



Reloj despertador y luz ambiente













WARE ME VP!



- 2 ¿Qué hace este proyecto?
 - Gelecciona el color de los **leds**.
 - Diferentes **secuencias** de colores.
 - **Velocidad** de encendido de led.
 - Muestra la hora
 - **Programar** una alarma
 - Alarma visual o sonora

ITE ANIMAS A DARLE VIDA!







Hardware necesario para Wate ME VP!









- ARDUINO NANO
- Pantalla LCD + I2C
- Pila y reloj
- Mando y sensor IR
- Anillo leds RGB
- Buzzer
- Cables arduino









Creative Commons Autoras: Elvira Castillo Twitter: @layoel Instagram: eca_ingeniera Aprenderemos a:

□ ¿Que es l2C?

Conectar la pantalla.

Escribir mensajes.





Antes de empezar... ¿Qué tenso que nacer para carsar vila Libreira?









- Las librerías las vamos a tener que cargar TODOS los días en los ordenadores del campus.
- Pero primero, antes de cargarlas, tenemos que configurar el IDE....









Para ello, entramos en el IDE de arduino, y nos vamos a preferencias.









- Aquí nos saldrá una pestaña como la que sale aquí, vemos que sale "Operador".
- Esto significa que se está usando la carpeta de operador, pero nosotras queremos nuestra carpeta "engiXX".
- Para ello...

	Preferencias	
justes Red		
Localización de proyecto		
1 1 10 11		Explorar
/home/operador/Arduing	1	







Para ello...

e proyecto	Seleccione nueva localización de proyecto	
ador/Arduino		
	Nueva Carpeta Suprimir Archivo Cambiar Nombre de Archivo	in quit
lid.	/home/operador 🔻	equi
año de Fuente	Carpetas	
	1	ī I
£,	of the second se	
	Arduino/	
	Descalgas/	
i detallada mie	Escritorio/	
	Imágenes/	
del compilado	Música/	
imeros de líne	Selección: /home/operador	
ódigo despué:	Actuan	
ar actualizacior	Arduno	argu
sibility feature	Filtro:	
	Todos los Archivos	
_s Adicionales		
cias pueden se	😵 Cancelar 🛛 😔 Acep	tar
/.arduino15/pre	erences.txt	

/Arduino					
	<u>N</u> ueva Carpeta	Suprimir Archivo	Cambia <u>r</u> Nombre d	le Archivo	
		/home			requie
le Fuente	Carpe <u>t</u> as				
	1				
	./				
	operadory				
allada mie					
ompilado					
os de líne					
o desnuér	<u>S</u> elección: /home	2			
ualizacior	/home				arou
y feature	Filtro:				a.g.
-	Todos los Archiv	os			
Icionates					
ueden se			🛛 🖾 Cancelar	💙 Aceptar	







Para ello...

<u>N</u> ueva Carpeta	uprimir Archivo Cambi	a <u>r</u> Nombre de Archivo
	/home/engi40	•
Carpe <u>t</u> as		
1		
/ Arduino/		
Descargas/		
Escritorio/		
Imágenes/		
Música/		
<u>S</u> elección: /home/	engi40	
/home/engi40		
Filtro:		
Todos los Archivo	5	
	(C)	Canadan Acarb

	Preferencias	MM	
ijustes Red	Preferencias		× × × ×
ijustes Red Localización de provecto /home/engi40/Arduino	Preferencias		Ex lorar
ijustes Red Localización de provecto /home/engi40/Arduino Editor de Idioma:	Preferencias		Ex lorar requiere reiniciar Arduino)

Así debería de quedar







🥺 sketch_jun1	9a Arduino 1.8.19	– 🗆 🗙
Archivo Edital P	Programa Herramientas Ayuda	
	Verificar/Compilar Ctrl+R	.
	Subir Ctrl+U	
sketch_jun	Subir Usando Programador Ctrl+Mayús+U	
17 void	Exportar Binarios compilados Ctrl+Alt+S	^
1001	Mostrar Carpeta de Programa Ctrl+K	
10 5	Incluir Librería	Administrar Bibliotecas Ctrl+Mavús+I
19 10	Añadir fichero	
20≊ {		Añadir biblioteca .ZIP
21	<pre>for (uint8_t pwmNum = 0; p</pre>	Arduino bibliotecas
22	{	Bridge
23	// Ajustar PWM con ON e	Esplora
24	pwm.setPWM(pwmNum, 0, d	Ethernet
25	}	Firmata
26 }		Keyboard
27 1		LiquidCrystal
<		Mouse

Ya después de esta configuración,

podemos añadimos las **librerías** necesarias.

Repetir este paso para cada una de las librerías.





Conectando el hardware del Wake Me VP!



https://sereingeniera.ugr.es/index.php/2020/07/08/ampliaciones-del-proyecto-laser-tag/







¿Qué es y cómo funciona una pantalla LCD?

Una **pantalla LCD** es una pantalla de retroiluminación LED que permite mostrar dos filas de 16 caracteres con los que podemos escribir texto.

Realmente no vamos a aprender a conectar la pantalla LCD directamente, ya que tiene muchos pines y si la conectáramos nos quedaríamos sin pines en el Arduino para todo lo demás. Por eso vamos a utilizar un controlador I2C.



En esta pantalla mostraremos mensajes de nuestro despertador luminoso y sonoro.





J ¿Que es l2C y para qué sirve?

El controlador I2C nos permite controlar la pantalla utilizando solamente dos cables de control (además del cable de 5v y la toma a tierra GND) a través de un tipo de comunicación entre placas llamada I2C.

- Los dos cables utilizados son SCL, que se conecta al pin analógico A5, y SDA, que se conecta al pin analógico A4:
- SCL envía señales de reloj (clock), que se encargan de decidir quién habla en cada momento, para evitar conflictos y SDA envía los datos.





MONTAJE PANTALLA LCD

 SCL-> A5
 GND-> GND

 SDA-> A4
 VCC-> 5V







Programando el *Wake Me VP!*



<u> https://sereingeniera.ugr.es/index.php/2020/07/08/ampliaciones-del-proyecto-laser-tag/</u>



En esta pantalla **mostraremos el día y la hora y los mensajes** que haya de configuración de nuestro reloj despertador

Recuerda los pines son:

SCL-> A5 GND-> GND SDA-> A4 VCC-> 5V





1. Incluir en el IDE la librería LiquidCrystal_I2C.h

Hay **dos formas de incluir librerías** a la biblioteca de nuestro arduino.

- Si tenemos el archivo comprimido con extensión .zip
- Buscarlas en el **repositorio** de Arduino

- Para la pantalla tenemos el archivo comprimido con el nombre
 LiquidCrystal_I2C-1.1.2.zip
- La **descargamos** y la incluimos de la 1º forma que vemos a continuación.





1. Descarga la librería y después la importamos en el arduino:

Está en el enlace:

http://downloads.arduino.cc/libraries/github.com/marcoschwartz/LiquidCrystal I2C-1.1.2.zip



- Para poder utilizar la pantalla con I2C además de descargarnos la librería de Arduino llamada <LiquidCrystal_I2C.h>, también necesitamos cargar otra llamada
 <Wire.h>.
- Al trabajar con funciones de la librería descargada, a las que no les hemos puesto el nombre nosotras, tenemos que acostumbrarnos y entender lo que hacen.

NOTA: Las librerías son como funciones creadas por otras personas y que nosotras podemos aprovechar y utilizar siempre que lo necesitemos.





1. Cargar librerías en el arduino:

En arduino hacemos clic en **Programa**-> incluir libreria-> añadir biblioteca Zip.

