha.hour() == 2 && fecha.minute() == 0)
 evento_inicio = true; //reactiva alarma
 evento_fin = true;
}
```

Capítulo 35.- EEPROM memoria incorporada No Volátil teoría / práctica



μControlador	EEPROM
ATmega 328p UNO, Nano, Pro Mini	1024 bytes
ATmega 2560 Mega	4096 bytes



Vamos a programas, vamos a necesitar un librería especifica que ya viene en el IDE.

🔤 sket	🥯 sketch_oct27a Arduino IDE 2.0.0						
Archivo	Archivo Edit Sketch Herramientas Help						
		🕂 Arduino Uno 👻					
	sketch_o	ct27a.ino					
	1	<pre>#include <eeprom.h></eeprom.h></pre>					
	2						
12	3	<pre>void setup() {</pre>					
	4	Serial.begin(9600);					
nh	5						
	6	<pre>Serial.print("Capacidadd de memoria: ");</pre>					
	7	<pre>Serial.println(EEPROM.length());</pre>					
	8	<pre>Serial.println(" ");</pre>					
· · ·	9						
	10	<pre>Serial.print("Valor almacenado en direccion 0: ");</pre>					
Q.	11	<pre>Serial.println(EEPROM.read(0)); // read(direccion)</pre>					
	12						
	13	Serial.print("Almacenando numero 39 en direccion 0");					
	14	EEPROM.write(0, 39); // wrtite(direccion, valor)					
	15	}					
	16						
	1/						
	18	// Nada por aqui					
	19	}					

Cuando ejecutemos el programa veremos como podemos ver la información de la celda, modificar su valor y una vez desconectado el Arduino, con otro programa volver a leer la información.

 Salida
 Monitor Serie ×

 Mensaje (Ctrl + Enter para enviar el mensaje a 'Arduino Uno' en 'COM3')

 Capacidadd de memoria: 1024

 Valor almacenado en direccion 0: 255

 Almacenando numero 39 en direccion 0



Conectamos de nuevo la placa y como ahora lo que queremos es leer la información de la celda O de nuestra placa, vamos a ejecutar este código:

Salida Monitor Serie ×
Mensaje (Ctrl + Enter para enviar el mensaje a 'Arc
Valor almacenado en direccion 0: 39

Como podrás observar el valor no se ha perdido.

• int: 2 bytes	(entero con o sin signo)
• float: 4 bytes	(punto decimal)
• char [n]: n bytes	(cadena de caracteres)

Que pasa si lo que queremos almacenar es un número de tipo int, float o char.

•	get(dirección	de	inicio,	dato)	Leer
•	put(dirección	de	inicio,	dato) 🕞	Escribir

Con estas dos funciones podemos almacenar valores sin tener que preocuparnos del número de celdas que va a utilizar.



evitando ciclos de escritura innecesarios

🔤 ske	tch_oct27a	Arduino IDE 2.0.0
Archivo	Edit Ske	etch Herramientas Help
Ø	€	🖞 Arduino Uno 👻
	sketch_o	ct27a.ino
	1	<pre>#include <eeprom.h></eeprom.h></pre>
	2	
12	3	float temperatura = 27.50;
	4	<pre>char cadena[] = "Hola esta es una prueba";</pre>
nh	5	
	6	<pre>void setup() {</pre>
	7	Serial.begin(9600);
	8	
· ~	9	<pre>EEPROM.put(0, temperatura);</pre>
	10	<pre>EEPROM.put(10, cadena);</pre>
Q	11	
	12	Serial.println("Valor de punto flotante en direccion 0:");
	13	<pre>Serial.println(EEPROM.get(0, temperatura));</pre>
	14	
	15	Serial.println(" ");
	16	Contral and the device of the outer sector sec
	1/	Serial.println("Valor de la cadena en dirección 10:");
	18	Serial.printin(EEPROM.get(10, cadena));
	19	3
	20	word loop() (
	21	// Nada nor aqui
	23	}
	25	J

Vamos a ejecutar el programa, este será el resultado:

```
      Salida
      Monitor Serie ×

      Mensaje (Ctrl + Enter para enviar el mensaje a 'Arduino Uno' en 'COM3')

      Valor de punto flotante en direccion 0:

      27.50

      Valor de la cadena en direccion 10:

      Hola esta es una prueba
```



Capítulo 36.- RFID RC522 kit lector, tarjeta, llavero 13.56 Mhz



También en lugar de la tarjeta se puede utilizar un llavero o etiquetas autoadhesivas.

Suponiendo que estos no vienen soldados.

Vamos a realizar el circuito:



Vamos a programar, para ello debemos de descargar una librería, buscaremos por rc522.

MFRC522 de GithubCommunity Read/Write a RFID Card or Tag using the ISO/IEC 14443A/MIFARE interface. Arduino RFID Library for MFRC522 (SPI) Más información

Ahora del menú Archivo seleccionaremos Ejemplos de este MFR522 y de este DumpInfo.

Dui Dui	mpinfo Ardu	uino IDE 2.0.0	- 0	×
Archivo	Edit Sket	tch Herramientas Help		
	Ə 🔊	Arduino Uno 🔹	\checkmark	·Q.
	DumpInfo.	ino		
	10 11 12 13 14	* When the Arduino and the MFRC522 module are connected (see the pin layout below), load this sketch into * then verify/compile and upload it. To see the output: use Tools, Serial Monitor of the IDE (hit Ctrl+Sh * you present a PICC (that is: a RFID Tag or Card) at reading distance of the MFRC522 Reader/PCD, the ser * will show the ID/UID, type and any data blocks it can read. Note: you may see "Timeout in communication * when removing the PICC from reading distance too early.	Arduino IDE ft+M). When ial output " messages	
∆ ¢	16 17 18 19 20 21 22 23	* If your reader supports it, this sketch/program will read all the PICCs presented (that is: multiple ta * So if you stack two or more PICCs on top of each other and present them to the reader, it will first ou * details of the first and then the next PICC. Note that this may take some time as all data blocks are d * keep the PICCs at reading distance until complete. * @license Released into the public domain. * * Typical pin layout used:	g reading). tput all umped, so	
	24 25 26 27	* MFRC522 Arduino Arduino Arduino Arduino * Reader/PCD Uno/101 Mega Nano v3 Leonardo/Micro Pro Micro * Signal Pin Pin Pin Pin Pin Pin		
	28 29 30 31 32 33 34	* * RST/Reset RST 9 5 D9 RESET/ICSP-5 RST * SPI SS SDA(SS) 10 53 D10 10 10 * SPI MOSI MOSI 11 / ICSP-4 51 D11 ICSP-4 16 * SPI MISO MISO 12 / ICSP-1 50 D12 ICSP-1 14 * SPI SCK SCK 13 / ICSP-3 52 D13 ICSP-3 15 * *		
	35 36 37	* More pin layouts for other boards can be found here: https://github.com/miguelbalboa/rfid#pin-layout */	Uno [no conectad	o] 🗘
🔤 Du	mpinfo Ardu	uino IDE 2.0.0	- 0	×
Archivo	Edit Sket	tch Herramientas Help		_
	Ə 🔊	Arduino Uno 👻	√	·Q
P	DumpInfo.	lino		
1	37 38 39 40	<pre>#include <spi.h> #include <mfrc522.h></mfrc522.h></spi.h></pre>		
lik	41 42 43	#define RST_PIN 9 // Configurable, see typical pin layout above #define SS_PIN 10 // Configurable, see typical pin layout above		
\$ ₩	44 45 46	<pre>MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance void setup() {</pre>		
Q	47 48 49 50 51 52 53 54 54	<pre>Serial.begin(9600); // Initialize serial communications with the PC while (!Serial); // Do nothing if no serial port is opened (added for Arduinos based on ATMEGA32U4) SPI.begin(); // Init SPI bus mfrc522.PCD_Init(); // Init MFRC522 delay(4); // Optional delay. Some board do need more time after init to be ready, see Readme mfrc522.PCD_DumpVersionToSerial(); // Show details of PCD - MFRC522 Card Reader details Serial.println(F("Scan PICC to see UID, SAK, type, and data blocks")); }</pre>		
	55 56 57 58 59 60 61 62	<pre>void loop() { // Reset the loop if no new card present on the sensor/reader. This saves the entire process when idle. if (! mfrc522.PICC_IsNewCardPresent()) { return; } // Select one of the cards</pre>		
	63 64 65 66 67 68	<pre>if (! mfrc522.PICC_ReadCardSerial()) { return; } // Dump debug info about the card; PICC_HaltA() is automatically called mfrc522.PICC_DumpToSerial(&(mfrc522.uid));</pre>		
	69 70	}		
		Lín. 1. col. 1 UTF-8 🔮 Arduino	Uno Ino conectad	οιΩ

```
#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN
                      9 // Configuración del pin 9
#define SS PIN
                       10 // Configuración del pin 10
MFRC522 mfrc522(SS_PIN, RST_PIN); // Inicializa el módulo lector
void setup() {
  Serial.begin(9600); // Inicializa la comunicación monitor serie
                    // No haga nada si no se abre ningún puerto serie
 while (!Serial);
(agregado para Arduino basado en ATMEGA32U4)
  SPI.begin();
                  // Inicializa SPI bus
 mfrc522.PCD_Init(); // Inicializa MFRC522
               // Retraso opcional. Algunas placas necesitan más
  delay(4);
tiempo después de la inicialización para estar listas.
 mfrc522.PCD_DumpVersionToSerial(); // Mostrar detalles de PCD -
Detalles del lector de tarjetas MFRC522
  Serial.println(F("Escanne PICC para ver UID, SAK, tipo y bloque de
datos..."));
}
void loop() {
  // Restablezca el bucle si no hay ninguna tarjeta nueva presente en el
sensor/lector. Esto guarda todo el proceso cuando está inactivo.
  if ( ! mfrc522.PICC IsNewCardPresent()) {
   return; // Salimos del loop y entramos de nuevo
  }
 // Seleccione una de las tarjetas
  if ( ! mfrc522.PICC ReadCardSerial()) {
   return; // Salimos del loop y entramos de nuevo
  }
 // Volcar información de depuración sobre la tarjeta: PICC HaltA() se
llama automáticamente
 mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
}
                               PCD
                              Proximity Coupling Device
                               (módulo lector)
                               PICC
                               Proximity Integrated
                               Circuit Chip
```

```
176
```

(tarjeta, llavero)

Subimos el programa a Arduino y rápidamente abrimos el monitor serie.

```
Salida Monitor Serie ×
Mensaje (Ctrl + Enter para enviar el mensaje a 'Arduino Uno' en 'COM'
Firmware Version: 0x92 = v2.0
Scan PICC to see UID, SAK, type, and data blocks...
```

Escanee PICc para ver UDI, SAK, tipo y bloque de datos.

Recuerda que tenemos que aproximar la tarjeta. Este será el resultado en el monitor serie.

```
Salida
  Monitor Serie 🗙
Mensaje (Ctrl + Enter para enviar el mensaje a 'Arduino Uno' en 'COM3')
Firmware Version: 0x92 = v2.0
Scan PICC to see UID, SAK, type, and data blocks...
Card UID: 89 21 B8 98
Card SAK: 08
PICC type: MIFARE 1KB
Sector Block 0 1 2 3
            4 5 6 7 8 9 10 11 12 13 14 15 AccessBits
   63 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
1.5
    59 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
14
    [000]
    [ 0 0 0 1
    55 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF
                               [001]
13
    [000]
    [000]
    [000]
```

Si aproximas el llavero sucederá lo mismo. Card UID: Identificador único en formato hexadecimal. Card SAK: Es un valor que identifica el tipo de circuito integrado en la tarjeta.

PICC type: Nombre comercial del circuito de la tarjeta. (MIFARE 1KB).

 Sector Block
 0
 1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 AccessBits

 15
 63
 00
 00
 00
 00
 00
 FF
 07
 80
 69
 FF
 FF

						B	yte	Num	nber	with	in a	Blog	k					
Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Description
15	3			Ke	y A			A	cces	ss Bi	ts			Ke	yВ			Sector Trailer 15
	2																	Data
	1																	Data
	0																	Data

Está organizada en 16 sectores. En el O encontramos:

0	3	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	Г	0	0	1	1
	2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0	0	0]
	1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0	0	0]
	0	89	21	B8	98	88	08	04	00	62	63	64	65	66	67	68	69	[0	0	0]
0		3	1		٢	(ey A			Acc	cess B	its		ł	Key B				Sect	or T	rai	ler	0
		2				1											ſ	Data				
		1															ſ	Data				
		0						M	anufa	acture	r Dat	a					r	Man	ufa	ctur	erl	Block

En la posición 0 encontramos los datos del fabricante.

Puto	0	4	2	2	4	5	C	7	0	0	10	44	10	10	4.4	15
Dyte	U	1	2	3	4	5	0	'	0	9	10	11	12	15	14	15
	12	NU	JID			L			Mai	nufac	turer	Data			001a	an01

NUID: Identificador único en este ejemplo es 89 21 B8 98 para nuestra tarjeta y 49 B9 ED A3 para nuestro llavero.

Ahora vamos a realizar un programa para que solo nos muestra el NUID.