🥺 sketch_jun23a	a Arduino 1.8.19			
Archivo Editar Pr	ograma Herramientas Ayuda			
sketch_jun	Verificar/Compilar Subir Subir Usando Programador Exportar Binarios compilados	Ctrl+R Ctrl+U Ctrl+Mayús+U Ctrl+Alt+S		
2 //	Mostrar Carpeta de Programa	Ctrl+K	5	run once.
2 //	Incluir Librería	;	>	Administrar Bibliotecas Ctrl+Mayuus+L
3	Añadir fichero			Administral bibliotecas Ctri+Wayus+1
4 }				Añadir biblioteca .ZIP
5 6ªvoid 1				Arduino bibliotecas

Single ESCRIBIENDO MENSAJES PANTALLA LCD



1. Cargar librerías en el arduino:

una vez añadida, para poder usarla hay que ir a:

Programa-> incluir biblioteca/libreria-> LiquidCrystal

🔤 LCD	Arduino	IDE 2.3.4		Gestionar bibliotecas Ctrl + Mayús + I
Archivo	Editar	Sketch Herramientas Ayuda		Añadir hibliotoca, 7ID
		Verificar/Compilar	Ctrl + R	Anadir Dibiloteca .ZIP
		Cargar	Ctrl + U	Bibliotecas de Arduino
P	LCD.in	Configura y Subir		Arduino_BuiltIn
	1			EEPROM
	2	Cargar usando el programador	Ctrl + Mayús + U	Ethernet
1_)	З	Exportar binario compilado	Alt + Ctrl + S	Firmata
	4	Optimizar para depuración		Filliata
M	5	Mostrar carpeta de Sketch	Alt + Ctrl + K	HID
Second Sec	7	Incluir bibliotoco		Keyboard
0	8			LiquidCrystal
0		Añadir fichero		







1. Incluir en el IDE la librería LiquidCrystal.h



🥯 sketch_jun2	3a Arduino 1.8.19				
Archivo Editar	Programa Herramientas Ayuda				
sketch_jun	Verificar/Compilar Subir Subir Usando Programador Exportar Binarios compilados	Ctrl+R Ctrl+U Ctrl+Mayús+U Ctrl+Alt+S			
1ª v old 2 //	Mostrar Carpeta de Programa	Ctrl+K		run once.	
3	Incluir Librería	;	>	Administrar Bibliotecas Ctrl+Mavús+L	
5	Añadir fichero			Administrar Dibliotecus currindyus rr	
4 }				Añadir biblioteca .ZIP	
5 6∎ void	100p() {			Arduino bibliotecas	







 Para poder usar la librería tenemos que añadirla en el archivo. Vamos a añadir también wire.
 Pantalla_Arduino 1.8.19

Pantalla_ Ardu	ino 1.8.19	
Archivo Editar Pr	rograma Herramientas Ayuda	
	Verificar/Compilar Ctrl+ Subir Ctrl+	R U
Pantalla_§	Subir Usando Programador Ctrl+ Exportar Binarios compilados Ctrl+	Mayús+U Alt+S
	Mostrar Carpeta de Programa Ctrl+	κ
	Incluir Librería	>
void setup	Añadir fichero	Añadir biblioteca .ZIP
// put you	ur setup code here, to run	once: Arduino bibliotecas
1		Bridge
3		EEPROM
and lease ()	1	Esplora
void 100p()	1	Ethernet
// put you	ur marn code nere, co run .	Firmata
		GSM
		HID
		Keyboard
		LiquidCrystal
		Mouse
		Robot Control
		Robot IR Remote
RAES TU TURMO		Robot Motor
		SD
		SPI
		Servo
		SoftwareSerial
		SpacebrewYun
		Stepper
		TFT
		Temboo
		WiFi
		Wire
		Arduino-LiquidCorstal-I2C-librany-master
Guardado.		Arduno-Erquiderystal=ize=itbrary=master

😊 Pan	talla_ Arduino 1.8.19
Archive) Editar Programa Herramientas Ayuda
Pan	
#inc	lude <wire.h></wire.h>
#inc	lude <liquidcrystal_i2c.h></liquidcrystal_i2c.h>
void	setup() {
//	put your setup code here, to run once:
}	
void	loop() {
11	put your main code here, to run repeatedly
,	
}	







- 3. Declaramos un objeto basándonos en nuestra pantalla
- Para ello utilizaremos la siguiente función de la librería LiquidCrystal_I2C que acabamos de incluir.
 - LiquidCrystal_I2C lcd(0x27, 16, 2);
- Se inicializa con:
 - Ox27 porque se inicializa en esa dirección
 - 16 = número de caracteres por línea
 - 2 = número de líneas

Vale pero, ¿qué tenemos que hacer y cómo? Poco a poco







4. Aprendemos a usar la librería para utilizar cada punto de nuestra pantalla.

- Antes de empezar a usar las funciones de la librería necesitamos saber cómo usar las posiciones de cada línea.
- ¿Sabes qué es un vector?



- El vector tiene **posiciones**
- La primera posición siempre empieza en 0
- **Cada posición** tiene **un valor** (en la imagen, la posición 0 tiene valor 9, la posición 4 tiene valor 3...)
- El tamaño máximo del vector es el número de posiciones que tiene. En este ejemplo el vector tiene **6 posiciones que van del 0 al 5**.
- En nuestra pantalla, habrá dos vectores, que irán del... 0 al 15







- 4. Aprendemos a usar la librería para mostrar mensajes por pantalla:
- Para poder utilizar nuestra pantalla vamos a tener que aprender las funciones de la librería:
 - LCD. lcd.begin() se utiliza para inicializar la comunicación con la pantalla LCD.
 - Lcd.backlight() enciende la retroiluminación de la pantalla LCD.
 - Let. lcd.setCursor(0, 0) establece la posición del cursor en la columna 0 y la fila 0.
 - LCD. lcd.print("Hola") muestra el texto "Hola" en la pantalla LCD.
 - Lcd.clear() borra el contenido de la pantalla LCD.
 - Esas son las básicas, pero hay alguna más a ver si eres capaz de localizarlas ;)

¿Creéis que podríais hacer que la pantalla muestre en la primera línea "Hola"?







- 4. Aprendemos a usar la librería para mostrar mensajes por pantalla: EXTRA:
- Vamos a entrar más en profundidad en cómo se muestran las letras en pantalla:
 Siempre se escribe de la fila 0 posición 0 a la fila 0 posición 15
 segunda fila es la fila 1 posición 0 a la fila 1 posición 15









- 4. Aprendemos a usar las funciones de la librería para mostrar mensajes por pantalla:
 - Lcd.init() inicia la pantalla
 - Lcd.backlight() enciende la luz
 - Icd.setCursor(0, 0) posiciona el cursor
 - Lcd.print("Hola") escribe en la pantalla
 - Lcd.clear() borra de la pantalla

NOTA: Traduce este pseudocódigo a código real con las funciones que te hemos enseñado



```
#include <LiquidCrystal I2C.h>
#include <Wire.h>
LiquidCrystal I2C lcd(0x27, 16, 2);
 // Inicia el LCD en la dirección 0x27, con 16 caracteres y 2 líneas
void setup()
       inicio la pantalla
       luz de fondo
 void loop() {
       empiezo a escribir en línea0 pos0
       añado el texto que quiera
       espero 2 segundos
       limpio pantalla
```



Ahora tienes que **escribir en la pantalla** en la **primera línea la fecha** y en la

segunda línea la hora. (escribirlos de forma manual)

- Prueba diferentes formatos de fecha y hora
- Prueba a añadir espacios para centrar los mensajes
- Prueba a que se desplace hacia la derecha y/o izquierda



el texto (investiga en internet cual es la función que lo hace)

Elige el formato de fecha y hora que más te guste. por ej: lun 3 mar 2025, 3/3/2025...







https://controlautomaticoeducacion.com/sistemas-embebidos/arduino/bus-comunicaci on-i2c/





iqué Haremos Hoy?



Autoras: Elvira Castillo Twitter: @layoel Instagram: eca_ingeniera Aprenderemos a:

- Configurar el módulo de Reloj.
- Mostrar en el monitor serie la fecha y hora
- Mostrar en la pantalla la fecha y la hora.
- Crear funciones.
- Programar una alarma.





Conectando el hardware del Wake Me VP!





MÓDULO RELOJ DE TIEMPO REAL



- El arduino no es capaz de hacer varias cosas a la vez.
- Cuando se desconecta de la corriente y se vuelve a conectar, se reinicia el programa.
- Pero, si le conectamos un reloj, podemos indicar al arduino que realice diferentes acciones según la hora del día.
- Es por eso que vamos a usar este componente reloj que tiene unas características peculiares.









- El módulo reloj DS1302 RTC permite ajustar la fecha y la hora, contiene un reloj-calendario en tiempo real.
- El módulo reloj proporciona información de segundos, minutos, días, semanas, meses y años.
- Incluye una pila con la que puede mantener la actualización de la fecha y hora aunque se apague el arduino no se desconfigura a menos que le quitemos la pila.








Instrucciones para conectar el módulo de reloj al arduino

- Conectar **VCC** con el pin de **5V** de Arduino
- GND con el GND de Arduino
- **CLK** al pin **D6** de Arduino
- **DAT** con el pin **D7** de Arduino
- **RST** al pin **D8** de Arduino









Programando el Wake Me VP!









Descargamos la biblioteca del enlace: https://github.c om/chrisfryer78 /ArduinoRTClibr ary/tree/master

ode 🤆	🖸 Issues 👔 Pull requests 💿 Actions 🖽 Project	ts 🖽 Wiki 🛈 Sect	urity 🗠 Insights	
	양 master 👻 양 1 Branch 🟷 0 Tags		Q Go to file	Code
	chrisfryer78 Create virtuabotixRTC_version2.ino		E Clone	
	examples	Create virtuabotixR1	HTTPS GitHub CLI	
	🗅 README.md	Create README.md	https://github.com/chrisfryer78/ArduinoRTClib	ora
	virtuabotixRTC.cpp	Create virtuabotixR1	Clone using the web URL.	
	virtuabotixRTC.h	Create virtuabotixR1	덮 Open with GitHub Desktop	
		2	🔋 Download ZIP	

An easy to use real time clock library for Arduino, it was in the public domain, but not on GitHub, so I uploaded it.







	Añadimos la	*	🔤 306 306 306 🔗 G VH 🛅 G 🗙 +	- 0
		Bibliotecas de Arduino	IDE 2.1.1-nightly-20230628	– 🗆 X
	biblioteca al	Arduino_BuiltIn	ketch Herramientas Ayuda	
	IDE de	EEPROM Ethernet	Verificar/Compilar Ctrl + R	.Q. ∿
	IDLUC	Firmata	Cargar Ctrl + U	
	arduino	HID Keyboard LiquidCrystal Mouse	Configura y Subir Cargar usando el programador Ctrl + Mayús + U Exportar binario compilado Alt + Ctrl + S Optimizar para depuración	
		Servo	Mostrar carpeta de Sketch Alt + Ctrl + K atedly:	
		SoftwareSerial	Incluir biblioteca	
		Stepper TFT Wire		
		Bibliotecas aportadas		≣ 6
		Adafruit BuslO Adafruit GFX Library Adafruit NeoPixel	a instalada	
		Adafruit seesaw Library		
		Adafruit ST7735 and ST7789 Library		
	3 [ArduinoRTClibrary-master ESP8266 and ESP32 OLED driver for SSD1306 displays		
		IDromoto	Lín. 10, co	ol. 1 Arduino Nano en COM6 🗘 1 🗖





- Las **funciones** que podemos usar con este módulo de reloj son:
 - Para configurar la fecha y hora exactas

reloj.setDS1302Time(segundos, minutos, horas, día de la semana, dia del mes, mes, año)

Para consultar la fecha y hora reloj.updateTime()









- Comencemos con el programa:
- Incluye la librería
- Crea las variables con los pines de arduino donde has conectado el reloj.
- Creamos el objeto reloj utilizando la función virtuabotixRTC
- En el setup inicia el monitor serie
- Fija la configuración del reloj usando la función SET que vimos antes.
- En el loop actualiza la fecha y hora con la función UPDATE

	J Ardu	ino IDE 2.1.1-nightly-20230628 — 🗆
chiv	vo Edita	r Sketch Herramientas Ayuda
0	Ə 🚯	🖞 Arduino Nano 👻 🗸 🔎
-	reloj.ino	
	1	<pre>#include "virtuabotixRTC.h"</pre>
	2	
	3	//define los pines PinSCLK, PinDAT, pinRST donde van conectados al arduino
th	4	//creamos el objeto reloj
2	5	<pre>virtuabotixRTC reloj(PinSCLK, PinDAT, pinRST);</pre>
€>	6	
~	7	<pre>void setup() {</pre>
2	8	<pre>Serial.begin(9600);</pre>
	9	//reloj.set Configuración del tiempo actual
	10	// Recuerda comentar la línea anterior, una vez configurado
	11	}
	12	and second of
	13	<pre>void loop(){</pre>
	14	
	15	// actualiza fecha y hora
	16	// imprime en el monitor serie la fecha y la hora
	17	
	18	}







Cambia los comentarios por el código correspondiente. //define los pines... //reloj.set... //actualiza fecha y hora //imprime en el monitor serie la fecha y la hora. (lo subrayado lo veremos más adelante)









- Comencemos con el programa:
- Incluye la librería
- Crea las variables con los pines de arduino donde has conectado el reloj.
 - Creamos el obieto reloi utilizando la



IOTA: El reloj se configura una sola vez y se guarda para siempre mientras tenga la pila puesta el módulo.

Una vez configurado, comenta la línea del **reloj. Set** que escribiste en el setup para definir la fecha, hora, día etc...

con la función UPDATE









- Ya hemos visto que las **funciones** que podemos usar con este módulo de reloj son:
 - Para configurar la fecha y hora exactas

reloj.setDS1302Time(segundos, minutos, horas, día de la semana, dia del mes, mes, año)

- Para consultar la fecha y hora reloj.updateTime()
- Además, **después de usar updateTime** podemos consultar las **variables** imprimibles que son:
 - reloj.dayofmonth Muestra el día del mes
 - **reloj.dayofweek** Muestra el dia de la semana (en número, no letras)
 - **reloj.month** Muestra el mes (en número)
 - reloj.year Muestra el año
 - reloj.hours Muestra la hora
 - **reloj.minutes** Muestra los minutos
 - reloj.seconds Muestra los segundos







- Ahora, en el loop guarda los valores de las variables del reloj anteriores en palabras es decir:
 - Crea las variables de tipo int con los siguientes nombres: diadelmes, anio, hora, minuto, segundo
 - Crea las variables de **tipo String** con los siguientes nombres: diadelasemana, mes
 - Guarda en diadelmes reloj.dayofmonth, en anio reloj.year, en hora reloj.hours, en minuto reloj.minutes, en segundo reloj.seconds
 Las variables de tipo string las iniciamos con la cadena vacía las usaremos más adelante...
 - **Ejemplos:**

```
int diadelmes = reloj.....;
string diadelasemana = " ";
```









void loop() { // put your main code here, to run repeatedly: reloj.updateTime(); NOTA: escribe la variable correcta en int diadelmes = reloj....; int anio = reloj....; int hora = reloj....; int minuto = reloj....; int segundo = reloj....; String diadelasemana = " "; String mes = " ";

cada caso. Los string son cadenas vacías por lo que entre las "" llevan un espacio hasta que les demos otro valor que será más adelante.









- Imprime en el monitor serie el valor de las **variables**:
 - diadelmes
 - reloj.dayofweek
 - 📮 reloj.month
 - 📮 anio
 - 📮 hora
 - 🖬 minuto
 - segundo











Para darle formato, tienes que ponerlo por partes: imprime **reloj.dayofweek** imprime **diadelmes** imprime "/" imprime **reloj.month** imprime "/" imprime anio La hora igual también por partes: imprime hora imprime ":" imprime **minuto** imprime ":" imprime **segundo**







- **Continúa editando el programa para realizar los siguientes cambios:**
- Muestra el día de la semana en texto en el monitor serie.
 - Recuerda los **condicionales** que vimos en el módulo de introducción a arduino:
 - **El pseudo código** sería:
 - Si **reloj.dayofweek** es igual a 1 entonces

diadelasemana = "Lun "

escribe en el monitor serie diadelasemana;

si **reloj.dayofweek** es igual a 2 entonces

diadelasemana = "Mar"

escribe en el monitor serie diadelasemana;





```
if (reloj.month == 1){
   mes = " Ene ";
if (reloj.month == 2){
   mes = " Feb ";
if (reloj.month == 3){
   mes = " Mar ";
if (reloj.month == 4){
   mes = " Abr ";
if (reloj.month == 5){
   mes = " May ";
if (reloj.month == 6){
   mes = " Jun ";
if (reloj.month == 7){
   mes = " Jul ";
  }
```









- **Continúa editando el programa para realizar los siguientes cambios:**
- Muestra el **mes en texto** abreviado en el monitor serie. (Ene, Feb, Mar, Abr,...)
 - Es igual que los días de la semana, un **condicional**.
 - Le pseudo código sería:
 - Si **reloj.month** es igual a 1 entonces

mes = " Ene "

escribe en el monitor serie mes

Si **reloj.month** es igual a 2 entonces

mes = " Feb "

escribe en el monitor serie mes

....





```
if (reloj.month == 1){
   mes = " Ene ";
if (reloj.month == 2){
   mes = " Feb ";
if (reloj.month == 3){
   mes = " Mar ";
if (reloj.month == 4){
   mes = " Abr ";
if (reloj.month == 5){
   mes = " May ";
if (reloj.month == 6){
   mes = " Jun ";
if (reloj.month == 7){
   mes = " Jul ";
```







Ahora que ya vemos la fecha y hora en el monitor serie...