```
#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN 9
#define SS_PIN 10
MFRC522 mfrc522(SS_PIN, RST_PIN);
void setup() {
   Serial.begin(9600);
   SPI.begin();
   mfrc522.PCD_Init();
}
```

```
void loop() {
 if(! mfrc522.PICC_IsNewCardPresent())
  return;
 if(! mfrc522.PICC_ReadCardSerial())
  return;
  Serial.print("UID:");
  for (byte i = 0; i < mfrc522.uid.size; i++){ // mfrc522.uid.sice es su</pre>
tamaño.
    if(mfrc522.uid.uidByte[1] < 0x10){</pre>
      Serial.print(" 0");
    }
      else{
      Serial.print(" ");
    }
    Serial.print(mfrc522.uid.uidByte[i], HEX);
  }
  Serial.println();
  mfrc522.PICC_HaltA(); // Llama a la función para decir a la tarjeta que
                         // ya no necesitamos más información.
}
```

Vamos a subir el programa, activemos el monitor serie y aproximemos la tarjeta y después el llavero.

Este será el resultado:



En nuestro caso el primero es el de la tarjeta y el segundo es el del llavero.

Vamos a reutilizar el primer programa para este ejercicio:

```
#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN 9
#define SS_PIN 10
MFRC522 mfrc522(SS_PIN, RST_PIN);
```
```
byte LecturaUID[4];
byte Usuario1[4] = {0x89, 0x21, 0xB8, 0x98}; // Tarjeta
byte Usuario2[4] = {0x49, 0xB9, 0xED, 0xA3}; // Llavero
void setup() {
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
  Serial.println("Listo");
}
void loop() {
 if(! mfrc522.PICC_IsNewCardPresent())
  return;
 if(! mfrc522.PICC_ReadCardSerial())
  return;
  Serial.print("UID:");
  for (byte i = 0; i < mfrc522.uid.size; i++){ // mfrc522.uid.sice es su</pre>
tamaño.
    if(mfrc522.uid.uidByte[1] < 0x10){</pre>
      Serial.print(" 0");
    }else {
      Serial.print(" ");
    }
    Serial.print(mfrc522.uid.uidByte[i], HEX);
    LecturaUID[i] = mfrc522.uid.uidByte[i];
    }
    Serial.print("\t"); // Espacio de tabulación.
    if(comparaUID(LecturaUID, Usuario1))
      Serial.println("Bienvenido Usuario 1");
    else if (comparaUID(LecturaUID, Usuario2))
      Serial.println("Bienvenido Usuario 2");
    else
      Serial.println("No te concozco");
   mfrc522.PICC_HaltA();
}
boolean comparaUID(byte lectura[], byte usuario[])
{
  for (byte i=0; i < mfrc522.uid.size; i++)</pre>
  {
    if(lectura[i] != usuario[i])
    return(false);
```

```
}
return(true);
}
```

Vamos a subir el programa y vamos a aproximar la tarjeta y después el llavero.

Este será el resultado:

```
      Salida
      Monitor Serie ×

      Mensaje (Ctrl + Enter para enviar el mensaje a 'Arduino Uno' en 'COM3')

      Listo

      UID: 89 21 B8 98
      Bienvenido Usuario 1

      UID: 49 B9 ED A3
      Bienvenido Usuario 2
```

Hemos puesto una segunda tarjeta de acceso a una zona restringida y este ha sido el mensaje:

Salida	Monit	or Serie 🗙				
Mensaje (Ctrl + Enter para enviar el mensaje a 'Arduino Uno' en 'COM3')						
Listo						
UID: 8	9 21 E	38 98	Bienvenido Usuario 1			
UID: 4	9 B9 E	ED A3	Bienvenido Usuario 2			
UID: F	5 E5 E	24 89	No te concozco			
<u> </u>						

Si no tienes más tarjetas de acceso modifica el valor de la tarjeta o el llavero sube de nuevo el programa y verás como ahora te dirá no te conozco.

```
byte LecturaUID[4];
byte Usuario1[4] = {0x89, 0x21, 0xB8, 0x98}; // Tarjeta
byte Usuario2[4] = {0x49, 0xB9, 0xED, 0xA3}; // Llavero
```

Cambia alguno de los valores que te señalo. El valor tiene que ser Hexadecimal.

Capítulo 37.- Matriz LED 8x8 con MAX7219 y LedControl



× × × ×

** ** **

1 N

** **





Vamos al diseño del circuito.



Vamos a la programación, para ello lo primero que tenemos que hacer es instalar una librería.

LedControl de Eberhard Fahle <e.fahle@wayoda.org> Versión 1.0.6 INSTALLED The library supports multiple daisychained drivers and supports Led-Matrix displays as well as 7-Segment displays. A library for the MAX7219 and the MAX7221 Led display drivers.

Este primer programa tiene como objetivo ir encendiendo y apagando cada uno de los leds de la matriz, desde el primer elemento hasta el último.

🔤 sket	🥯 sketch_oct26a Arduino IDE 2.0.0					
Archivo	Edit Ske	tch Herramientas Help				
	€ 🕞	Arduino Uno 👻				
P	sketch_o	ct26a.ino				
	1	<pre>#include <ledcontrol.h></ledcontrol.h></pre>				
	2	<pre>LedControl lc=LedControl(11,13,10,1);</pre>				
12	3					
	4					
Infly	5	<pre>void setup() {</pre>				
	6	<pre>lc.shutdown(0,false);</pre>				
	7	<pre>lc.setIntensity(0,4);</pre>				
	8	<pre>lc.clearDisplay(0);</pre>				
~	9	}				
	10					
Q	11	<pre>void loop() {</pre>				
	12	<pre>for(int fila = 0; fila <8; fila++){</pre>				
	13	<pre>for(int columna = 0; columna < 8; columna++){</pre>				
	14	<pre>lc.setLed(0,fila,columna,true);</pre>				
	15	delay(100);				
	16	<pre>IC.SetLed(0,fila,columna,false); delew(400);</pre>				
	1/	deray(100);				
	18					
	20) <i>3</i> \				



En LedControl el número de dispositivos si solo tienes uno será 1 pero puedes tener hasta 8, de este modo el segundo sería el 2 el tercero el 3 y así sucesivamente.

En Shutdown el número de módulo empieza por 0 y el 8 tiene el número 7, y no deseamos que esté apagado.

En SetIntensity indicamos la intensidad del led o brillo del mismo, primero recibe el número del dispositivo y para el brillo un número entre 0 y 15.

En ClearDisplay que borra la matriz, que recibe el número de dispositivo.

En el programa añadimos dos bucles for anidados para que haga un recorrido por todos los leds.

Vamos a modificar el código:



Ahora podrás observar que enciende todos los leds pero no los apaga.

Ahora vamos a ver como mostrar un carácter, digito o letra.



Queremos visualizar el número 0.



Vamos a la programación:

Vamos a comentar la función lc.SetRow(número de dispositivo, fila, valor).

```
setRow(Nro. de dispositivo, fila, valor)
                                         8 bits
#include <LedControl.h>
LedControl lc=LedControl(11,13,10,1);
#define demora 1000
byte cero[8]={
  B00111000, // La letra B significa número binario.
  B01000100,
  B01000100,
  B01000100,
 B01000100,
  B01000100,
  B00111000
};
byte uno[8]={
  B00010000,
  B00110000,
  B00010000,
  B00010000,
  B00010000,
  B00010000,
  B00010000,
  B00111000
};
byte dos[8]={
  B00111000,
  B01000100,
  B00000100,
  B00000100,
  B00001000,
  B00010000,
  B00100000,
  B01111100
};
byte tres[8]={
  B00111000,
  B01000100,
  B00000100,
  B00011000,
  B00000100,
  B00000100,
```

```
B01000100,
  B00111000
};
byte cuatro[8]= { // array que contiene todos los elementos de las
                  // filas necesarias por mostrar el digito cuatro
  B00001000,
  B00011000,
  B00101000,
  B01001000,
  B01001000,
  B01111100,
  B00001000,
  B00001000
};
byte cinco[8]= { // array que contiene todos los elementos de las
B01111100, // filas necesarias por mostrar el digito cinco
  B01000000,
  B01000000,
  B01111000,
  B00000100,
  B00000100,
  B01000100,
  B00111000
};
byte seis[8]= { // array que contiene todos los elementos de las
B00111000, // filas necesarias por mostrar el digito seis
  B01000100,
  B01000000,
  B01111000,
  B01000100,
  B01000100,
  B01000100,
  B00111000
};
byte siete[8]= { // array que contiene todos los elementos de las
  B01111100,
                     // filas necesarias por mostrar el digito siete
  B00000100,
  B00000100,
  B00001000,
  B00010000,
  B00100000,
  B00100000,
  B00100000
};
byte ocho[8]= { // array que contiene todos los elementos de las
```

```
// filas necesarias por mostrar el digito ocho
  B00111000,
  B01000100,
  B01000100,
  B00111000,
  B01000100,
  B01000100,
  B01000100,
  B00111000
};
byte nueve[8]= {
                    // array que contiene todos los elementos de las
                    // filas necesarias por mostrar el digito nueve
  B00111000,
  B01000100,
  B01000100,
  B01000100,
  B00111100,
  B00000100,
 B01000100,
 B00111000
};
void setup() {
  lc.shutdown(0,false);
 lc.setIntensity(0,4);
 lc.clearDisplay(0);
}
void loop() {
 mostrar 0();
                      // llama funcion mostrar 0
                      // demora
 delay(demora);
                      // llama funcion mostrar 1
 mostrar_1();
 delay(demora);
                      // demora
 mostrar_2();
                      // llama funcion mostrar 2
                      // demora
 delay(demora);
 mostrar_3();
                      // llama funcion mostrar 3
 delay(demora);
                      // demora
 mostrar_4();
                      // llama funcion mostrar_4
 delay(demora);
                      // demora
 mostrar_5();
                      // llama funcion mostrar 5
 delay(demora);
                      // demora
 mostrar_6();
                      // llama funcion mostrar_6
  delay(demora);
                      // demora
 mostrar_7();
                      // llama funcion mostrar 7
  delay(demora);
                      // demora
 mostrar_8();
                      // llama funcion mostrar 8
  delay(demora);
                     // demora
 mostrar_9();
                      // llama funcion mostrar_9
  delay(demora);
                      // demora
```

```
void mostrar_0(){ // funcion mostrar_0
for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array</pre>
 {
   lc.setRow(0,i,cero[i]); // establece fila con valor de array cero
en misma posicion
 }
}
void mostrar_1(){ // funcion mostrar_1
 for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array</pre>
 {
   lc.setRow(0,i,uno[i]); // establece fila con valor de array uno en
misma posicion
 }
}
void mostrar_2(){ // funcion mostrar_2
 for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array</pre>
 {
   lc.setRow(0,i,dos[i]); // establece fila con valor de array dos en
misma posicion
 }
}
void mostrar_3(){ // funcion mostrar_3
 for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array</pre>
 {
  lc.setRow(0,i,tres[i]); // establece fila con valor de array tres
en misma posicion
 }
}
void mostrar_4(){ // funcion mostrar_4
 for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array</pre>
 {
  lc.setRow(0,i,cuatro[i]); // establece fila con valor de array
cuatro en misma posicion
 }
}
void mostrar_5(){ // funcion mostrar_5
 for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array</pre>
 {
   lc.setRow(0,i,cinco[i]); // establece fila con valor de array
cinco en misma posicion
 }
}
```

}

```
void mostrar_6(){ // funcion mostrar_6
 for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array</pre>
 {
   lc.setRow(0,i,seis[i]); // establece fila con valor de array seis
en misma posicion
 }
}
void mostrar_7(){ // funcion mostrar_7
 for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array</pre>
 {
   lc.setRow(0,i,siete[i]); // establece fila con valor de array
siete en misma posicion
 }
}
void mostrar_8(){ // funcion mostrar_8
 for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array</pre>
 {
   lc.setRow(0,i,ocho[i]); // establece fila con valor de array ocho
en misma posicion
 }
}
void mostrar_9(){ // funcion mostrar_9
 for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array</pre>
 {
   lc.setRow(0,i,nueve[i]); // establece fila con valor de array
nueve en misma posicion
 }
}
```

En el siguiente código te adjunto el mismo programa que hemos realizado pero mucho más optimizado.

```
#include <LedControl.h>
LedControl lc=LedControl(11,13,10,1);
#define demora 1000
byte numeros[][8]= {
{
    B00111000, // Número 0
    B01000100,
    B00111000
```

```
{
 B00010000, // Número 1
 B00110000,
 B00010000,
 B00010000,
 B00010000,
 B00010000,
 B00010000,
 B00111000
},
{
 B00111000, // Número 2
 B01000100,
 B00000100,
 B00000100,
 B00001000,
 B00010000,
 B00100000,
 B01111100
},
{
 B00111000, // Número 3
  B01000100,
  B00000100,
 B00011000,
 B00000100,
  B00000100,
 B01000100,
 B00111000
},
{
 B00001000, // Número 4
  B00011000,
  B00101000,
  B01001000,
 B01001000,
 B01111100,
 B00001000,
 B00001000
},
{
 B01111100, // Número 5
  B01000000,
```

},

```
B01000000,
  B01111000,
  B00000100,
  B00000100,
  B01000100,
  B00111000
},
{
             // Número 6
  B00111000,
  B01000100,
  B01000000,
  B01111000,
  B01000100,
  B01000100,
  B01000100,
 B00111000
},
{
             // Número 7
  B01111100,
  B00000100,
  B00000100,
  B00001000,
  B00010000,
  B00100000,
  B00100000,
  B00100000
},
{
              // Número 8
  B00111000,
  B01000100,
  B01000100,
  B00111000,
  B01000100,
  B01000100,
  B01000100,
  B00111000
},
{
              // Número 9
  B00111000,
  B01000100,
  B01000100,
  B01000100,
  B00111100,
  B00000100,
  B01000100,
```

```
193
```

```
B00111000
}
};
void setup() {
 lc.shutdown(0,false);
 lc.setIntensity(0,4);
 lc.clearDisplay(0);
}
void loop() {
 for(int j=0; j<10; j++){</pre>
     mostrar(numeros[j]); // llama funcion mostrar_0
   delay(demora); // demora
 }
}
void mostrar(byte num[]){ // funcion mostrar_0
 for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array</pre>
 {
      lc.setRow(0,i,num[i]); // establece fila con valor de array cero
en misma posicion
 }
}
```

A continuación vamos a escribir el siguiente código:

```
#include "LedControl.h" // incluye libreria LedControl
LedControl lc=LedControl(11,13,10,1); // crea objeto
#define demora 250 // constante demora con valor de 250
byte flecha_arriba_1[8] = { // array con primer cuadro de animación de
flecha
B00000000,
B0011000,
B0011100,
B0111110,
B11011011,
B00011000,
B00011000,
B00011000,
B00011000
};
byte flecha_arriba_2[8] = { // array con segundo cuadro de animación de
```

flecha

B00011000,

```
B00111100,
B01111110,
B11011011,
B00011000,
B00011000,
B00011000,
```

};

```
byte flecha_arriba_3[8] = { // array con tercer cuadro de animación de
flecha
    B00111100,
    B0111110,
    B11011011,
    B00011000,
    B00011000,
    B00000000,
    B00000000,
    B00000000
};
byte flecha_arriba_4[8] = { // array con cuarto cuadro de animación de
```

```
flecha
```

```
B01111110,
B11011011,
B00011000,
B00011000,
B00011000,
B00000000,
B00000000,
B00011000
```

```
};
```

```
byte flecha_arriba_5[8] = { // array con quinto cuadro de animación de
flecha
B11011011,
B00011000,
B00011000,
B00000000,
B00000000,
B000011000,
B00011000,
B00011000,
B00111100
};
```

byte flecha_arriba_6[8] = { // array con sexto cuadro de animación de flecha B00011000, B00011000,

```
B00011000,
B00000000,
B00000000,
B00011000,
B00111100,
B01111110
};
```

```
byte flecha_arriba_7[8] = { // array con séptimo cuadro de animación de
flecha
B00011000,
B00011000,
B0000000,
B00011000,
B0011100,
B0111110,
B11011011
};
```