¿Serías capaz de mostrarla en la pantalla

LCD que programamos ayer?







- Ahora que ya vemos la fecha y hora en el monitor serie...
- Una ayudita... tan solo tienes que:
 - Incluir en el código que tienes para el reloj en el código de ayer de la pantalla. (te recomiendo que crees un scketch nuevo que se llame PantallaReloj donde unas ambos códigos (en lugar de editar los que ya te funcionan de manera individual será más fácil detectar errores)
 - En el **setup** iniciar la pantalla y la luz de fondo.
 - Escribe en la línea 0 la fecha de hoy usando las función de la lcd para definir la posición del cursor y la función de imprimir con las variables que hemos creado en el programa del reloj (las numéricas actualizar_hora, diadelmes, reloj.month, anio, hora, minuto, segundo, o si lo prefieres combinalas con las que están en palabra diadelasemana, mes,)
 - Escribe en la línea 1 la hora (usa el formato que más te guste, también puedes añadir el dia de la semana en texto o el mes en texto como hicimos en el ejemplo anterior)
 - De la pantalla periódicamente.











 Ahora que ya sabemos como escribir en la pantalla, vamos a crear un código que: Actualiza el valor de la variable actualizar_hora igual a reloj.updateTime(); Si hora es mayor o igual que 7 y hora es menor que 12 Escribe en la primera línea de la pantalla el mensaje ¡Buenos Días! sino Si hora es mayor que 12 y hora es menor que 19 Escribe en la primera línea de la pantalla el mensaje ¡Buenas Tardes!

sino (en cualquier otro caso):

Escribe el mensaje ¡Buenas noches! en la pantalla.

Escribe en la segunda línea de la pantalla la fecha de hoy





7. Crear funciones.

Oficina de Software Libre

- Las funciones son **agrupaciones de código**
- Ya conocemos las funciones
 - □ void setup(){ ... }
 - void loop() { ... }
- Cuando delante del nombre de la función aparece void significa que la función no devuelve nada.
- Las funciones pueden devolver variables.
- Delante del nombre de la función hay que añadir el tipo de variable que devuelve. /
- Esta función es void porque no devuelve nada, ejecuta todo lo de dentro, pero no devuelve ninguna variable.

UNIVERSIDAD **DEGRANADA** void saludofecha(int hora, String diadelasemana, int diadelmes, String mes, int anio){ METE AQUÍ DENTRO EL CÓDIGO QUE HAS ESCRITO Esto hav que hacerlo en un nuevo skech. Las funciones

se añaden debajo del loop o encima del setup





- Ahora que vas teniendo mucha funcionalidad, vamos a pasar el código a funciones para que el loop se vea más limpio y sea más fácil de leer.
- Copia el código correspondiente de cada funcionalidad dentro de su función.
- Las funciones se definen fuera del loop.
- En la siguiente diapositiva tienes la estructura de las funciones.
- Cambia los ... por el código que tienes escrito que hace esa funcionalidad dentro de tu loop.

```
void saludofecha(int hora, String diadelasemana,
                                                                                                UNIVERSIDAD
           Oficina de
Software Libre
Universidad
                    int diadelmes, String mes, int anio ){
                                                                                                DE GRANADA
                       . . .
En estos casos no
  hay return, la
   función no
devuelve ningún
                    void fechahora(String diadelasemana, int diadelmes,
valor, por eso es de
                    String mes, int anio, int hora, int minuto, int segundo){
    tipo void
                       . . .
                    //escribimos los dias de la semana en texto
Si te fijas, en estos
                    String DimeDia(int numDia){
casos devuelven
                       String dia= " ";
 una cadena de
 texto, por eso
                       . . .
delante del nombre
                       return dia;
  de la función
 aparece el tipo
String. Tanto dia
 como mes son
                       escribimos los meses abreviados
texto, son strings.
                    String DimeMes(int numMes){
                       String mes= " ";
                       . . .
                       return mes;
```





Ahora el loop debe quedar así:

```
void loop() {
  // put your main code here, to run repeatedly:
 reloj.updateTime();
 int diadelmes = reloj.dayofmonth ;
  int anio = reloj.year ;
  int hora = reloj.hours ;
  int minuto = reloj.minutes ;
  int segundo = reloj.seconds ;
  String diadelasemana = " ";
  String mes = " ";
mes = DimeMes(reloj.month);
diadelasemana = DimeDia(reloj.dayofweek);
//fechahora(diadelasemana, diadelmes, mes, anio, hora, minuto, segundo);
saludofecha(hora, diadelasemana, diadelmes, mes, anio);
```







Ahora que ya tenemos el código más limpio, vamos a programar nuestra alarma. Tenemos que crear dos variables nuevas:

- alarm_hora y alarm_minutos
- ahora os dejo el pseudo código:

Si (hora es igual a alarm_hora Y minutos es igual a alarm_minutos){ limpia la pantalla pon el cursor en la posición inicial escribe en la pantalla ¡Despierta! espera medio segundo





```
void loop() {
 // put your main code here, to run repeatedly:
 reloj.updateTime();
 int diadelmes = reloj.dayofmonth ;
 int anio = reloj.year ;
 int hora = reloj.hours ;
 int minuto = reloj.minutes ;
 int segundo = reloj.seconds ;
 String diadelasemana = " ";
 String mes = " ";
 //¿a que hora saltará la alarma?
 int alarm hora = 23;//hora de programación de la alarma
 int alarm_minuto = 35; //minutos de programación de la alarma
 mes = DimeMes(reloj.month);
 diadelasemana = DimeDia(reloj.dayofweek);
 lcd.clear();
 //si es la hora, saltará la alarma durante 1 minuto)
  if(hora == alarm_hora && minuto == alarm_minuto){
     Thunder();
     lcd.clear();
     lcd.setCursor(0, 0);
     lcd.print(";Despierta!");
     delay(500);
```

UNIVERSIDAD DE GRANADA



MÓDULO RELOJ DE TIEMPO REAL



El arduino no es capaz de hacer varias cosas a la vez.

Cuando se desconecta de la corriente y se vuelve a conectar, se reinicia.

Ahora programamos que en la pantalla aparezca la fecha y la hora actual.

Este coincide los pines que uso en la imagen con el ejemplo https://eloctavobit.com/modulos-sensores/real-time-cl

ock-rtc-ds1302-modulo-reloj

https://www.youtube.com/watch?v= QNrMw79qZ8I https://miliohm.com/ds1 302-rtc-with-arduino-tut orial/ codigo de prueba

> PROGRAMAR acciones con el reloj DS1302 RTC en Arduino

https://www.youtube.com/watch?v= u3kUj-hV6Rk

https://www.tecneu.com/blogs/tutoriales-de-electronica/integracion-del-ds1302-con-arduino-configuracion-del-fecha-y-hora?srsltid=AfmBOorrgAdgS2MnOgaHgb1fnwlrtPlvSenhev3SynbwMhimW3OP2h9C









Autoras: Elvira Castillo Twitter: @layoel Instagram: eca_ingeniera

Aprenderemos a

- Conectar los leds.
- Programar los leds, creando diferentes secuencias.





Montaje del *Wake Me VP!*









Leds RGB direccionables.

Tienen 4 pines:

- DIN pin de entrada
- **DO** pin de salida
- □ +5V proporciona energía a los leds
- GND Toma de tierra

Si usaramos otro anillo de led, el **DIN** del primero debe ir conectado al **DO** del segundo.