```
byte flecha_arriba_8[8] = { // array con octavo cuadro de animación de
flecha
  B00011000,
  B00000000,
  B00000000,
  B00011000,
  B00111100,
  B01111110,
  B11011011,
  B00011000
};
byte flecha_arriba_9[8] = { // array con noveno cuadro de animación de
flecha
  B00000000,
  B00000000,
  B00011000,
  B00111100,
  B01111110,
  B11011011,
  B00011000,
  B00011000
};
void setup() {
 lc.shutdown(0,false); // enciende la matriz
lc.setIntensity(0,4); // establece brillo
  lc.clearDisplay(0); // blanquea matriz
}
```

```
void loop(){
                     // llama función mostrar 1
 mostrar_1();
 delay(demora);
                     // demora
 mostrar_2();
                     // llama función mostrar_2
 delay(demora);
                     // demora
 mostrar_3();
                     // llama función mostrar_3
                     // demora
 delay(demora);
                     // llama función mostrar 4
 mostrar 4();
 delay(demora);
                     // demora
                     // llama función mostrar 5
 mostrar_5();
 delay(demora);
                     // demora
 mostrar_6();
                     // llama función mostrar 6
 delay(demora);
                     // demora
                     // llama función mostrar_7
 mostrar_7();
 delay(demora);
                    // demora
 mostrar_8();
                     // llama función mostrar_8
 delay(demora);
                    // demora
                     // llama función mostrar 9
 mostrar_9();
                     // demora
 delay(demora);
}
```

```
void mostrar_1(){ // función mostrar_1
for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array
{
    lc.setRow(0,i,flecha_arriba_1[i]); // establece fila con valor de
array flecha_arriba_1
}
}</pre>
```

```
void mostrar_2(){ // función mostrar_2
for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array
{
    lc.setRow(0,i,flecha_arriba_2[i]); // establece fila con valor de
array flecha_arriba_2
}
</pre>
```

```
void mostrar_3(){ // función mostrar_3
for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array
{
    lc.setRow(0,i,flecha_arriba_3[i]); // establece fila con valor de
array flecha_arriba_3
}</pre>
```

```
void mostrar_4(){ // función mostrar_4
for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array
{</pre>
```

```
lc.setRow(0,i,flecha_arriba_4[i]); // establece fila con valor de
array flecha_arriba_4
}
}
```

```
void mostrar_5(){ // función mostrar_5
  for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array
   {
      lc.setRow(0,i,flecha_arriba_5[i]); // establece fila con valor de
      array flecha_arriba_5
    }
}</pre>
```

```
void mostrar_6(){ // función mostrar_6
for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array
{
    lc.setRow(0,i,flecha_arriba_6[i]); // establece fila con valor de
array flecha_arriba_6
}
}</pre>
```

```
void mostrar_7(){ // función mostrar_7
for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array
{
    lc.setRow(0,i,flecha_arriba_7[i]); // establece fila con valor de
array flecha_arriba_7
}
}</pre>
```

```
void mostrar_8(){ // función mostrar_8
for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array
{
    lc.setRow(0,i,flecha_arriba_8[i]); // establece fila con valor de
array flecha_arriba_8
}
}</pre>
```

```
void mostrar_9(){ // función mostrar_9
for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array
{
    lc.setRow(0,i,flecha_arriba_9[i]); // establece fila con valor de
array flecha_arriba_9
}</pre>
```

Vamos a optimizarlo:

```
#include "LedControl.h" // incluye libreria LedControl
```

```
LedControl lc=LedControl(11,13,10,1); // crea objeto
#define demora 250 // constante demora con valor de 250
byte flecha[][8] = {
{ // array con primer cuadro de animación de flecha_arriba_1
 B00000000,
 B00011000,
  B00111100,
 B01111110,
 B11011011,
 B00011000,
 B00011000,
 B00011000
},
{ // array con segundo cuadro de animación de flecha_arriba_2
 B00011000,
 B00111100,
  B01111110,
  B11011011,
 B00011000,
  B00011000,
 B00011000,
 B00000000
},
{ // array con tercer cuadro de animación de flecha_arriba_3
  B00111100,
 B01111110,
  B11011011,
 B00011000,
  B00011000,
  B00011000,
  B00000000,
 B00000000
},
{ // array con cuarto cuadro de animación de flecha_arriba_4
  B01111110,
 B11011011,
  B00011000,
  B00011000,
  B00011000,
  B00000000,
  B00000000,
  B00011000
},
```

```
{ // array con quinto cuadro de animación de flecha_arriba_5
  B11011011,
  B00011000,
  B00011000,
  B00011000,
  B00000000,
  B00000000,
  B00011000,
  B00111100
},
{ // array con sexto cuadro de animación de flecha_arriba_6
  B00011000,
  B00011000,
  B00011000,
  B00000000,
  B00000000,
  B00011000,
  B00111100,
  B01111110
},
{ // array con séptimo cuadro de animación de flecha_arriba_7
  B00011000,
  B00011000,
  B00000000,
  B00000000,
  B00011000,
  B00111100,
  B01111110,
  B11011011
},
{ // array con octavo cuadro de animación de flecha_arriba_8
  B00011000,
  B00000000,
  B00000000,
  B00011000,
  B00111100,
  B01111110,
  B11011011,
  B00011000
},
{ // array con noveno cuadro de animación de flecha_arriba_9
  B00000000,
  B00000000,
  B00011000,
  B00111100,
```

```
B01111110,
  B11011011,
  B00011000,
  B00011000
}
};
void setup() {
  lc.shutdown(0,false); // enciende la matriz
lc.setIntensity(0,4); // establece brillo
  lc.clearDisplay(0); // blanquea matriz
}
void loop(){
  for(int j=0; j<9; j++){</pre>
    mostrar(flecha[j]);
    delay(demora); // demora
  }
}
void mostrar(byte num[]){ // funcion mostrar_1
  for (int i = 0; i < 8; i++) // bucle itera 8 veces por el array</pre>
  {
    lc.setRow(0,i,num[i]);
  }
```

Ahora para finalizar vamos a ver como encadenar un segundo módulo:



Así será el código:

}

```
🤓 sketch_oct26c | Arduino IDE 2.0.0
Archivo Edit Sketch Herramientas Help
      Ð
 \checkmark
                 Arduino Uno
                                            •
      sketch_oct26c.ino
              include "LedControl.h"
                                          // incluye libreria LedControl
          1
          2
f_)
          3
              LedControl lc=LedControl(11,13,10,2); // Ahora tenemos dos módulos.
          4
          5
              void setup() {
          6
               lc.shutdown(0,false);
          7
                lc.setIntensity(0,4);
          8
÷
                lc.clearDisplay(0);
         9
         10
                                         // enciende la matriz segundo módulo
                lc.shutdown(1,false);
         11
                lc.setIntensity(1,4);
                                           // establece brillo segundo módulo
         12
                lc.clearDisplay(1);
                                         // blanquea matriz segundo módulo
         13
              3
         14
         15
              void loop(){
                for(int fila = 0; fila < 8; fila++) {</pre>
         16
                  for(int columna = 0; columna < 8; columna++) {</pre>
         17
                    lc.setLed(0,fila,columna,true);
         18
         19
                    delay(100);
                    lc.setLed(0,fila,columna,false);
         20
         21
                    delay(100);
         22
                   }
         23
                }
         24
         25
                for(int fila = 0; fila < 8; fila++) {</pre>
                                                                     // itera por las filas
                  for(int columna = 0; columna < 8; columna++) { // itera por las columnas</pre>
         26
                    lc.setLed(1,fila,columna,true);
                                                                     // enciende LED
         27
                                                                     // demora de 100 mseg.
         28
                     delay(100);
         29
                     lc.setLed(1,fila,columna,false);
                                                                     // apaga LED
         30
                    delay(100);
                                                                     // demora de 100 mseg.
         31
                   }
         32
                 }
         33
              }
```

Capítulo 38.- NANO modelos FT232RL y CH340G (con enlaces a drivers)









Para descargar controlador driver para FT232RL

https://ftdichip.com/drivers/vcp-drivers/



Para descargar controlador driver para CH304G

http://www.wch-ic.com/downloads/CH341SER_EXE.html

→ C ▲ No es seguro wch-ic.com/	downloads/CH341SER_EXE.html	e e 🗴 🛪 🗆 🗖
	D Jaami7 Clock Speed	Low Rower Com To ESUA/Mitz 2x CAN 1220FAAC 4x LART 2x OPA 2x CA
	Index / Service / Downloads / Dr	er&Tools / CH341SER.EXE
	popular searches CH563 CH395 CH559 Ethernet US8 extension MCU	CH341SER.EXE The scope of application version upload time size
	DataSheet	CH3405, CH3407, CH340C, 3.7 2022-05-11 631K8 CH3405, CH3405, CH341A, CH3415, CH341B, CH341A,
	Development Source	CH341U
	Driver&Tools	CH340/CH341 USB to serial port Windows driver, supports Windows XP/Vista/7/8/8.1/10/11/ SERVER 2003/2008/2012/2015/20219/2022 -32/46/bit, Microsoft WHQL Certified, supports USB to 3-line and 9-line serial port. Usef to distribute to user with the endotor.
	Others	
	Video	download
	Contact Us	relation files
		file name file content
		CH34ISERZIP CH340/CH341 USB to serial port Windows driver, supports Windows X9/Vista/7/8/8.1/10/11/ SERVER 2003/2008/2012/2016/2019/2022 - 32/4/bit, Microsoft WHQL Certified, supports USB to 3-line and 3-line serial port.
		CH341SER_LINUX. CH340/CH341 USB to serial port LINUX driver, supports 32/64-bit operation system.

CH341SER.ZIP			
The scope of application	version	upload time	size
CH340G, CH340T, CH340C, CH340N, CH340K, CH340E, CH340B, CH341A, CH341F, CH341T, CH341B, CH341C, CH341U	3.7	2022-05-13	545KB

CH340/CH341 USB to serial port Windows driver, supports Windows XP/Vista/7/8/8.1/10/11/ SERVER 2003/2008/2012/2016/2019/2022 -32/64bit, Microsoft WHQL Certified, supports USB to 3-line and 9-line serial port.





Vamos a descomprimir e instalarlo.



Capítulo 39.- OLED 128x64 12C bicolor basado en SDD1306 por

bus 12C

Ver video







- Excelente contraste y ángulo de visión
- Reducido consumo

Vamos a realizar el correspondiente circuito:



Vamos a programar, para ello vamos a instalar dos librerías.

Adafruit GFX Library de Adafruit Versión 1.11.3 INSTALLED Install this library in addition to the display library for your hardware. Adafruit GFX graphics core library, this is the 'core' class that all our other graphics libraries derive from. Adafruit SSD1306 de Adafruit SSD1306 oled driver library for monochrome 128x64 and 128x32 displays SSD1306 oled driver library for monochrome 128x64 and 128x32 displays



Vamos a subir el programa.

Ahora vamos a realizar los siguientes cambios.

```
16 \vee void loop() {
       oled.clearDisplay();
17
       oled.setTextColor(WHITE);
18
19
       oled.setTextSize(1);
       oled.setCursor(0, 0);
20
       oled.print("Texto en amarillo ocupa dos lineas, 16 px");
21
22
       oled.setCursor(0, 16);
23
       oled.print("Hola, este texto ya esta en la zona de color ◀
24
       oled.display();
25
     Н
```

cyan y entran 6 lineas en total, mucha informacion en poco espacio!");



Ahora vamos a mostrar gráficos, para ello vamos a realizar el siguiente programa.

```
#include <Wire.h>
 1
 2
     #include <Adafruit_GFX.h>
     #include <Adafruit SSD1306.h>
 З
 4
 5
     #define ANCHO 128
     #define ALTO 64
 6
 7
     #define OLED_RESET 4
 8
     Adafruit SSD1306 oled(ANCHO, ALTO, &Wire, OLED RESET);
9
10
     void setup() {
11
12
      Wire.begin();
13
       oled.begin(SSD1306_SWITCHCAPVCC, 0x3C);
14
     3
15
```

```
16
     void loop() {
        oled.clearDisplay();
17
        for(int i=1; i<oled.height()/2; i++){</pre>
18
          oled.drawCircle(oled.width()/2, oled.height()/2, i, WHITE);
19
          oled.display();
20
21
          delay(250);
22
        }
23
     }
```

Ira dibujando círculos empezando por uno en el centro pequeño e irán dibujándose círculos más grandes cada vez.



En la última línea agregamos un clearDisplay().

```
16 void loop() {
17 oled.clearDisplay();
18 for(int i=1; i<oled.height()/2; i++){
19 oled.drawCircle(oled.width()/2, oled.height()/2, i, WHITE);
20 oled.display();
21 delay(250);
22 oled.clearDisplay();
23 }</pre>
```

Ahora cuando lo subas podrás apreciar la diferencia.



Ahora cambiar el código de void loop().

16	<pre>void loop() {</pre>
17	<pre>oled.clearDisplay();</pre>
18	<pre>oled.drawRect(20,20,88,44,WHITE);</pre>
19	<pre>oled.setCursor(28,34);</pre>
20	<pre>oled.setTextSize(2);</pre>
21	<pre>oled.setTextColor(WHITE);</pre>
22	<pre>oled.print("ALARMA");</pre>
23	<pre>oled.display();</pre>
24	}






Esta será la programación:



Este será el resultado cuando el sensor detecte el agua.

Salida	Monitor Serie \times
Messag	e (Enter to send message to 'Arduino Uno' on 'COM3')
0	
143	
141	
142	
145	
147	

Vamos a modificar el circuito original:



Vamos a programar:

🥯 sketch_oct28a Arduino IDE 2.0.1								
Archivo Editar Sketch Herramientas Ayuda								
	€ 🕞	Arduino Uno 👻						
Ph	sketch_oct28a.ino							
	1	int SENSOR;						
-	2	#define LED 2						
1-	3							
	4	<pre>void setup() {</pre>						
Illa	5	<pre>pinMode(LED, OUTPUT);</pre>						
	6	}						
	7							
E E	8	void loop() {						
	9	SENSOR = analogRead(A0);						
	10	$if(SENSOR > 0)$ {						
Q	11	digitalWrite(LED, HIGH);						
	12	}						
	13	else {						
	14	<pre>digitalWrite(LED, LOW);</pre>						
	15	}						
	16	delay(1000);						
	17	}						

Ahora pon el sensor en el agua, se tiene que encender el led, para que este se apague tienes que secar el sensor.

Capítulo 41.- Buzzers active y pasivo (zumbadores) tonos y melodías







20 Hz a 20 Khz

67

con el paso de los años la respuesta de frecuencia se 50 Hz a 15 Khz ve reducida al rango promedio:

Vamos a realizar el correspondiente circuito:



Vamos a programar:

🥯 sketch_oct24a Arduino IDE 2.0.0								
Archivo	o Edit Sketch Herramientas Help							
Arduino Uno -								
	sketch_oct24a.ino							
	1	#define PULSADOR 2						
	#define BUZZER_ACTIVO 8							
12	3							
	4	<pre>void setup() {</pre>						
Inh	5	<pre>pinMode(PULSADOR, INPUT_PULLUP);</pre>						
	6	<pre>pinMode(BUZZER_ACTIVO, OUTPUT);</pre>						
	7	}						
	8							
~	9	<pre>void loop() {</pre>						
	10	if(digitalRead(PULSADOR) == LOW){						
Q	11	<pre>for(int i = 0; i < 5; i++){</pre>						
	12	<pre>digitalWrite(BUZZER_ACTIVO, HIGH);</pre>						
	13	delay(500);						
	14	<pre>digitalWrite(BUZZER_ACTIVO, LOW);</pre>						
	15	delay(500);						
	16							
	17	}						
	18	}						

INPUT_PULLUP -> Lo estamos forzando a un nivel alto.



Vamos a crear un circuito que además tiene que tener un potenciómetro.



Consiste en que el potenciómetro de una señal superior a 1000 el zumbador tiene que sonar de lo contrario tiene que estar en silencio.

```
1 int Pot = A0;
2 int Buz = 3;
3 int Potencia;
4 void setup()
5 {
 6
     pinMode(Buz, OUTPUT);
7
   }
8
9 void loop()
10 {
   Potencia = analogRead(Pot);
11
12
    if(Potencia > 1000){
13
      digitalWrite(Buz, HIGH);
    }else{
14
15
       digitalWrite(Buz, LOW);
16
     }
17 }
```







Vamos a diseñar el circuito:



Es circuito es el mismo que el anterior solo hay que cambiar el Buzzer activo por un Buzzer pasivo.





Vamos a programar:

<pre>#define</pre>	NOTE_B0	31	//	notas	у	frecuencias
<pre>#define</pre>	NOTE_C1	33				
<pre>#define</pre>	NOTE_CS1	35				
<pre>#define</pre>	NOTE_D1	37				
<pre>#define</pre>	NOTE_DS1	39				
<pre>#define</pre>	NOTE_E1	41				
<pre>#define</pre>	NOTE_F1	44				
<pre>#define</pre>	NOTE_FS1	46				
<pre>#define</pre>	NOTE_G1	49				
<pre>#define</pre>	NOTE_GS1	52				
<pre>#define</pre>	NOTE_A1	55				
<pre>#define</pre>	NOTE_AS1	58				
<pre>#define</pre>	NOTE_B1	62				
<pre>#define</pre>	NOTE_C2	65				
<pre>#define</pre>	NOTE_CS2	69				
<pre>#define</pre>	NOTE_D2	73				
<pre>#define</pre>	NOTE_DS2	78				
<pre>#define</pre>	NOTE_E2	82				
<pre>#define</pre>	NOTE_F2	87				
<pre>#define</pre>	$NOTE_FS2$	93				
<pre>#define</pre>	NOTE_G2	98				
<pre>#define</pre>	NOTE_GS2	104	ŀ			
<pre>#define</pre>	NOTE_A2	110)			
<pre>#define</pre>	NOTE_AS2	117	,			
<pre>#define</pre>	NOTE_B2	123	3			
<pre>#define</pre>	NOTE_C3	131	-			
<pre>#define</pre>	NOTE_CS3	139)			
<pre>#define</pre>	NOTE_D3	147	,			
<pre>#define</pre>	NOTE_DS3	156	5			
<pre>#define</pre>	NOTE_E3	165	5			

#define NOTE_F3 175 #define NOTE_FS3 185 #define NOTE_G3 196 #define NOTE_GS3 208 #define NOTE_A3 220 #define NOTE AS3 233 #define NOTE_B3 247 #define NOTE_C4 262 #define NOTE CS4 277 #define NOTE D4 294 #define NOTE DS4 311 #define NOTE E4 330 #define NOTE F4 349 #define NOTE_FS4 370 #define NOTE_G4 392 #define NOTE_GS4 415 #define NOTE_A4 440 #define NOTE_AS4 466 #define NOTE B4 494 #define NOTE_C5 523 #define NOTE_CS5 554 #define NOTE D5 587 #define NOTE DS5 622 #define NOTE_E5 659 #define NOTE_F5 698 #define NOTE_FS5 740 #define NOTE G5 784 #define NOTE_GS5 831 #define NOTE A5 880 #define NOTE AS5 932 #define NOTE B5 988 #define NOTE C6 1047 #define NOTE CS6 1109 #define NOTE D6 1175 #define NOTE DS6 1245 #define NOTE E6 1319 #define NOTE_F6 1397 #define NOTE_FS6 1480 #define NOTE G6 1568 #define NOTE_GS6 1661 #define NOTE_A6 1760 #define NOTE AS6 1865 #define NOTE B6 1976 #define NOTE_C7 2093 #define NOTE CS7 2217 #define NOTE D7 2349 #define NOTE DS7 2489 #define NOTE E7 2637 #define NOTE F7 2794

```
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE D8 4699
#define NOTE_DS8 4978
#define PULSADOR 2
#define BUZZER PASIVO 8
int melodia[] = {
 NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
};
int duraciones[] = {
 4,8,8,4,4,4,4,4
};
void setup() {
pinMode(PULSADOR, INPUT_PULLUP);
pinMode(BUZZER_PASIVO, OUTPUT);
}
void loop() {
  if(digitalRead(PULSADOR) == LOW){
    for (int i = 0; i < 8; i++){</pre>
      int duracion = 1000 / duraciones[i];
      tone(BUZZER_PASIVO, melodia[i], duracion);
      int pausa = duracion * 1.30;
      delay(pausa);
      noTone(BUZZER PASIVO);
    }
 }
}
```

Ahora vamos a cambiar la melodía.

```
int melodia[] = { // array con las notas de la melodia
NOTE_C4, NOTE_C4, NOTE_D4, NOTE_C4, NOTE_F4, NOTE_E4, NOTE_C4, NOTE_C4,
NOTE_D4, NOTE_C4, NOTE_G4, NOTE_F4, NOTE_C4, NOTE_C4, NOTE_C5, NOTE_A4,
NOTE_F4, NOTE_E4, NOTE_D4, NOTE_AS4, NOTE_AS4, NOTE_A4, NOTE_F4, NOTE_G4,
NOTE_F4
};
```

```
int duraciones[] = { // array con la duracion de cada nota
    8, 8, 4, 4, 4, 2, 8, 8, 4, 4, 2, 8, 8, 4, 4, 4, 4, 4, 4, 8, 8, 4, 4, 4,
2
};
```

El bucle for tiene que ser de 25.



Ya la podemos reproducir.

Según el siguiente circuito:



Queremos que en Potenciómetro que da un valor analógico desde 0 a 1023 lo convirtamos en valores desde 60 hasta 10.000 y según el movimiento de este nuestro Buzzer pasivo cambiará el todo del sonido.

```
1 int Pot = A0;
2 int Buz = 3;
3 int Potencia;
4 float Poten;
5 void setup()
6
   {
7
     pinMode(Buz, OUTPUT);
   serial.begin(9600);
8
9
   }
10
11 void loop()
12 {
13
    Potencia = analogRead(Pot);
14
    Poten = (Potencia/1023.*9940.) + 60.;
15
    digitalWrite(3, HIGH);
    delayMicroseconds (Poten);
16
17
    digitalWrite(3, LOW);
18
    delayMicroseconds(Poten);
19
   Serial.println(Poten);
20 }
```

Capítulo 28.- Paso a Paso (unipolar) 28BYJ-48 con Driver ULN2003







Vamos con el circuito:







Vamos a programar con el primer ejemplo (Paso completo simple):

```
int IN1 = 8;
int IN2 = 9;
int IN3 = 10;
int IN4 = 11;
int demora = 20; //Demora entre pasos, no debe ser menor a 10 ms.
void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
}
void loop() {
  for (int i = 0; i < 512; i++) //512*4 = 2048 pasos
  {
    digitalWrite(IN1, HIGH); // paso 1
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    delay(demora);
    digitalWrite(IN1, LOW); // paso 2
    digitalWrite(IN2, HIGH);
```

```
digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    delay(demora);
    digitalWrite(IN1, LOW); // paso 3
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    delay(demora);
    digitalWrite(IN1, LOW); // paso 4
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    delay(demora);
  }
 digitalWrite(IN1, LOW); // Detiene por 5 segundos.
  digitalWrite(IN2, LOW);
 digitalWrite(IN3, LOW);
 digitalWrite(IN4, LOW);
 delay(5000);
}
```

Ahora vamos a unificar el código utilizando una matriz.



Este será el código:

```
int IN1 = 8;
int IN2 = 9;
int IN3 = 10;
int IN4 = 11;
int demora = 20; //Demora entre pasos, no debe ser menor a 10 ms.
// Paso completo simple
int paso [4][4] = {
    {1, 0, 0, 0},
    {0, 1, 0, 0},
    {0, 0, 1, 0},
```

```
\{0, 0, 0, 1\}
};
void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
}
void loop() {
  for (int i = 0; i < 512; i++) //512*4 = 2048 pasos</pre>
  {
    for (int i = 0; i < 4; i++)</pre>
    {
      digitalWrite(IN1, paso[i][0]);
      digitalWrite(IN2, paso[i][1]);
      digitalWrite(IN3, paso[i][2]);
      digitalWrite(IN4, paso[i][3]);
      delay(demora);
    }
  }
  digitalWrite(IN1, LOW); // Detiene por 5 segundos.
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
  delay(5000);
}
```

Ahora vamos a utilizar Paso completo con dos bobinas:

Paso	А	В	С	D
1	1	1	0	0
2	0	1	1	0
3	0	Ø	1	1
4	1	0	0	1

Solo hemos cambiado los valores de la matriz:

```
int paso [4][4] = {
    {1, 1, 0, 0},
    {0, 1, 1, 0},
    {0, 0, 1, 1},
    {1, 0, 0, 1}
};
```

Ahora vamos a programar MEDIO PASO:

Paso	А	В	С	D
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

Este será el código:

```
int IN1 = 8;
int IN2 = 9;
int IN3 = 10;
int IN4 = 11;
int demora = 20; //Demora entre pasos, no debe ser menor a 10 ms.
// Paso completo simple
int paso [8][4] = {
 \{1, 0, 0, 0\},\
  \{1, 1, 0, 0\},\
 \{0, 1, 1, 0\},\
 \{0, 0, 1, 0\},\
 \{0, 0, 1, 0\},\
 \{0, 0, 1, 1\},\
 \{0, 0, 0, 1\},\
 \{1, 0, 0, 1\}
};
void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
}
void loop() {
  for (int i = 0; i < 512; i++) //512*8 = 4096 pasos
  {
    for (int i = 0; i < 8; i++)</pre>
    {
```

```
digitalWrite(IN1, paso[i][0]);
     digitalWrite(IN2, paso[i][1]);
     digitalWrite(IN3, paso[i][2]);
     digitalWrite(IN4, paso[i][3]);
     digitalWrite(IN1, paso[i][4]);
     digitalWrite(IN2, paso[i][5]);
     digitalWrite(IN3, paso[i][6]);
     digitalWrite(IN4, paso[i][7]);
     delay(demora);
   }
 }
 digitalWrite(IN1, LOW); // Detiene por 5 segundos.
 digitalWrite(IN2, LOW);
 digitalWrite(IN3, LOW);
 digitalWrite(IN4, LOW);
 delay(5000);
}
```



puntos

Ver video





Su conexión



Para ello tenemos que acceder a la siguiente página web para descargar las librerías.

https://github.com/avishorp/TM1637

https://github.com/PaulStoffregen/Time



Instalamos las librerías y empezamos a programar.

```
#include <TM1637Display.h>
 1
 2
     #define CLK 2
 3
     #define DIO 3
 4
 5
     TM1637Display display(CLK, DIO);
 6
 7
 8
     void setup() {
 9
     display.setBrightness(5);
10
     }
11
12
     void loop() {
       display.showNumberDec(millis()/1000);
13
14
     }
```

setBrightness() admite 8 niveles de brillo entre 0 y 7.

Vamos a realizar un segundo programa:

<TM1637Display.h>
showNumberDec(número, ceros, longitud, posición)
onumero: número a mostrar
ceros: muestra/oculta ceros precedentes
(true, false)
longitud: longitud del número a mostrar
(1 a 4)
posición: inicio de ubicación del número
a mostrar (0 a 3)



```
#include <TM1637Display.h>
 1
 2
 3
     #define CLK 2
     #define DIO 3
 4
 5
 6
     TM1637Display display(CLK, DIO);
 7
 8 ∨ void setup() {
9
     display.setBrightness(5);
10
     }
11
12 \vee void loop() {
13
       display.showNumberDec(1,false);
14
       delay(2000);
       display.showNumberDec(1,true);
15
       delay(2000);
16
       display.clear();
17
18
19 \sim
       for(int i=0; i<4; i++){</pre>
20
         display.showNumberDec(i, false, 1, i);
21
         delay(2000);
         display.clear();
22
23
        3
24
     }
```

Ya lo puedes subir para ver el resultado.

Vamos a realizar un programa que simulará un reloj.

```
#include <TimeLib.h>
 1
 2
     #include <TM1637Display.h>
 3
 4
     #define CLK 2
 5
 6
     #define DIO 3
 7
 8
     TM1637Display display(CLK, DIO);
9
10
     void setup() {
      display.setBrightness(5);
11
12
     setTime(16, 34, 45, 9, 11, 2022);
13
14
15
     void loop() {
       time_t t = now();
16
17
       display.showNumberDecEx(hour(t), 0b01000000, false, 2, 0);
18
       display.showNumberDec(minute(t), false, 2, 2);
19
       delay(100);
20
```

Tú en el recuadro pondrás la hora, minutos, segundos, día, mes y año actual.

Cuando subas el programa, ya tenemos un reloj.





Ahora modifica el código de void loop():

```
void loop() {
  time_t t = now();
  display.showNumberDecEx(hour(t), 0b01000000, false, 2, 0);
  display.showNumberDec(minute(t), false, 2, 2);
  delay(100);
  display.showNumberDecEx(hour(t), 0b00000000, false, 2, 0);
  display.showNumberDec(minute(t), false, 2, 2);
  delay(100);
}
```

Ahora podrás observar que los dos puntos parpadean.

Ahora vamos a realizar otro programa:







```
#include <TM1637Display.h>
 1
 2
     #define CLK 2
 3
 4
     #define DIO 3
 5
     TM1637Display display(CLK, DIO);
 6
 7
     const uint8_t celsius[] = {
 8
9
      SEG_A | SEG_B | SEG_G | SEG_F, // cero volado
10
      SEG_A | SEG_F | SEG_E | SEG_D // C
11
     };
12
13
     void setup() {
14
     display.setBrightness(5);
15
     }
16
     void loop() {
17
       display.showNumberDec(21, false, 2, 0);
18
19
       display.setSegments(celsius, 2, 2);
20
     3
```





Vamos a programar.