- Leds RGB direccionables.
- Se llaman direccionables porque las señales se transmiten en una dirección a través de los leds.
- Si fuera una tira de led, podríamos ver cómo están conectados como aparece en las imágenes
- El anillo está conectado igual pero no lo vemos.
- Tu cable del pin correspondiente de arduino siempre tiene que ir al DIN de tu anillo de leds.













Vamos a soldar los cables Pero...









Conectando el hardware del Wake Me VP!









Leds RGB direccionables.

Se llaman direccionables porque las señales se transmiten en una dirección a través

de los leds

DIN

GND

5Va

Conecta

Nota: El anillo de led es igual que la tira de led (ambos están formados por Leds RGB direccionables con chip propio integrado) pero en el caso del anillo su localización es circular. Las conexiones son las mismas pero en el anillo están integradas y sólo nos muestra, en el inicio del anillo los pines DIN GND VCC/5V y en el fin del anillo el DO en lugar de aparecer entre todos los leds como en la tira de led.











Leds RGB direccionables.

- Se llaman direccionables porque las señales se transmiten en una dirección a través de los leds. Del Arduino al primer led, al segundo, al tercero... hasta el último
- El chip que contienen integrado, "asigna un número" (dirección de memoria) a cada led para identificarlo en la tira.




RGB Ring Connection Figure 3









Programando el WAKE ME VP!









Abrimos un nuevo sketch de arduino en el IDE para hacer pruebas con el aro de leds. Después lo integraremos con todo lo anterior







1. Incluir en el IDE la librería Adafruit_NeoPixel

Hay **dos formas de incluir librerías** a la biblioteca de nuestro arduino.

- Si tenemos el archivo comprimido con extensión .zip
- Buscarlas en el repositorio de Arduino

- Para la tira de led tenemos el archivo comprimido con el nombre
 Adafruit_NeoPixel.zip
- La **descargamos** y la incluimos de la 1º forma.







1. Incluir en el IDE la librería Adafruit_NeoPixel

Vamos al enlace y descargamos la librería haciendo clic en download









1. Incluir en el IDE la librería Adafruit_NeoPixel



🥺 sketch_jun23a	a Arduino 1.8.19		
Archivo Editar Pr	ograma Herramientas Ayuda		
sketch_jun	Verificar/Compilar Subir Subir Usando Programador	Ctrl+R Ctrl+U Ctrl+Mayús+U	
1 ⊓void 2 //	Exportar Binarios compilados Ctrl+Alt+S Mostrar Carpeta de Programa Ctrl+K		- run once.
3	Incluir Librería > Añadir fichero		Administrar Bibliotecas Ctrl+Mayús+I
4) 5 6ªvoid 1			Añadir biblioteca .ZIP Arduino bibliotecas







2. Para poder usar la librería tenemos que añadirla en el sketch.

etch_jun23a Arduino 1.8.19	
vo Editar Programa Herramientas Ayuda	
Verificar/Compilar Ctrl+R	
Subir Ctrl+U	
etch jun Subir Usando Programador Ctrl+Mayús	+U
Exportar Binarios compilados Ctrl+Alt+S	
/ / Mostrar Carpeta de Programa Ctrl+K	END ODGO
Incluir Librería	> MAGE
Añadir fichero	VVIFI
}	Contribución bibliotecas
	AccelStepper
oid loop() {	Adafruit BuslO
// nut your main code here	Adafruit_SH1106-master
// put your main code here,	ArduinoSTL
	BatReader
	CanSat Kit Library
	EEPROM
	Remete
	LedMatrix
	LiquidCovstal-I2C
	MaxMatrix
	NewPing
	OLED SSD1306 - SH1106
	Oscillator
	Otto
KHES TUTUK IO	OttoSerialCommand
	RadioHead_master
	SPI
	SoftwareSerial
	TLE94112
	US
	Wire
	openCanSat
	Recomendado bibliotecas
	Adafruit BMP280 Library
	Adafruit Circuit Playground
	Adafruit GFX Library
	Adafruit NeoPixel
	Adafruit PWM Servo Driver Library
	Adatruit Unified Sensor









- **3.** Declaramos un objeto con la cadena de leds que vamos a emplear.
- Para ello utilizaremos la siguiente función de la librería NeoPixel que acabamos de incluir.
 - Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
- Los parámetros que se pasan a la función son:
 - **NUMPIXELS** que será una variable donde pondremos el número de leds que queremos controlar
 - PIN que será el pin de arduino donde hemos conectado el IN de la tira de led
 - NEO_GRB + NEO_KHZ800 este parámetro indica que usaremos leds de colores y la controladora que utilizan.

□ Vale pero, ¿qué tenemos que hacer y cómo?









Declaramos un objeto con la cadena de leds que vamos a emplear. 3.

Aquí tenéis **un ejemplo**

para controlar 3 leds,

vuestra tarea es adaptarlo para poder usar los 24 leds.

> Esto hay que hacerlo en el sketch donde vamos a programar los leds para probarlos y añadirlo al sketch del programa final

🥺 sketch_jun23a Arduino 1.8.19 — 🗆						
Archivo Editar Programa Herramientas Ayuda						
sketch_jun23a §						
1 #include <adafruit_neopixel.h></adafruit_neopixel.h>						
3 #define PIN 12 4 #define NUMPIXELS 3						
5 6 <mark>using namespace std;</mark> 7						
8 Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);						
10 void setup() {						
11 // nut your setup code here, to run once.						
12						
13 }						
14						
$15^{\text{void}} \log()$						
16 // put your main code here, to run repeatedly:						
17						
18 }						







- 4. Aprendemos a usar la librería para encender uno a uno los leds del aro de led de colores.
- Antes de empezar a usar las funciones de la librería necesitamos saber cómo acceder a las posiciones de cada led.
- Sabes que es un **vector**?



- El vector tiene **posiciones**
- La primera posición siempre empieza en 0
- **Cada posición** tiene **un valor** (en la imagen, la posición 0 tiene valor 9, la posición 4 tiene valor 3...)
- El tamaño máximo del vector es el número de posiciones que tiene. En este ejemplo el vector tiene **6 posiciones que van del 0 al 5**.
- Los led se encienden en el mismo orden que el vector del 0 al 23 en nuestro caso si usas un aro o al 47 si usas dos.







4. Aprendemos a usar la librería para encender uno a uno los leds del aro de led de colores.



Cómo encedemos un led?









4. Aprendemos a usar la librería para encender uno a uno los leds de la tira de led de colores.



Cómo encedemos un led?

iUsaremos las funciones ya predefinidas de la librería Neopixel!









- 4. Aprendemos a usar la librería para encender uno a uno los leds de la tira de led de colores.
- Utilizaremos estas funciones ya predefinidas de la librería Neopixel.
 - **pixels.begin();** esta función inicializa los leds.
 - **pixels.clear();** esta función borra la configuración guardada en los leds.
 - **pixels.Color(R, G, B);** esta función es para seleccionar el color.
 - **R** es el parámetro donde se pondrá el valor del color **rojo** (toma valores de 0 a 255)
 - G es el parámetro donde se pondrá el valor del color **verde** (toma valores de 0 a 255)
 - **B** es el parámetro donde se pondrá el valor del color **azul** (toma valores de 0 a 255)
 - pixels.setPixelColor(led, color);
 - led es el número de led que queremos que se encienda del color que pongamos.
 - color es el color que se va a poner y se sustituye por: pixels.Color(R, G, B);







- 4. Aprendemos a usar la librería para encender uno a uno los leds de la tira de led de colores.
 1 #include <Adafruit_NeoPixel.h>
- En este ejemplo se usan 4 leds.



- Qué leds se encienden?
- En qué color?
- Adapta el código para los 24 leds
- Ve haciendo pruebas para encender uno u otro led o varios a la vez y probar los diferentes colores.