```
Filtro pasa bajos

factor * valorAnterior + (1 - factor) * valorLeido

0,75 * valorAnterior + 0,25 * valorLeido

valorFiltrado
\begin{cases}
75 & del valor Anterior \\
+ \\
25 & del valor Leído
\end{cases}
```

```
1
    float factor = 0.75;
 2
     float maximo = 0.0;
    int minimoEntreLatidos = 300;
 3
 4
   float valorAnterior = 500;
 5
    int latidos = 0;
 6
 7
    void setup() {
 8
     pinMode(13, OUTPUT);
 9
      Serial.begin(9600);
10
     Serial.println("Inicializando mediciones");
11
     B
12
13
     void loop() {
14
      static unsigned long tiempoLPM = millis();
15
       static unsigned long entreLatidos = millis();
16
17
      int valorLeido = analogRead(A0);
       float valorFiltrado = factor * valorAnterior + (1 - factor) * valorLeido;
18
19
      float cambio = valorFiltrado - valorAnterior;
       valorAnterior = valorFiltrado;
20
21
22
      if((cambio >= maximo) && (millis() > entreLatidos + minimoEntreLatidos)){
23
       maximo = cambio;
        digitalWrite(13, HIGH);
24
25
        entreLatidos = millis();
26
       latidos++;
27
       }
28
      else{
       digitalWrite(13, LOW);
29
30
       3
```

```
maximo = maximo * 0.97;
31
       if(millis() >= tiempoLPM + 15000){
32
         Serial.print("Latidos por minuto: ");
33
         Serial.println(latidos * 4);
34
35
        latidos = 0;
         tiempoLPM = millis();
36
37
       }
38
       delay(50);
39
```

Este será el resultado:

Salida Monitor Serie × Message (Enter to send message to 'Arduino Uno' on 'COM3') Inicializando mediciones Latidos por minuto: 64 Latidos por minuto: 92 Latidos por minuto: 92

Capítulo 44.- Fuente de protoboard Arduino y electrónica MB102















Con el tester digital podrás averiguar que llegan los voltios que tú necesitas.



También podemos conectar un Arduino uno para encenderlo.



Este es su esquema.



Capítulo 45.- Niveles lógicos y Niveles de señal 5 – 3,3 V






En este ejemplo donde colocamos una resistencia entre el pulsador y GND, lo hacemos para asegurar que mientras no esté presionado el pin digital establecido como entrada, asegura un nivel lógico bajo y cambiará a alto al presionar dicho interruptor.

En definitiva existe un rango de tensión que el pin digital establecido como entrada lo hereda como HIGH o LOW.







Usado en las placas UNO, PRO y NANO.







Capítulo 46.- LM2596 Conversor DC-DC reductor de Salida









Conversor DC-DC LM2596



Reductor (step-down o buck)

- Voltaje de entrada: 4,5 a 40V
 ↓
 Voltaje de salida: 1,23 a 37V
- Voltaje de salida: 1,23 a 37V (ajustable)
- Suministro de corriente: 3A (máx.)









Funciona adecuadamente





Capítulo 47.-MPU6050 acelerómetro y giróscopo Teoría y Practica











MPU6050 Acelerómetro y Giróscopo











Vamos al circuito:

ADO GND (sin conexión) VCC	Mega 5V GND 21 20 2	Arduino 5V GND A5 A4 2	MPU6050 Ardu VCC 5V GND GNI SCL A5 SDA A4 INT 2	M
AD0 Dir GND (sin conexión) VCC		<i>₽</i>		
	ección I2C 0x68 0x69	Dire ión)	0 ID (sin conexión) C	AD0 GN D VCC

Para la programación tenemos que descargar un par de librerías.

Vamos a acceder al siguiente enlace: https://github.com/jrowberg/i2cdevlib

 (BBS) Arduino desde cero en Esp × Ω → C ili github.com/prowberg/i Aplicaciones 	GitHub - jrowberg/12cdevlib: 12: × +				e	* 1	-	0
1. Second second	Product ~ Solutions ~ Op	en Source 🗸 Pricing	Search	/ Sign in Sign up				
jrowberg / i2cdevlib Public				C. Notifications	¥ Fork (7.4k)	∯r s	itar (3.5k	
.> Code ③ Issues 240 11 P	ull requests 😡 Discussions 💽 Actions	🗄 Projects 🖽 Wiki 🔘 Security 🗠 Insights						
	P master - P1 branch 🛇 0	tags	Go to file Code -	About				
	jrowberg Remove extra begin/en	dTransmission calls during reads 96x3x4e 1	4 days ago 🕥 690 commits	I2C device library collection for AVR/Arduino or other C++-based MCUs				
	Arduino	Remove extra begin/endTransmission calls during reads	14 days ago	@ www.i2cdevlib.com				
	BeagleBoneBlack/I2Cdev	Fixed remaining I2C_PATH	4 years ago	C Readme				
	EFM32/I2Cdev	Add EFM32 platform	8 years ago	☆ 3.5k stars				
	ESP32_ESP-IDF	Delete ESP32_ESP-IDF/.vscode directory	4 months ago	♥ 7.4k forks				
	Jennic Jennic	Added Jennic platform and MPU6050 device implementation	7 years ago					
	MSP430	Remove extra begin/endTransmission calls during reads	14 days ago	Releases				
	PIC18	Changed readme to markdown.	8 years ago	No releases published				
	E RP2040	Merge pull request #706 from mjcross/patch-2	3 months ago					
	RaspberryPi_bcm2835	remove unused Arduino functions for RPi	2 years ago	Packages				
	STM32	Rename STM32MDK back to original STM32	13 months ago	No packages published				
	STM32HAL	Rename new STM32 to STM32HAL	13 months ago					
	dsPIC30F	coding style correct	6 years ago	Contributors 90				
	nRF51/I2CDev	Fix bug in writeBytes method for nRF51	5 years ago	020 11 100				
	gitignore	calibration added	4 months ago	👘 🚱 🕲 🗵				
	README.md	Update main README documentation	13 months ago	+ 79 contributors				
	E BEADAE							
	:= README.md			- Francisco - Fran				



Del botón Code seleccionaremos Download ZIP.



Vamos a descomprimir el archivo.

(2kroleville)	master
Image: Archivo Inicio Compartir Vista Image: Archivo Inicio Compartir Vista Image: Archivo Image: Archivo Image: Archivo Image: Archivo Anclar al Acceso rápido Copiar Pegar Image: Archivo Image: Archivo Image: Archivo Image: Archivo Image: Archivo Image: Archivo Archivo Image: Archivo Image: Archivo Image: Archivo Image: Archivo Archivo Pegar Image: Archivo Image: Archivo Image: Archivo Image: Archivo Archivo Pegar Image: Archivo Image: Archivo Image: Archivo Image: Archivo Archivo Archivo Image: Archivo Image: Archivo Image: Archivo Image: Archivo Image: Archivo Archivo Archivo Image: Archivo Image: Archivo Image: Archivo Image: Archivo Image: Archivo<	ceso Acto Mover Copiar a * a * Copiar Organizar
\leftrightarrow \rightarrow \checkmark \uparrow \square \Rightarrow i2cdevlib-master \Rightarrow Ard	Nombre
 ☆ Acceso rápido Escritorio Descargas 	HMC5883L
 Documentos Imágenes Arduino 	I2Cdev IAQ2000 ITG3200
Arduino	L3G4200D

Dentro de la carpeta Arduino copiaremos la carpeta I2Cdev.

Ahora iremos al directorio de las librerías de Arduino.

Esta se encuentra en Documentos/Arduino/Libreries.

Una segunda librería se encuentra en el siguiente enlace:

https://github.com/ZHomeSlice/Simple_MPU6050

💶 (385) Arduino desde cero en Esp: 🗙 🎧 GitHub - Z							v -	0	×			
\leftarrow \rightarrow C i github.com/ZHomeSlice/Simple	_MPU6050							ß	☆	* 5	n 🗆 🤻	b E
Aplicaciones												
(Product ~ Solutions ~ Open S	ource v Pricing		Search		7 Sign in	Sign up					
ZHomeSlice / Simple_MPU6050	Public							¥ Fork 21		슈 Star	56 -	
<> Code Issues Pull requests	💿 Actions 🗄 Projects 🛈 Securit	Insights										
	P master - P 1 branch 🗞 0 tags		Go to file	Code -	About							
	2 ZHomeSlice Made Changes to allow fi	r updated Simple_Wire Compatibility	4ac5f5e on 20 Sep 🔞 85 o	commits	Going Live							
	Examples	Made Changes to allow for updated Simple_Wire Comp	patibility last	t month	δ₫δ MIT license							
				ears ago	56 stars							
	.gitignore	Initial commit	3 уе	ears ago	V 21 forks							

Repetimos para descargar el archivo.



En este caso no hace falta descomprimirlo, lo instalaremos desde Arduino.

Del menú Sketch Seleccionamos incluir biblioteca y de este Añadir biblioteca .ZIP

sketch_oct28a | Arduino IDE 2.0.1 Archivo Editar Sketch Herramientas Ayuda #include "Simple_Wire.h" #include "Simple_MPU6050.h" // incluye libreria Simple MPU6050 #define MPU6050 ADDRESS AD0 LOW // direccion I2C con AD0 en 0x68 LOW o sin conexion #define MPU6050_ADDRESS_AD0_HIGH 0x69 // direccion I2C con AD0 en HIGH #define MPU6050 DEFAULT ADDRESS MPU6050 ADDRESS AD0 LOW // por defecto AD0 en LOW Simple_MPU6050 mpu; // crea objeto con nombre mpu // ENABLE_MPU_OVERFLOW_PROTECTION(); // activa proteccion, ya no se requiere // #define OFFSETS -5114, 484, 1030, 46, 14, 6 // Colocar valores personalizados #define spamtimer(t) for (static uint32_t SpamTimer; (uint32_t)(millis() - SpamTimer) >= (t); SpamTimer = millis()) // spamtimer funcion para generar demora al escribir en monitor serie sin usar delay() #define printfloatx(Name,Variable,Spaces,Precision,EndTxt) print(Name); {char S[(Spaces + Precision + 3)];Serial.print(F(" ")); Serial.print(dtostrf((float)Variable,Spaces,Precision ,S));}Serial.print(EndTxt); // printfloatx funcion para mostrar en monitor serie datos para evitar el uso se multiples print() // mostrar valores funcion que es llamada cada vez que hay datos disponibles desde el sensor void mostrar_valores (int16_t *gyro, int16_t *accel, int32_t *quat, uint32 t *timestamp) { uint8_t SpamDelay = 100; // demora para escribir en monitor serie de 100 mseg

```
Quaternion q; // variable necesaria para calculos posteriores
 VectorFloat gravity; // variable necesaria para calculos
posteriores
 float ypr[3] = { 0, 0, 0 }; // array para almacenar valores de yaw,
pitch, roll
  float xyz[3] = { 0, 0, 0 }; // array para almacenar valores
convertidos a grados de yaw, pitch, roll
  spamtimer(SpamDelay) { // si han transcurrido al menos 100 mseg
entonces proceder
   mpu.GetQuaternion(&q, quat); // funcion para obtener valor para
calculo posterior
   mpu.GetGravity(&gravity, &q); // function para obtener valor para
calculo posterior
   mpu.GetYawPitchRoll(ypr, &q, &gravity); // funcion obtiene valores de
yaw, ptich, roll
   mpu.ConvertToDegrees(ypr, xyz); // funcion convierte a grados
sexagesimales
   Serial.printfloatx(F("Yaw") , xyz[0], 9, 4, F(", ")); // muestra
en monitor serie rotacion de eje Z, yaw
   Serial.printfloatx(F("Pitch"), xyz[1], 9, 4, F(", ")); // muestra
en monitor serie rotacion de eje Y, pitch
   Serial.printfloatx(F("Roll"), xyz[2], 9, 4, F(", ")); // muestra
en monitor serie rotacion de eje X, roll
   Serial.println(); // salto de linea
  }
}
void setup() {
  uint8 t val;
#if I2CDEV IMPLEMENTATION == I2CDEV ARDUINO WIRE // activacion de bus
I2C a 400 Khz
 Wire.begin();
 Wire.setClock(400000);
#elif I2CDEV IMPLEMENTATION == I2CDEV BUILTIN FASTWIRE
  Fastwire::setup(400, true);
#endif
 Serial.begin(115200); // inicializacion de monitor serie a 115200
bps
 while (!Serial); // espera a enumeracion en caso de modelos con
USB nativo
 Serial.println(F("Inicio:")); // muestra texto estatico
#ifdef OFFSETS
                            // si existen OFFSETS
  Serial.println(F("Usando Offsets predefinidos")); // texto estatico
 mpu.SetAddress(MPU6050 ADDRESS AD0 LOW).load DMP Image(OFFSETS); //
inicializacion de sensor
```

```
#else
```

// sin no existen OFFSETS

```
Serial.println(F(" No se establecieron Offsets, haremos unos nuevos.\n"
// muestra texto estatico
                   " Colocar el sensor en un superficie plana y esperar
unos segundos\n"
                  " Colocar los nuevos Offsets en #define OFFSETS\n"
                  " para saltar la calibracion inicial \n"
                  " \t\tPresionar cualquier tecla y ENTER"));
 while (Serial.available() && Serial.read()); // lectura de monitor
serie
 while (!Serial.available()); // si no hay espera
 while (Serial.available() && Serial.read()); // lecyura de monitor
serie
 mpu.SetAddress(MPU6050 ADDRESS AD0 LOW).CalibrateMPU().load DMP Image()
; // inicializacion de sensor
#endif
 mpu.on_FIFO(mostrar_valores); // llamado a funcion mostrar_valores si
memoria FIFO tiene valores
}
void loop() {
 mpu.dmp_read_fifo(); // funcion que evalua si existen datos nuevos
en el sensor y llama
```

} // a funcion mostrar_valores si es el caso

Este será el resultado:

Salida	Monitor Se	erie ×				* 6
Mess	age (Enter to s	send messa	ge to 'Arduino	Uno' on 'C	M3')	Ambos NL & CR 👻 115200 baud
Yaw	0.9639,	Pitch	4.4999,	Roll	2.0813,	
Yaw	0.9642,	Pitch	4.5000,	Roll	2.0743,	
Yaw	0.9642,	Pitch	4.5000,	Roll	2.0743,	
Yaw	0.9642,	Pitch	4.5000,	Roll	2.0743,	
Yaw	0.9642,	Pitch	4.5000,	Roll	2.0743,	
Yaw	0.9643,	Pitch	4.4930,	Roll	2.0742,	
Yaw	0.9642,	Pitch	4.5000,	Roll	2.0743,	
Yaw	0.9643,	Pitch				

La comunicación tiene que ser 115200 baud.

Ahora vamos a ver el segundo programa:

```
#include "Simple_Wire.h"
#include "Simple_MPU6050.h" // incluye libreria Simple_MPU6050
#define MPU6050_ADDRESS_AD0_LOW 0x68 // direccion I2C con AD0 en
LOW o sin conexion
#define MPU6050_ADDRESS_AD0_HIGH 0x69 // direccion I2C con AD0 en
HIGH
#define MPU6050_DEFAULT_ADDRESS MPU6050_ADDRESS_AD0_LOW // por
defecto AD0 en LOW
Simple_MPU6050 mpu; // crea objeto con nombre mpu
```

```
// ENABLE_MPU_OVERFLOW_PROTECTION(); // activa proteccion, ya no se
requiere
// #define OFFSETS -5114, 484,
                                    1030,
                                               46,
14,
      6 // Colocar valores personalizados
#define spamtimer(t) for (static uint32_t SpamTimer; (uint32_t)(millis())
- SpamTimer) >= (t); SpamTimer = millis())
// spamtimer funcion para generar demora al escribir en monitor serie sin
usar delay()
#define printfloatx(Name,Variable,Spaces,Precision,EndTxt) print(Name);
{char S[(Spaces + Precision + 3)];Serial.print(F(" "));
Serial.print(dtostrf((float)Variable,Spaces,Precision
,S));}Serial.print(EndTxt);
// printfloatx funcion para mostrar en monitor serie datos para evitar el
uso se multiples print()
// mostrar_valores funcion que es llamada cada vez que hay datos
disponibles desde el sensor
void mostrar_valores (int16 t *gyro, int16 t *accel, int32 t *quat,
uint32 t *timestamp) {
  uint8 t SpamDelay = 100; // demora para escribir en monitor serie
de 100 mseg
  Quaternion q; // variable necesaria para calculos posteriores
 VectorFloat gravity;
                        // variable necesaria para calculos
posteriores
  float ypr[3] = { 0, 0, 0 }; // array para almacenar valores de yaw,
pitch, roll
 float xyz[3] = { 0, 0, 0 }; // array para almacenar valores
convertidos a grados de yaw, pitch, roll
  spamtimer(SpamDelay) { // si han transcurrido al menos 100 mseg
entonces proceder
   mpu.GetQuaternion(&q, quat); // funcion para obtener valor para
calculo posterior
   mpu.GetGravity(&gravity, &q); // funcion para obtener valor para
calculo posterior
   mpu.GetYawPitchRoll(ypr, &q, &gravity); // funcion obtiene valores de
yaw, ptich, roll
   mpu.ConvertToDegrees(ypr, xyz); // funcion convierte a grados
sexagesimales
   Serial.printfloatx(F("Yaw") , xyz[0], 9, 4, F(", ")); // muestra
en monitor serie rotacion de eje Z, yaw
   Serial.printfloatx(F("Pitch"), xyz[1], 9, 4, F(", ")); // muestra
en monitor serie rotacion de eje Y, pitch
   Serial.printfloatx(F("Roll"), xyz[2], 9, 4, F(", ")); // muestra
en monitor serie rotacion de eje X, roll
   Serial.println();
                       // salto de linea
   if ( xyz[1] > 10.0 ){ // si el pitch es mayor a 10 grados
```

```
Serial.println("Nariz arriba"); // muestra texto
   }
     else if ( xyz[1] < -10.0){ // si el pitch es menor a -10 grados</pre>
       Serial.println("Nariz abajo"); // muestra texto
      }
 }
}
void setup() {
  uint8 t val;
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE // activacion de bus
I2C a 400 Khz
 Wire.begin();
 Wire.setClock(400000);
#elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
  Fastwire::setup(400, true);
#endif
 Serial.begin(115200); // inicializacion de monitor serie a 115200
bps
 while (!Serial); // espera a enumeracion en caso de modelos con
USB nativo
  Serial.println(F("Inicio:")); // muestra texto estatico
#ifdef OFFSETS
                            // si existen OFFSETS
 Serial.println(F("Usando Offsets predefinidos"));
                                                     // texto estatico
 mpu.SetAddress(MPU6050_ADDRESS_AD0_LOW).load_DMP_Image(OFFSETS); //
inicializacion de sensor
#else
                       // sin no existen OFFSETS
 Serial.println(F(" No se establecieron Offsets, haremos unos nuevos.\n"
// muestra texto estatico
                  " Colocar el sensor en un superficie plana y esperar
unos segundos\n"
                  " Colocar los nuevos Offsets en #define OFFSETS\n"
                  " para saltar la calibracion inicial \n"
                  " \t\tPresionar cualquier tecla y ENTER"));
 while (Serial.available() && Serial.read()); // lectura de monitor
serie
 while (!Serial.available());
                                    // si no hay espera
 while (Serial.available() && Serial.read()); // lecyura de monitor
serie
 mpu.SetAddress(MPU6050 ADDRESS AD0 LOW).CalibrateMPU().load DMP Image()
; // inicializacion de sensor
#endif
 mpu.on FIFO(mostrar valores); // llamado a funcion mostrar valores si
memoria FIFO tiene valores
}
```

```
void loop() {
```

mpu.dmp_read_fifo(); // funcion que evalua si existen datos nuevos
en el sensor y llama
} // a funcion mostrar_valores si es el caso

Este será el resultado:

da	Monitor Se	rie 🗙						
ssag	ge (Enter to s	end messa	ge to 'Arduino	Uno' on 'O	COM3')	Α	mbos NL & CR 🔻	
a.	0.0280,	Pitch	0.0140,	Roll	-0.0140,			
w	0.0280,	Pitch	0.0070,	Roll	-0.0210,			
a	0.0280,	Pitch	0.0140,	Roll	-0.0280,			
7	0.0280,	Pitch	0.0140,	Roll	-0.0280,			
1	0.0280,	Pitch	0.0140,	Roll	-0.0280,			
1	0.0280,	Pitch	0.0140,	Roll	-0.0210,			
W	0.0280,	Pitch	0.0140,	Roll	-0.0210,			
W	0.0280,	Pitch	0.0140,	Roll	-0.0140,			



Capítulo 48.- TTP223 Sensor Táctil Capacitivo Digital Touch sensor





Vamos a montar el circuito:

		TTP223B Arduino VCC 5V GND GND SIG 4
00		LED Arduino Ánodo (mediante 3 resistencia 330 Ω) Cátodo GND
		13
		₽

Vamos al primer programa.

```
#define LED 3
 1
 2
     #define TTP223B 4
 3
     void setup() {
 4
     pinMode(LED, OUTPUT);
 5
       pinMode(TTP223B, INPUT);
 6
 7
     }
 8
9
     void loop() {
       if (digitalRead(TTP223B) == HIGH){
10
        digitalWrite(LED, HIGH);
11
12
       }else{
13
         digitalWrite(LED, LOW);
14
       }
```

15	delay(50);	
16	}	

Vamos a modificar el circuito anterior.



Vamos a modificar el programa.

1	#define LED 3
2	#define TTP223B_MAS_4
3	#define TTP223B_MENOS 5
4	int BRILLO = 0;
5	
6	<pre>void setup() {</pre>
7	<pre>pinMode(LED, OUTPUT);</pre>
8	<pre>pinMode(TTP223B_MAS, INPUT);</pre>
9	<pre>pinMode(TTP223B_MENOS, INPUT);</pre>
10	}
11	
12	<pre>void loop() {</pre>
13	<pre>if(digitalRead(TTP223B_MAS) == HIGH){</pre>
14	BRILLO = BRILLO + 20;
15	if (BRILLO >= 255){
16	BRILLO = 255;
17	}
18	analogWrite(LED, BRILLO);
19	delay(100);
20	}else{

```
21
          if(digitalRead(TTP223B_MENOS) == HIGH){
         BRILLO = BRILLO - 20;
22
         if (BRILLO <= 0){
23
24
           BRILLO = 0;
25
          }
26
         analogWrite(LED, BRILLO);
27
         delay(100);
28
        3
29
       }
30
```

Ya podemos subir el programa, un sensor aumenta el brillo del led y el otro lo disminuye.



Otro modelo: Solo da señal por una sola cara.



Capítulo 49.- 74Hc595 Registro de desplazamiento (Shift register)









Este circuito también denominado registro de desplazamiento.



A los 8 pulsos de reloj.



Para que la carga no se realice hasta que se hayan introducido todos los valores.



Por mediación de un registro de almacenamiento.



Logic Diagram (Positive Logic)



OE: Output Enable, que limita las salidas.

SRCLR: Shift Register Clear, que resetea al registro de desplazamiento.

+ A +	5V 🔶 🦳		+	$R = 330\Omega$	LEDO
10 uF	vcc	(16)	(15) QA		
16V	SRC	LR (10)	(1) QB		
D7	O SER	(14)	(2) QC		
+ D8	O RCL	К (12)	(3) QD		
D9	o SRC	LK (11)	(4) QE		
			(5) QF		
	T DE	(13)	(6) QG		
	GND	(8)	(7) QH		XÎ
			(9) QH'	—X	1000
					4

Este es el circuito:



Vamos a escribir el código:

```
int pinDato = 9; // Pin conectado a DS pin 14 de 74HC595
int pinClock = 11; // Pin conectado a SHCP pin 11 de 74HC595
int pinRegistro = 12; // Pin conectado a STCP pin 12 de 74HC595
void setup() {
  // configura los pines a la salida porque se usan en el bucle principal
  pinMode (pinDato, OUTPUT);
 pinMode (pinClock, OUTPUT);
 pinMode (pinRegistro, OUTPUT);
}
void loop() {
 // rutina que cuenta de 0 a 255
 for (int j = 0; j <256; j++) {</pre>
   // pone a BAJO pinRegistro y lo mantiene bajo durante el tiempo que
este transmitiendo
    digitalWrite(pinRegistro, LOW);
    // envia el dato
   shiftOut(pinDato, pinClock, MSBFIRST, j);
    // retorna el pinRegistro ALTO para sacar los datos hacia las salidas
    digitalWrite(pinRegistro, HIGH);
    delay(200); // breve espera para poder ver el conteo
  }
}
```

Ya podemos subir el programa y podrás ver el resultado.

Vamos a realizar otro programa:

	LED	7	6	5	4	3	2	1	0
		27	26	25	24	2 ³	2 ²	21	20
er	n Decimal	128	64	32	16	8	4	2	1

🔤 sket	tch_oct27a	Arduino IDE 2.0.1
Archivo	Editar S	Sketch Herramientas Ayuda
Ø	€	🖞 Arduino Uno 👻
	sketch_c	oct27a.ino
	1	#define SER 9
	2	#define RCLK 11
1	3	#define SRCLK 12
	4	
Ith	5	#define LED0 1
	6	#define LED1 2
	7	#define LED2 4
	8	#define LED3 8
~	9	#define LED4 16
	10	#define LED5 32
X	11	#define LED6 64
	12	#define LED/ 128
	13	upid cotup() (
	14	void Setup() {
	15	pinMode(SEK, OUTPUT);
	17	ninMode(SRCLK_OUTPUT);
	18	}
	19	,
	20	<pre>void loop() {</pre>
	21	digitalWrite(RCLK, LOW);
	22	<pre>shiftOut(SER, SRCLK, MSBFIRST, LED1);</pre>
	23	<pre>digitalWrite(RCLK, HIGH);</pre>
	24	delay(500);
	25	}

Ejecuta el programa y verás que se enciende el primer led, cambiar por LED2, y observarás que ahora se enciende el segundo led y así sucesivamente.

Si queremos encender varios leds.

```
void loop() {
    digitalWrite(RCLK, LOW);
    shiftOut(SER, SRCLK, MSBFIRST, LED0 + LED1);
    digitalWrite(RCLK, HIGH);
    delay(500);
}
```

Otra combinación:

```
void loop() {
    digitalWrite(RCLK, LOW);
    shiftOut(SER, SRCLK, MSBFIRST, LED0 + LED1 + LED7);
    digitalWrite(RCLK, HIGH);
    delay(500);
}
```

Para finalizar el capítulo que enseño como encadenar dos 74HC595.



Ahora podremos tener 16 leds.

En la programación realizaremos los siguientes cambios.