Esto hay que hacerlo en el sketch donde vamos a crear el programa del reloj

```
3 #define PIN
                     12
4 #define NUMPIXELS
6 using namespace std;
8 Adafruit NeoPixel pixels (NUMPIXELS, PIN, NEO GRB + NEO KHZ800);
13<sup>e</sup>void setup() {
    // put your setup code here, to run once:
14
    pixels.begin();
19<sup>e</sup>void loop() {
    // put your main code here, to run repeatedly:
    pixels.clear();
    pixels.setPixelColor(0, pixels.Color(30, 0, 0));
    pixels.setPixelColor(1, pixels.Color(0, 30, 0));
    pixels.setPixelColor(2, pixels.Color(0, 0, 30));
    pixels.setPixelColor(3, pixels.Color(30, 30, 30));
```







- Para poder hacer esto, vamos a aprender a escribir condicionales.
 - Condicional simple
 - **Gi** es cierto... entonces...
 - Condicional doble
 - Si es cierto... entonces.... sino... entonces...
 - **Condicional compuesto**
 - Si esto o esto es cierto... entonces...
 - **Si** esto **y** esto es cierto... **entonces...**









- 5. Encender de uno en uno los led pares en rojo y led impares en verde
- Para hacer condicionales necesitamos operadores que pueden ser:
 - Matemáticos:
 - □ +, -, *, /, %.
 - **Relacionales:**

□ ==,!=, <, >, >=, <=

Lógicos

□ && , and, ||, or, !

S	Р	!P	
	True	False	
	False	True	

Р	Q	P&&Q	P Q
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False







Podemos usar un condicional simple usando los operadores == (igual) y != (distinto)

```
If (condición){
    sentencias;
```

== ➡ ¿Son iguales? != ➡ ¿Son distintos?









Veamos que hace el código:

- Creo una variable numLed y le asigno el valor 2. (línea 21)
- Creo una variable calculo a la que le asigno el resto de dividir numLed entre 2 (para saber si es par o impar) línea 23.
- Gine Si el valor de calculo es igual a 0
 - entonces enciende el led numLed de color rojo.
- Si el valor de calculo es distinto de 0
 - entonces enciende el led numLed de color verde.

```
8 Adafruit NeoPixel pixels (NUMPIXELS, PIN, NEO GRB + NEO KHZ800);
 9
10=
11
12
13<sup>o</sup>void setup() {
14
     // put your setup code here, to run once:
15
     pixels.begin();
16
17 }
18
19<sup>e</sup>void loop() {
20
21
     int numLed = 2; //El led 2
    /*Calcula par o impar. si el resto es 0 es par sino es impar*/
22
23
     int calculo = numLed \% 2;
24
25∎
     if(calculo == 0){
26
       pixels.setPixelColor(numLed, pixels.Color(30, 0, 0));
27
28
298
     if(calculo != 0){
30
       pixels.setPixelColor(numLed, pixels.Color(0, 30, 0));
```







```
8 Adafruit NeoPixel pixels (NUMPIXELS, PIN, NEO GRB + NEO KHZ800);
                                          9
                                         10
Veamos que hace el código:
                                         11
Creo una variable numLed y le
                                         12
                                         13<sup>o</sup>void setup() {
      asigno el valor 2. (línea 21)
                                             // put your setup code here, to run once:
                                         14
      Creo una variable calculo a la
15
                                             pixels.begin();
      que le asigno el resto de dividir
                                        16
                                        17 }
      numLed entre 2 (para saber si es
                                         18
      par o impar) línea 23.
                                        19<sup>e</sup>void loop() {
      Si el valor de calculo es igual a 0
int numLed = 2; //El led 2
           entonces enciende el led
       /*Calcula par o impar. si el resto es 0 es par sino es impar*/
            numLed de color rojo.
                                             int calculo = numLed \% 2;
      Si el valor de calculo es distinto
if(calculo == 0) {
      de 0
                                         26
                                             entonces enciende el led
      numLed de color verde.
                                             if(calculo != 0){
                                               pixels.setPixelColor(numLed, pixels.Color(0, 30, 0));
```







Y si queremos que haga las comprobaciones de manera automática para todos los leds?

iUsaremos bucles!









Bucles

```
Loop(){
Lo que se va a repetir siempre
}
```

```
for (inicio; fin; incremento){
   Lo que se va a repetir desde inicio a fin
}
```

while(condición){
 lo que quiero que haga mientras se cumpla la condición
}







- 5. Encender de uno en uno los led pares en rojo y led impares en verde
 - Bucles:
- Loop(){
 Lo que se va a repetir siempre
 }
- for (inicio; fin; incremento){
 Lo que se va a repetir desde inicio a fin
 }

while(*condición*){ lo que quiero que haga mientras se cumpla la condición







```
    Bucles:
    Loop(){
        Lo que se va a repetir siempre
        }
```

```
± / J
18_
19 void loop() {
20
21
    int numLed = 2; //El led 2
    /*Calcula par o impar. si el resto es 0 es par sino es impar*/
22
     int calculo = numLed % 2;
23
24
    if(calculo == 0){
250
26
      pixels.setPixelColor(numLed, pixels.Color(30, 0, 0));
27
     }else{
28
         pixels.setPixelColor(numLed, pixels.Color(0, 30, 0));
29
```







- 5. Encender de uno en uno los led pares en rojo y led impares en verde
 - Bucles:
 - for (inicio; fin; incremento){
 Lo que se va a repetir desde inicio a fin
 }







- 5. Encender de uno en uno los led pares en rojo y led impares en verde
 - Bucles:
 - while(*condición*){ lo que quiero que haga mientras se cumpla la condición }







6. Aprendemos el condicional simple y el condicional compuesto









6. Aprendemos el condicional simple y el condicional compuesto

- Veamos el ejemplo de par o impar, usando un condicional doble.
- Sería:
 - Si el cálculo es igual a cero,
 - entonces enciende el led2 de color rojo.
 - Sino,
 - entonces enciende el led 2 de color verde.

```
#detine NUMPIXELS 3
 6 using namespace std;
 8 Adafruit NeoPixel pixels (NUMPIXELS, PIN, NEO GRB + NEO KHZ800);
 9
10
11
12
13<sup>a</sup>void setup() {
     // put your setup code here, to run once:
14
15
     pixels.begin();
16
17 }
18
19<sup>e</sup>void loop() {
20
21
     int numLed = 2; //El led 2
22
    /*Calcula par o impar. si el resto es 0 es par sino es impar*/
23
    int calculo = numLed % 2;
24
25
    if(calculo == 0){
26
       pixels.setPixelColor(numLed, pixels.Color(30, 0, 0));
27
     }else{
28
         pixels.setPixelColor(numLed, pixels.Color(0, 30, 0));
29
```







6. Aprendemos el condicional simple y el condicional compuesto

```
#define NUMPIXELS 3
Veamos el ejemplo de par o
                                                    6 using namespace std;
                                                    8 Adafruit NeoPixel pixels (NUMPIXELS, PIN, NEO GRB + NEO KHZ800);
impar, usando un condicional
                                                    9
doble.
                                                   11
                                                   12
Sería:
                                                   13<sup>a</sup>void setup() {
                                                        // put your setup code here, to run once:
                                                   14
       Si el cálculo es igual a cero,
 15
                                                        pixels.begin();
             entonces enciende el led2 de
        16
                                                   17 }
             color rojo.
                                                   18
                                                   19<sup>e</sup>void loop() {
 Sino,
             entonces enciende el led 2 de
        int numLed = 2; //El led 2
                                                   21
                                                       /*Calcula par o impar. si el resto es 0 es par sino es impar*/
             color verde.
                                                        int calculo = numLed % 2;
                                                       if(calculo == 0){
                                                         pixels.setPixelColor(numLed, pixels.Color(30, 0, 0));
                                                      >> else {
                                                            pixels.setPixelColor(numLed, pixels.Color(0, 30, 0));
```







Ahora que ya sabes encender los leds, edita el código para que, se encienda de uno en uno los led pero que se apague el anterior.

Te dejo el algoritmo. Piensa como se programa aplicando lo aprendido antes.



Para i=0 mientras i sea menor que el número de pixel incrementa i en 1

función que pone el color al pixel i

si i es distinto de 0

Esto hay que hacerlo en el sketch del reloj de arena

función que pone el color al pixel i anterior y poner el color 0 que es apagado aplicar la configuración a los pixel esperar medio segundo







7. Crear funciones.

- Las funciones son a**grupaciones de código**
- Ya conocemos las funciones
 - void setup(){ ... }
 - void loop() { ... }
- Cuando delante del nombre de la función aparece void significa que la función no devuelve nada.
- Las funciones pueden devolver variables.
- Delante del nombre de la función hay que añadir el tipo de variable que devuelve.
- Esta función es void porque no devuelve nada, aplica la configuración pero no devuelve ninguna variable.









¿Recuerdas cuando creamos la alarma?

Este era el pseudo código:

Si (hora es igual a alarm_hora Y minutos es igual a alarm_minutos) limpia la pantalla

pon el cursor en la posición inicial

escribe en la pantalla ¡Despierta! espera medio segundo

Ahora cuando salte la alarma **¡haz que se encienda la secuencia** de luces que más te guste!







void loop() { // put your main code here, to run repeatedly: reloj.updateTime(); int diadelmes = reloj.dayofmonth ; int anio = reloj.year ; int hora = reloj.hours ; int minuto = reloj.minutes ; int segundo = reloj.seconds ; String diadelasemana = " "; String mes = " "; //¿a que hora saltará la alarma? int alarm hora = 23;//hora de programación de la alarma int alarm minuto = 35; //minutos de programación de la alarma mes = DimeMes(reloj.month); diadelasemana = DimeDia(reloj.dayofweek); lcd.clear(); //si es la hora, saltará la alarma durante 1 minuto) if(hora == alarm hora && minuto == alarm minuto){ Thunder(); lcd.clear(); lcd.setCursor(0, 0); lcd.print(";Despierta!"); delay(500); fechahora(diadelasemana, diadelmes, mes, anio, hora, minuto, segundo);





iqué Haremos Hoy?



Autoras: Elvira Castillo Twitter: @layoel Instagram: eca_ingeniera

Aprenderemos a:

- Conectar el sensor IR.
- Detectar los códigos de un mando cualquiera.
- Utilizar los códigos
 para interactuar con el reloj despertador.





Conectando el hardware del Wate Me VP!



<u> https://sereingeniera.ugr.es/index.php/2021/07/08/9-de-julio-dia-6-juntemos-todo/</u>







- Conectamos el **sensorIR del mando**
 - El sensor tiene 3 patillas

S VCC

- S va al pin de arduino que enviará la señal pin2
- VCC va al pin de arduino 5v
- va al pin tierra GND de Arduino

OJO mirad bien el nombre de los pines en el componente. Algunos vienen cambiados (podéis usar el polímetro para comprobarlo)







Montaje del WARE ME VP!




icontrol remoto!



25

Programar el mando IR

- Incluimos la librería para el mando
- Declaramos la variable donde tenemos conectado el pin del IR
- En el setup habilitamos la recepción del IR.
- □ Si se decodifica:
 - imprimir el hexadecimal
 - imprimir la decodificación
 - □ imprimir seleccionar—>
 - imprimir la tecla
 - imprimir ------
 - enviar datos.

```
N
🥯 Mando IR Arduino 1.8.13
                                                                 GND
                                                             DUT
Archivo Editar Programa Herramientas Ayuda
        Nuevo
                                                                         LSO
                                                     to pin 2
  Mando IR
#include <IRremote.hpp>
                                                                 GND
const int RECV PIN = 2;
void setup() {
  Serial.begin(9600);
  //comando para establecer la comunicación con el sensor IR
  IrReceiver.begin (RECV PIN, ENABLE LED FEEDBACK);
void loop() {
  //con este código podemos obtener los valores obtenidos por nuestro mando
  if (IrReceiver.decode()) {
    Serial.println(IrReceiver.decodedIRData.decodedRawData,HEX);
    Serial.println(IrReceiver.decodedIRData.decodedRawData);
    Serial.println("Seleccionar--> ");
    Serial.println(IrReceiver.decodedIRData.command);
    Serial.println("\n-----");
```

IrReceiver.resume();

https://www.luisllamas.es/arduino-mando-a-distancia-infrarrojo/







- Utilizando el programa anterior:
- Rellenar las variables con el valor que obtengamos al pulsar la tecla del mando.
- Con el valor que te devuelve por el puerto serie, rellena el valor de las variables (cambiando el 00 por el valor que se muestra en el puerto serie que será un número entero distinto para cada tecla.

const int KEY_UP = 00;
const int KEY_LEFT = 00;
const int KEY_OK = 00;
<pre>const int KEY_RIGHT = 00;</pre>
const int KEY_DOWN = 00;
<pre>const int KEY_1 = 00;</pre>
const int KEY_2 = 00;
<pre>const int KEY_3 = 00;</pre>
const int KEY_4 = 00;
const int KEY_5 = 00;
const int KEY_6 = 00;
const int KEY_7 = 00;
const int KEY_8 = 00;
<pre>const int KEY_9 = 00;</pre>
<pre>const int KEY_0 = 00;</pre>
<pre>const int KEY_ASTERISK = 00;</pre>
const int KEY_POUND = 00;







 Una vez tenemos la codificación, vamos a utilizar otra estructura condicional que es switch case.









- Crea una variable (fuera del setup y del loop) que sea de tipo entero que se llame teclaPulsada.
- Según la estructura de switch case,
 añade un case para cada tecla que
 vayas a usar.
- Dentro escribirás lo que quieres que haga cada tecla cuando la pulses.

void loop() {

```
//leer del mando la tecla pulsada
if (IrReceiver.decode()) {
    if(IrReceiver.decodedIRData.decodedRawData != 0){
      teclaPulsada = IrReceiver.decodedIRData.command;
    }else{
     teclaPulsada = 0;
   IrReceiver.resume();
// Esto es como un "if else" ---> if teclapulsada es el switch y
// cada uno de los case son los else
switch(teclaPulsada){
   case KEY 1:
  //aqui lo que quieres que haga si pulsa la tecla keyl
  break:
  case KEY 2:
  //aqui lo que quieres que haga si pulsa la tecla key2
   break:
  //sique añadiendo case key... para cada una de las teclas
  //del mando que vas a utilizar
```







```
En mi caso utilizo las funciones que creamos anteriormente y alguna más.
switch (teclaPulsada){
  case KEY 1: //muestra el buenos días y la fecha en la pantalla si quieres añade luz
    saludofecha(hora, diadelasemana, diadelmes, mes, anio);
    break:
  case KEY 2: //muestra la fecha y la hora en la pantalla. Aqui no puedes usar luz ni sonido.
   fechahora(diadelasemana, diadelmes, mes, anio, hora, minuto, segundo);
    break:
  case KEY 3: // funcion que muestra a que hora se ha programado la alarma
    programada alarma(alarm hora, alarm minuto);
    break;
  case KEY RIGHT: //apaga las luces de tu lampara.
      ApagarLuz();
  break:
  case KEY LEFT: //enciende las luces de tu lampara.
      Luz Ambiente();
  break;
delav(500);
```







Como se trata de hacer un reloj despertador, podemos añadir en los cases de las teclas las funciones que ya tenemos creadas, o añadir nuevas.

Por ejemplo estas son las funciones que yo he creado:

- programada_alarma (alarm_hora, alarm_minuto);
 Muestra en la LCD a que hora está programada la alarma
- saludofecha (hora, diadelasemana, diadelmes, mes, anio);
 Muestra en la LCD buenos días, tardes o noches dependiendo de la hora y debajo el día.
- fechahora (diadelasemana, diadelmes, mes, anio, hora, minuto, segundo);
 Muestra en la LCD la fecha y en la siguiente línea la hora
- ApagarLuz();

Apaga todos los leds del anillo de leds

Luz_Ambiente();

Enciende todos los leds de color blanco.

Pero tu puedes crear diferentes funciones con colores para cambiar de color los leds o secuencias.





DE LAS QUE TE FALTAN?

¿TE ANIMAS A CREAR EL CÓDIGO

switch (teclaPulsada){

```
case KEY 1: //muestra el buenos días y la fecha en la pantalla si guieres añade luz
  saludofecha(hora, diadelasemana, diadelmes, mes, anio);
  break;
case KEY 2: //muestra la fecha y la hora en la pantalla. Aqui no puedes usar luz ni sonido.
 fechahora(diadelasemana, diadelmes, mes, anio, hora, minuto, segundo);
  break;
case KEY 3: // funcion que muestra a que hora se ha programado la alarma
  programada alarma(alarm hora, alarm minuto);
  break;
case KEY_RIGHT: //apaga las luces de tu lampara.
    ApagarLuz();
break:
case KEY LEFT: //enciende las luces de tu lampara.
    Luz_Ambiente();
break;
```





UNIVERSIDAD DE GRANADA

iqué HAREMOS HOY?