```
#define LED0 1
#define LED1 2
#define LED2 4
#define LED3 8
#define LED4 16
#define LED5 32
#define LED6 64
#define LED7 128
#define LED8 256
#define LED9 512
```

```
#define LED10 1024
#define LED11 2048
#define LED12 4096
#define LED13 8192
#define LED14 16384
#define LED15 32768

parte alta parte baja
4096 : 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
(en decimal)
shiftOut() permite solo 8 bits
>> Operador de desplazamiento de bits
parte alta parte baja
LED12 >> 8 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
void loop() {
    digitalWrite(RCLK, LOW);
    shiftOut(SER, SRCLK, MSBFIRST, (LED12 >> 8)); //parte alta
    shiftOut(SER, SRCLK, MSBFIRST, LED12); //parte baja
    digitalWrite(RCLK, HIGH);
    delay(500);
}
```

Capítulo 50.- Actualizar el IDE a la última versión estable (2022)





Para poder actualizar tenemos que cerrar la aplicación.

Nos vamos al siguiente enlace: <u>https://www.arduino.cc/</u>


Seleccionaremos la opción Software.

Software | Arduino ← → C ☆ @ arduino.cc/en/so 🖈 🐊 İncógnito STORE PROFESSIONAL Q Search on Arduino.cc SIGN IN 00 HARDWARE SOFTWARE CLOUD - DOCUMENTATION - COMMUNITY - BLOG ABOUT Arduino Web Editor Start coding online and save your sketches in the cloud. The m up-to-date version of the IDE includes all libraries and also supports new Arduino boards. CODE ONLINE GETTING STARTED Downloads NLOAD OPTIONS 00 Arduino IDE 1.8.19 dows Win 7 and newer dows ZIP file The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board. Wir dows app Win 8.1 or 10 Get 🔢 Linux 32 bits Linux 64 bits Linux ARM 32 bits Refer to the Getting Started page for Installation instructions. Linux ARM 64 bits

Se puede visualizar la última versión disponible y que es estable.



Si observamos un versión nueva y estable procederemos a su descarga e instalación.

En mi caso seleccionaré Windows, seguido de JUST DOWNLOAD.

Cuando vayamos a instalar el programa podremos observar el siguiente mensaje:



La versión 1.8.15 de Arduino ya está instalada y debe desinstalarse antes de instalar esta versión. Esto no afectará a los sketches o librerías en el directorio sketch.

Haga clic en Aceptar para desinstalar.

Haga clic en Cancelar si tiene en el directorio de instalación que desea conservar (la mayoría de los usuarios no lo hacen). Copie en otro lugar y luego vuelva a ejecutar el instalador.

El directorio por defecto de librerías está:

Documentos\Arduino\libreries.

Una vez seleccionemos aceptar.



Nos muestra el directorio donde se instalará la aplicación.

Presionamos Unnistall.



El software Arduino se ha desinstalado. Puede eliminar de forma segura el directorio de instalación Arduino (si aún existe) y el directorio de Arduino sketch si lo desea.

Ahora se inicializa la instalación.



Dejamos las opciones por defecto.

Arduino Setup: Installation Folder		-		×
Setup will install Arduino in the fol folder, click Browse and select an installation.	lowing folder. other folder. (To install in Click Install	a differen to start the	t
Destination Folder				
C:\Program Files (x86)\Arduino\	I		Browse	
Space required: 541.6MB Space available: 114.6GB				
Correl Miles Present France		< Back	Inst	d

Muestra el nuevo directorio donde instalará los archivos.

Seleccionaremos Install.

💿 Arduino Setup: I	nstalling	-	
Extract: spec	s-attiny 1606		
Show details			
Cancel N	ulisoft Install System v3.0	< Back	Close N

Capítulo 51.- IDE 2.0 Instalación y Completo análisis



Para realizar la instalación de esta versión vamos a ir al siguiente enlace:

https://www.arduino.cc/en/software

🚥 (1) Arduino desde cero en Españ 🗴 💿 Software Arduir	• x +		~ -	0
← → C a arduino.cc/en/software		6 ල් \$	* =	
PROFESSIONAL EDUCATION STORE		Q. Search on Arduino.cc		SIGN IN
∞	HARDWARE SOFTWARE CLOUD DOCUMENTATION + COMMUNITY + BLOG ABOUT			
	Arduino Web Editor Start dong onlive and sare into the the into the data. The exact subscience we store of the big induces all barries and allow subscripts free Advances. The big induces all barries and allow populations and allow and allow and allow and allow and allow coole ONLINE GETTING STARTED	w.		
	Arduino IDE 2.0.1 Mindows with the latest bugfixes are available through the section below. Nightly builds with the latest bugfixes are available through the			
	source code The Arduino IDE 2.0 is open source and its source code is hosted on GitHub.		() H	elp

Lo podrás instalar para Windows, Linyx y macOs. Nosotros vamos a seleccionar la versión para Windows.

Since t bee	Support the Arduino IDE he release 1.x release in March 2015, the Arduino IDE has n downloaded 66.552.524 times — impressive! Help its development with a donation.
\$3	\$5 \$10 \$25 \$50 Other
	JUST DOWNLOAD
	Learn more about donating to Arduino.

Aquí podrás aportar un donativo para el proyecto o bien seleccionar JUST DOWNLOAD, que es lo que vamos a seleccionar.



Una vez descargado procederemos a su instalación.

Si tienes instalada una versión anterior, podrás mantener las dos versiones.

Una vez instalado observaremos una acceso directo en nuestro escritorio.



A diferencias de la versión anterior el icono es redondo.



En la parte superior encontramos la barra de menú.

sketch_nov1a Arduino IDE 2.0.1	_		×
Archivo Editar Sketch Herramientas Ayuda			
Arduino Uno		√	.Q.

Las opciones de acceso rápido.

Podremos verificar el código del programa, subir el programa a la placa y la opción de depurar solo disponibles para algunas placas.

Podemos seleccionar el tipo de placa que tenemos conectada a nuestro ordenador.

En la parte derecha veremos al Plotter Serie útil para mostrar de forma gráfica valores mediante simples Series.print(), por ultimo el monitor Serie que ya conocemos al haber realizado capítulos para ver datos por pantalla.

sketch_nov1a Arduino IDE 2.0.1	-		×
Image: Section Hermitieness Aydus Image: Section Hermitieness Aydus <td></td> <td>\mathbf{v}</td> <td>۰©۰</td>		\mathbf{v}	۰ © ۰
<pre>sketch_nov1a ino void setup() { // put your setup code here, to run once:</pre>			
Lin. I, col. I UIT-3 #Arduino U	io ino cor	rectadoj	ų

Sobre la columna de la izquierda tenemos:

	Con esta carpeta tendremos programas. Muestra dos bot	acceso a la cones lo que	carpeta c e está gua	donde tenemos nuestros rdado en el ordenador o en la
	nube. Costor do placas:	2		
<u>t</u>)	Administrar placas actuales Gestor de bibliotecas:	o instalar pl	lacas nuev	/as.
th	Nos permite instalar librería sensores y dispositivos.	s que podei	mos neces	sitar en algunos proyectos, para
¢ €	Acceso a depurar, nos perm código se encuentra en ejec	ite inspeccio ución.	onar múlt	iples parámetros mientras el
Q	Con la lupa tendremos acces reemplazar texto.	so rápido al	contenido	o del código. También podremos

Ahora del menú Archivo vamos a seleccionar Preferencias...

Preferencias		
	Configuración Red	
Ruta del Sketchbook:		
c:\Users\pmver\OneDrive\Docum	nentos\Arduino	EXPLORAR
🗌 Ver los ficheros dentro de los	bocetos	
Tamaño de letra del editor:	14	
Escala de la interfaz:	✓ Automático 100 %	
Tema de color:	Light (Arduino)	
Lenguaje del editor:	español 🗸 (Recarga necesaria)	
Mostrar salida verbosa durante	🗌 Compliar 🗌 Carga	
alertas de compilación	None 🗸	
○ Verificar el código después de ✓ Autoguardado	cargarlo	
Sugerencias rápidas del edito	r	
URLs adicionales de gestor de pl	acas:	E Contraction and a contractio
		CANCELAR ACEPTAR

Podremos modificar el tamaño de letra.

Podemos cambiar el Tema de color.



Puedes seleccionar el que más te guste.

Ahora vamos a conectar una placa Arduino uno.

🔤 sket	ch_nov1a Arduino	IDE 2.0.1		-		×
Archivo Editar Sketch Herramientas Ayuda						
	Ard	uino Uno			\checkmark	·Q··
	sketch_nov ψ	Arduino Uno COM3				
1	2 3 4	Desconocido COM15	to run once:			

Ya nos reconoce que en el puerto COM3 tenemos conectada la placa de Arduino.

Lín. 1, col. 1	UTF-8	Arduino Uno on 1COM3	D	

En la parte inferior derecha también nos está diciendo que tenemos conectado un Arduino uno en el puerto COM3.

Del menú Archivo seleccionaremos Ejemplos y de este Blink.

🔤 Blin	nk Arduino I	DE 2.0.1 —		×
Archivo	Editar Sk	xetch Herramientas Ayuda		
Ø	€ 🕞	ប៉ំ Arduino Uno 👻	\checkmark	۰ Q ۰
Ph	Blink.ino			
	9	If you want to know what pin the on-board LED is connected to on your Arduino		
53	10	model, check the Technical Specs of your board at:		
	11	https://www.arduino.cc/en/Main/Products		
0.0	12			
	13	modified & May 2014		
	14	modified 2 See 2016		- 1
	16	by Arturo Guadalupi		
201	17	modified 8 Sep 2016		
\sim	18	by Colby Newman		
Q	19			
	20	This example code is in the public domain.		
	21			
	22	https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink		
	23	*/		
	24	// the setup function runs once when you press reset or nower the board		
	26	y the setup () {		
	27	// initialize digital pin LED BUILTIN as an output.		
	28	<pre>pinMode(LED_BUILTIN, OUTPUT);</pre>		
	29	}		
	30			
	31	// the loop function runs over and over again forever		
	32	dialoop() {		
	33	delay(1000). // wait for a second		
	35	digitalWrite(LED BUILTIN, LOW): // turn the LED off by making the voltage LOW		
	36	delay(1000); // wait for a second		
	37	}		
	38			
		Lín. 1, col. 1 UTF-8 🔮 Arduino Uno on 1	1COM3	4

Se nos abre otra ventana con el código.

Vamos a seleccionar verificar.

Salida	≡ 6
El Sketch usa 924 bytes (2%) del espacio de almacenamiento de programa. El máximo es 32256 byt	es.
Las variables Globales usan 9 bytes (0%) de la memoria dinámica, dejando 2039 bytes para las v	ariables locales. El m

En la parte inferior se nos abre otra ventana, si tenemos errores en el programa aquí se mostrarán.



Vamos a subir el programa a la placa.

Salida	≣ 6
El Sketch usa 924 bytes (2%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.	
Las variables Globales usan 9 bytes (0%) de la memoria dinámica, dejando 2039 bytes para las variables local	les.El m

El primer icono de la derecha borra los mensajes de salida.



En la parte inferior nos muestra dos notificaciones, si lo seleccionamos.



Muestra la Compilación completada y después la carga completada.



Este botón nos sirve para mostrar u ocultar el panel de salida.

21				
22		https://www.arduino.cc/en/Tutorial/Bui	ltInExamples/Blink	
23		*/		
24				
25		// the setup function runs once when you	press reset or power the board	
26	\sim	<pre>void setup() {</pre>		
27		<pre>// initialize digital pin LED_BUILTIN</pre>	as an output.	
28		<pre>pinMode(LED_BUILTIN, OUTPUT);</pre>		
29		}		
30				
31		// the loop function runs over and over again forever		
32	\sim	<pre>void loop() {</pre>		
33	_	<pre>digitalWrite(LED_BUILTIN, HIGH); // t</pre>	urn the LED on (HIGH is the voltage level)	
34		delay(1000); // w	ait for a second	
35		<pre>digitalWrite(LED_BUILTIN, LOW); // t</pre>	urn the LED off by making the voltage LOW	
36		delay(1000); // w	ait for a second	
37		}		

Con estos triángulos podemos expandir y contraer parte de nuestro código.

Ahora vamos a abrir del apartado ejemplos con programa AnalogReadSerial.

SeadSerial Arduino IDE 2.0.1 - 🗆 🗙							
Archivo	Editar S	ketch Herramientas Ayuda					
\bigcirc	€ 🗧	🖞 Arduino Uno 👻	·Q··				
Р	AnalogRe	eadSerial.ino					
	1	/*					
_	2	AnalogReadSerial					
1_)	3						
	4	Reads an analog input on pin 0, prints the result to the Serial Monitor.					
ITh	5	Graphical representation is available using Serial Plotter (Tools > Serial Plotter menu).					
011/2	6	Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and ground.					
	8	This example code is in the public domain					
*	9	This example code is in the public domain.					
	10	https://www.arduino.cc/en/Tutorial/BuiltInExamples/AnalogReadSerial					
Q	11	*/					
	12						
	13	<pre>// the setup routine runs once when you press reset:</pre>					
	14	<pre>void setup() {</pre>					
	15	<pre>// initialize serial communication at 9600 bits per second:</pre>					
	16	Serial.begin(9600);					
	1/	}					
	10	// the loop routine runs over and over again forever:					
	20	void loop() {					
	21	// read the input on analog pin 0:					
	22	<pre>int sensorValue = analogRead(A0);</pre>					
	23	// print out the value you read:					
	24	<pre>Serial.println(sensorValue);</pre>					
	25	<pre>delay(1); // delay in between reads for stability</pre>					
	26	}					
	27						
		Lín. 1, col. 1 UTF-8 📲 Arduino Uno on 1COM	Д				

Nos lo abre en una nueva ventana, ya podemos cerrar la anterior.

Este programa nos muestra por el monitor serial los valores obtenidos por la entrada analógica AO, con esta no tiene ningún sensor conectado, al ejecutar el programa y activar el monitor Serial se mostrarán valores aleatorios.



Se abre una nueva ventana al de salida.

Se podrá cerrar seleccionando la x.

Modifica la demora a 1 segundo, delay(1000);

Al cargar de nuevo el programa a la placa y activar el monitor serie, observarás que ahora se muestra un resultado cada segundo.

Si seleccionamos de nuevo del menú Archivo / Preferencias, podemos modificar la escala, de este modo cambiar el tamaño de letra tanto en el código como el monitor de salida.



En la parte superior de este proyecto tenemos un comentario que ocupar varias líneas, vamos a contraer dicha zona.



De este modo podemos ocultar parte del código y poder trabajar con más comodidad en otra parte del programa.

Si le damos de nuevo expandiremos este código, se volverá a ver toda esta parte.

Otro detalle al tener en cuenta es que al seleccionar parte del código se muestran unos puntos.



Estos punto que aparecen entre las palabras nos indicar que hay espacios en blanco, si tenemos que hacer una comparación con otro valor y no hay los mismos espacios en blanco nos dirá que dichas cadenas no son iguales.



Al situarnos sobre una palabra reservada del leguaje de programación nos devuelve una referencia rápida de lo que la función nos devuelve.

También nos dice los parámetro que tiene que llevar la función.

En esta versión tendremos ayuda sobre el código que estamos escribiendo.

Si tenemos una placa Arduino Nano 33 BLE podremos escribir el siguiente código:

	<pre>void setup() {</pre>
	<pre>pinMode(LEDR, OUTPUT);</pre>
	<pre>pinMode(LEDG, OUTPUT);</pre>
	<pre>pinMode(LEDB, OUTPUT);</pre>
	}
	<pre>void loop() {</pre>
	<pre>digitalWrite(LEDR, LOW);</pre>
	<pre>digitalWrite(LEDG, HIGH);</pre>
10	digitalWrite(LEDB, HIGH);
11	delay(1000);
12	digitalWrite(LEDR, HIGH);
13	<pre>digitalWrite(LEDG, LOW);</pre>
14	digitalWrite(LEDB, HIGH);
15	delay(1000);
16	digitalWrite(LEDR, HIGH);
17	digitalWrite(LEDG, HIGH;
18	digitalWrite(LEDB, LOW);
19	delay(1000);
20	}



Capítulo 52.- Comunicación entre dos placas de Arduino

Para este capítulos vamos a realizar el correspondiente circuito:



La placa de Arduino en el apartado de pines digitales encontramos el 0 y el 1 que nos permiten realizar la conexión entre placas de Arduino.

Los vamos a conectar de la siguiente forma: El pin 1 TX de una placa lo conectamos al 0 RX de la otra placa y de este modo el pin 0 TX de la primera placa lo conectamos al 1 TX de la segunda placa.

También hemos de conectar GND entre las dos placas.

La primera placa tiene dos botones en de la izquierda tiene que encender el led de la segunda placa y el de la derecha tiene que apagar el led de la segunda placa.

Este es el código para la placa que se encuentra a la izquierda:

```
1 int boton1 = 3;
 2 int boton2 = 2;
 3 int estado = 0;
 4
   void setup()
 5
   -{
    pinMode(boton1, INPUT);
pinMode(boton2, INPUT);
 6
 7
 8
      Serial.begin(9600);
9
   -}
10
11 void loop()
12 {
13
     if ( digitalRead (boton1) == HIGH) {
14
       estado = 1;
15
16
      -}
17
     if( digitalRead(boton2) == HIGH) {
      estado = 0;
18
19
        }
20
21
      if (estado == 1) {
22
        Serial.write('1');
23
24
      if(estado == 0){
25
        Serial.write('0');
26
      3
27 }
```

La segunda placa contiene un led con su respectiva resistencia que está conectada al pin 8, esta segunda placa tiene que recibir comunicación de la primera para encender y apagar el led.

Este será el código:

```
1 int estado = 0;
 2 int led = 8;
 3
 4 void setup()
5 {
 6
     Serial.begin(9600);
 7
     pinMode(led, OUTPUT);
   }
8
9
10 void loop()
11 {
12
     if(Serial.available()){
13
       estado = Serial.read();
14
     }
15
     if (estado == '1') {
16
          digitalWrite(led, HIGH);
17
     }else if(estado == '0'){
18
          digitalWrite(led, LOW);
19
     }
20 }
```

Capítulo 53.- i2c entre dos Arduino

Vamos a realizar el siguiente circuito:



Este será el código de la placa Maestro:

```
1 #include <Wire.h>
 2
 3 // Comunicación IC2 entre dos Arduinos.
 4 // MAESTRO
 5
 6
 7 void setup()
 8
    {
 9 // Unimos este dispositivo al bus I2C
 10
    Wire.begin();
 11 }
 12 byte pin[] = {2,3,4,5,6};
 13 byte estado = 0;
 14
 15 void loop()
 16 {
 17
     for (int i = 0; i < 5; i++)
 18
 19
     // Comenzamos la transmisión al dispositivo 1.
 20
     Wire.beginTransmission(1);
 21
 22
      // Enviamos un byte, será el pin a encender.
 23
      Wire.write(pin[i]);
 24
 25
      // Enmviamos un byte, L: Pondrá en estado bajo
 26
      // H: Pondrá en estado alto.
 27
      Wire.write(estado);
 28
 29
      // Paramos la transmisión.
 30
     Wire.endTransmission();
 31
 32
     // Esperamos 300 milisegundos.
     delay(300);
 34
      }
 35
 36 // Cambiamos de estado.
    if (estado == 0)
 37
 38
     {
 39
      estado = 1;
 40
     }
 41
     else
 42
     - {
 43
        estado = 0;
 44
     }
 45 }
```

Este será el código de la placa Esclavo:

```
1 // Comunicación I2C entre dos Arduinos
2 // ESCLAVO
3
4 #include <Wire.h>
5
6 void setup(){
7 // Pines en modo salida
8 pinMode(2, OUTPUT);
9 pinMode(3, OUTPUT);
10 pinMode(4, OUTPUT);
```

```
11 pinMode(5, OUTPUT);
12
      pinMode(6, OUTPUT);
 13
 14
       // Unimos este dispositivo al bus I2C con dirección 1.
 15
      Wire.begin(1);
 16
 17
      // Registramos el evento al recibir datos
 18
      Wire.onReceive(recxeiveEvent);
 19
 20
       // Iniciamos el monitor serie para monitorear la comunicación.
 21
      Serial.begin(9600);
 22
    }
 23
 24
    void loop() {
 25
     delay(300);
 26
 27
 28
    // Función que se ejecuta simpre que se reciben datos del maester.
 29
    // Siempre que en el master se ejecute la sentencia endTransmission.
    // Recibiraá toda la información que hayamnos pasado a través de la
 30
 31
 32 void recxeiveEvent(int howMany) {
 33
 34
      int pinOut = 0;
 35
      int estado = 0;
 36
 37
      // Si hay dos bytes disponibles
      if (Wire.available() == 2)
 38
 39
      - {
 40
      // Leemos el primero que será el pin.
 41
        pinOut = Wire.read();
 42
         Serial.print("LED ");
 43
         Serial.println(pinOut);
 44
 45
       // Si hay un byte disponible
 46
 47
      if (Wire.available() == 1)
 48
       estado = Wire.read();
 49
        Serial.print("Estado ");
 50
 51
         Serial.println(estado);
 52
       -}
 53
      // Activamos y desactivamos salida.
 54
      digitalWrite(pinOut, estado);
 55
      3
```

Ahora vamos a modificar el circuito:



Los leds se tienen que encender de derecha a izquierda al mismo. Tanto los de arriba como los de abajo tienen que ir a la par.

Tienes que realizar unas pequeñas modificaciones para incluir el pin número 7.



Capítulo 54.- Serial de datos entre 2 tarjetas Arduino en Tinkercad Vamos a diseñar el siguiente circuito:



Desde la página web: https://www.tinkercad.com/

Intenta realizar el correspondiente circuito, con sus componentes y conexiones.

Ahora vamos a programar la placa que se encuentra a la izquierda:

int Sensor = A0; //Asignamos el puerto analógico A0 para el sensor TMP36. int ValorSensor = 0; //Variable para el almacenamiento del valor del sensor. float Temperatura = 0; // Variable para almacenar el valor de la tempertura.

```
int Grados_Cent; //Variable para grados centígrados.
int Grados_Fahr; //Variable para grados Fahrenheit.
int Kelv; //Varriable para Kelvin.
int Rojo = 10; //Asignación del puerto digital 10 para el color rojo.
int Verde = 9; //Asignación del puerto digital 9 para el color verde.
int Amarillo = 8; //Asignación del puerto digital 8 para el color amarillo.
```

unsigned long Tiempo_Tx;

void setup()

{

```
Serial.begin(115200); // Inicializamos el puerto de comunicaciones a una velocidad
delay(1000); // de 115200 bps.
//Configuramos los puertos digitales 8, 9 y 10 como salidas.
pinMode(Rojo, OUTPUT);
pinMode(Verde, OUTPUT);
pinMode(Amarillo, OUTPUT);
// Inicializamos los pines 8, 9 y 10 como nivel lógico 0.
digitalWrite(Rojo, LOW);
digitalWrite(Verde, LOW);
digitalWrite(Amarillo, LOW);
```

```
Tiempo_Tx = millis(); //Leemos el tiempo de inicio del programa.
```

}

```
void loop()
```

{

```
ValorSensor = analogRead(Sensor);
Temperatura = ((ValorSensor*(5000/1024.0))-500)/10;
```

```
Grados_Cent = (int)Temperatura; //Grados Celsius o centígrados.
Grados_Fahr = (int)(Grados_Cent-1.8)+32; //Grados Fahrenheit
Kelv = (int)(Grados_Cent + 273.15); //Kelvin
```

```
//Transmisión de datos
```

```
if((millis() - Tiempo_Tx) > 500) //Transmision encada segundo 1 seg = 1000 miliseg.
{
```

Serial.print(Grados_Cent); //Transmisión de grados centígrados.

```
Serial.print("C");
                             //Enviamos la unidad.
       Serial.print(Grados_Fahr); //Transmisión en grados Fahrenhit.
                             //Enviamos unidad.
       Serial.print("F");
       Serial.print(Kelv);
                              //Transmisión en Kelvin.
        Serial.print("K");
                             //Enviamso la unidad.
        Tiempo_Tx = millis();
                                //Tiempo de actualización.
}
//Recepción de datos.
if(Serial.available()>0)
{
       int datos = Serial.read(); //Lectura de datos.
       if(datos=='a')
                          //Temperatura alta.
        {
               digitalWrite(Rojo, HIGH);//Enciende LED para tempertura alta.
               digitalWrite(Amarillo, LOW);
               digitalWrite(Verde, LOW);
        }
       if(datos=='b')
                          //Temperatura baja.
       {
               digitalWrite(Rojo, LOW);
                digitalWrite(Amarillo,HIGH);//Enciende LED para tempertura baja.
                digitalWrite(Verde, LOW);
        }
        if(datos=='n')
                          //Temperatura normal.
        {
               digitalWrite(Rojo, LOW);
               digitalWrite(Amarillo, LOW);
                digitalWrite(Verde, HIGH); //Enciende LED para tempertura normal.
       }
}
```

A continuación vamos a programar la placa que se encuentra en la parte derecha:

#include <LiquidCrystal.h> //Incluimos librería LCD //Inicializamos la libreria LCD con sus respectivos puertos de configuración. LiquidCrystal lcd(12,11,5,4,3,2); char datos[13]; //Variable array para capturar datos de llegada. int posicion = 0;//Varaible de posicionar datos de llegada.

}

int Grados Cent; //Varaible para grados centígrados. int Grados_Fahr; //Varaible para grados Fahrenkeit. int Kelv: //Variable para Kelvin.

ł

}

ł

```
unsigned long Tiempo_LCD; //Varaible de tiempo parta impresión de texto en LCD.
unsigned long Tiempo_Tx; //Varaible de tiempo para trasmisión de datos.
```

```
void setup()
                               //Inicializamos la pantalla LCD de 16x2.
       lcd.begin(16,2);
       Serial.begin(115200); //Inicializamos el puerto de comunicación a una
                               //velocidad de 115200 bps.
        delay(1000);
                               //Retardo
        Tiempo_LCD = millis(); //Leemos tiempo de inicio del programa
                               //para pantalla LCD.
        Tiempo_Tx = millis(); //Leemos tiempo de inicio de progrtama
                               //para Tx
void loop()
        boolean segmentacion; //Variable para segmentación de datos después de la
                               //recepción de datos.
                               //Recepción de datos
       while(Serial.available()>0)
       {
               //Mientras haya datos en el buffer ejecuta la función.
               delay(5); //Ponemos un pequeño retardo para mejorar la recepción de datos.
                datos[posicion] = Serial.read(); //Leemos un carácter del string "cadena"
                                               //de la "posición", luego lee el siguiente.
                posicion++;
                               //Aumentamos la variable en 1 para ir levendo los datos de
                               //llegada uno a uno.
               segmentacion = true; //Después de la recepción de datos, aprobamos (true)
                                     //para introducir la segmentación.
        }
        posicion = 0; //Encerramos el contador de posición para empezar de nuevo
                     //en la próxima recepción de datos.
       //Segmentación de datos.
        if(segmentacion == true)
        {
               int index = 0;
               resultado = strtok(datos, separador); //Separamos los datos correspondientes
                                                    // a C, F y K.
               while((resultado != NULL) && (index <3))</pre>
                {
                       valores[index++] = atoi(resultado); //Almacenamos los datos
                                                         //segmentados en el arrray.
```

```
//y los transformamos.
               resultado = strtok(NULL, separador); //Anulamos los resultados para
                                                    //una próxima.
               //segmentación.
        }
       segmentacion = false; //Después de la segmentación, lo desaprobamos (false)
      //hasta nuevos datos de recepción.
}
Grados_Cent = valores[0]; //Leemos los datos de grados Celsius en la posición 0.
Grados_Fahr = valores[1]; //Leemos los datos de grados Fahrenheit en la posición 1.
Kelv = valores[2];
                     //Leemos los datos Kelvin en la posición 2 del array.
//Impresión en la pantalla LCD
if((millis() - Tiempo_LCD)>500)
        Icd.clear(); //Limpia la pantalla antes de presentar los nuevos datos.
        lcd.setCursor(0,1); //Fijamos el cursor de la pantalla en columna 0 fila 1.
       lcd.print(Grados_Cent); //Imprime temperatura en grados Celsius.
       lcd.print("C"); //Imprimimos la unidad.
       Icd.setCursor(6,1); //Fijamos el cursor en la pantalla en columna 6 fila 1.
        lcd.print(Grados_Fahr); //Imprimimos temperatura en Fahrenheit.
        lcd.print("F"); //Imprimimos la unidad.
        lcd.setCursor(12,1); //Fijamos el cursor de la pantalla columna 12 fila 1.
       lcd.print(Kelv); //Imprimmos temperatura en Kelvin.
       lcd.print("K"); //Imprimimos unidad.
       Tiempo LCD = millis(); //Tiempo de actualización.
}
//Transferencia de datos
if((millis() - Tiempo_Tx) > 500)
       if((Grados Cent>=20) && (Grados Cent<=25))
        {
               lcd.setCursor(0,0); //Fijamos el cursor de la pantalla en columna 0, fila 0
               lcd.print("TEMPERATURA NORM"); //Imprime comentario.
               Serial.print("n"); //Enviamos el carácter n para temperaturas normales.
       }
       else if (Grados Cent<20)
       {
               lcd .setCursor(0,0); //Fijamos el cursor en columna 0, fila 0.
               lcd.print("TEMPERATURA BAJA"); //Imprime comentario.
               Serial.print("b"); //Enviamos el carácter b para temperatusas bajas.
       }
       else if (Grados Cent>25)
       {
               lcd .setCursor(0,0); //Fijamos el cursor en columna 0, fila 0
               lcd.print("TEMPERATURA ALTA"); //Imprime comentario.
               Serial.print("a"); //Enviamos el carácter a para temperatusas altas.
        }
```

{

{

```
311
```

Tiempo_Tx = millis(); }

Ahora comprueba que funciona correctamente. Ahora vamos a realizar la practica con los siguientes componentes:

Vamos a necesitar:

}

3 leds de color rojo, verde y amarillo.

2 tarjetas de Arduino NANO.

1 Breadboard.

1 sensor de temperatura.

1 potenciómetro.

5 resistencias de 220 a 330 Ω .

Cables para las conexiones.

Vamos a conectar 2 tarjetas de Arduino para la transmisión de datos.

Conecta los sensores y actuadores correspondientes.

Realiza la conexión de cables:

Conecta primero la tarjeta Arduino de la izquierda y sube su correspondiente programa.

Conecta ahora la tarjeta Arduino de la derecha y sube su correspondiente programa.

Los programas los puedes seleccionar y copiar para luego pegarlos en el Arduino ID, para no tenerlos que copiar de nuevo.