Autoras: Elvira Castillo Twitter: @layoel Instagram: eca_ingeniera

Aprenderemos a

- Conectar el buzzer.
- Programar diferentes melodías para el despertador.





Montaje del *Wake Me VP!*









- Usaremos ahora el zumbador(*buzzer*) para crear una melodía:
- ★ Una de despertador, cuando suene la alarma









- Usaremos ahora el zumbador(*buzzer*) para crear una melodía:
- ★ Una de despertador, cuando suene la alarma

El buzzer se conecta al pin 10







Programando el Wake Me VP!









Recordad...

- ★ No podemos usar dos funciones tones() al mismo tiempo.
 Se debe de dejar un pequeño intervalo de tiempo para que suene la nota → delay(time)
- ★ La frecuencia de sonido va desde 31 Hz a 65535 Hz.









				FRECUENCI	A DE LAS NOTAS	MUSICALES EN H	ERCIOS (Hz)			
- 2		OCTAVA 0	OCTAVA 1	OCTAVA 2	OCTAVA 3	OCTAVA 4	OCTAVA 5	OCTAVA 6	OCTAVA 7	OCTAVA 8
- 1	Do	16,3516	32,7032	65,4064	130,813	261,626	523,251	1046,50	2093,00	4186,01
[Do# / Reb	17,3239	34,6479	69,2957	138,591	277,183	554,365	1108,73	2217,46	4434.92
N	Re	18,3540	36,7081	73,4162	146,832	293,665	587,330	1174,66	2349,32	4698,64
fu	Re# / Mib	19,4454	38,8909	77,7817	155,563	311,127	622,254	1244,51	2489,02	4978,04
ti€	Mi	20,6017	41,2035	82,4069	164,814	329,628	659,255	1318,51	2637,02	5274,04
Sŧ	Fa	21,8268	43,6536	87,3071	174,614	349,228	698,456	1396,91	2793,83	5587,66
ne	Fa# / Solb	23,1246	46,2493	92,4986	184,997	369,994	739,989	1479,98	2959,96	5919,92
+1/	Sol	24,4997	48,9995	97,9989	195,998	391,995	783,991	1567,98	3135,96	6271,92
LIC	Sol# / Lab	25,9565	51,9130	103,826	207,652	415,305	830,609	1661,22	3322,44	6644,88
nc	La	27,5000	55,0000	110,000	220,000	440,000	880,000	1760,00	3520,00	7040,00
- 1	La# / Sib	29,1353	58,2705	116,541	233,082	466,164	932,328	1864,66	3729,31	7458,62
ъÎ	Si	30,8677	61,7354	123,471	246,942	493,883	987,767	1975,53	3951,07	7902,14

desde **31 Hz** a **65535 Hz**.



VEMOS LA ESTRUCTURA DEL PROGRAMA!

1

6

16



Programar un buzzer

×

Pasa este pseudocódigo a lenguaje del arduino.

```
/*
     Inicializamos buzzer
 2
 3
     */
 5
     void setup() {
       // put your setup code here, to run once:
 7
       buzzer es un pin de salida (OUTPUT)
 8
 9
10
     void loop() {
11
       // put your main code here, to run repeatedly:
12
       suena el buzzer con una frecuencia
13
       dejar x milisegundos
14
       apagar el sonido del buzzer
15
```













Se haría algo parecido a como se explica en este código.

- Veamos el ejemplo de música.
- Primero, inicializamos el buzzer.
 - Se conecta al pin 10.
 - Además, en void setup(), se indica en pinMode(buzzer), OUTPUT) que es OUTPUT(salida).
 - Después, en el void loop(), indicamos con tone(), la frecuencia que queremos.
 - Después dejamos que suene durante x milisegundos.
 - Y después hacemos que deje de sonar con noTone().

1	<pre>int buzzer = 9;</pre>
2	
3	<pre>void setup() {</pre>
4	<pre>Serial.begin(9600);</pre>
5	<pre>pinMode(buzzer,OUTPUT);</pre>
6	}
7	
8	<pre>void loop() {</pre>
9	<pre>tone(buzzer, 311.13);</pre>
10	delay(480);
11	noTone (buzzer);
12	delay(50);
13	
14	tone (buzzer, 311.13);
15	delay(260);
16	<pre>noTone (buzzer) ;</pre>
17	delay(50);
18	
19	<pre>tone(buzzer,261.63);</pre>
20	delay(250);
21	noTone (buzzer);
22	delay(50);
23	
24	}







Se haría algo parecido a como se explica en este código.









- ★ En el código indicaremos estas instrucciones donde ponemos la condición de sonarAlarma (la melodía de nuestro despertador).
- ★ Tendremos que inventarnos esa melodía, e indicar en el código cuando debería de sonar ;)









Las funciones son **agrupaciones de código**

- Ya conocemos las funciones
 - void setup(){ ... }
 - void loop() { ... }
- Cuando delante del nombre de la función aparece void significa que la función no devuelve nada.
- Las funciones pueden devolver variables.
- Delante del nombre de la función hay que añadir el tipo de variable que devuelve.
- Esta función es void porque no devuelve nada, ejecuta la música solo.



Esto hay que hacerlo en el scketch donde vamos a crear el despertador. Las funciones se añaden debajo del setup







Imaginemos que el código de abajo es el de despierta...para decir que debe sonar la alarma, la función despierta que hemos creado, la tenemos que poner aquí.

si es_la_hora, entonces

función despierta

ponemos la variable es_la_hora a false

1

si hora_alarma es igual hora_actual y minuto_alarm es igual minuto_actual, entonces ponemos es_la_hora igual a true encendemos los led con el color que queramos los enseñamos esperamos 500ms limpiamos la configuración lo enseñamos sonar alarma









Autoras: Elvira Castillo Twitter: @layoel Instagram: eca_ingeniera

Aprenderemos a

- Conectar el sensor de color.
- Programar la detección de colores.





Montaje del *Wake Me VP!*









- Usaremos ahora el sensor de color RGB (*TCS34725*) para cambiar el color de los leds:
- ★ Aunque no vamos a entrar a explicar cómo funciona la comunicación I2C (inter integrated circuits), debemos saber que se trata de un protocolo basado en la existencia de dos líneas de comunicación, una de ellas lleva la señal de reloj y es conocida como (SCL), y la otra línea lleva los datos y es conocida como (SDA).
- ★ Los Pines SDA y SLC se encuentran especificados en todos los componentes que usan este tipo de protocolo de comunicación, como podemos ver si volvemos a la primera foto de arriba.
- ★ Por tanto, para hacer funcionar nuestro sensor, alimentamos nuestro sensor de color desde Arduino mediante los pines G y 5V y conectamos el pin SDA y SCL del sensor a los 2 pines de nuestro arduino nano SCL y SDA.





★





Ahora que sabemos donde conectarlo, seguiremos el esquema siguiente:







Programando el WAKE ME VP!









1. Incluir en el IDE la librería Adafruit_TCS34725

Vamos al enlace y descargamos la librería haciendo clic en download









 Incluir en el IDE la librería Adafruit_TCS34725 recuerda cómo lo hicimos con la librería de Adafuit_Neopixel para los leds.

🥯 sketch_jun23a	Arduino 1.8.19			
Archivo Editar Pro	ograma Herramientas Ayuda			
Sketch_jun	Verificar/Compilar Subir Subir Usando Programador Exportar Binarios compilados	Ctrl+R Ctrl+U Ctrl+Mayús+U Ctrl+Alt+S		
2 11	Mostrar Carpeta de Programa	Ctrl+K	5	run once.
2 //	Incluir Librería	>		Administrar Bibliotecas Ctrl+Mawús+L
3	Añadir fichero			Administrar bibliotecas Ctr+Mayus+1
4 }				Añadir biblioteca .ZIP
5 6ªvoid l				Arduino bibliotecas Bridge







1.	Incluir en el IDE	tch_feb22	a Arduino IDE 2.3.4	24725 recuerdo cómo lo hicimos con la
	Archive	Editar	Sketch Herramientas Ayuda	
	libreria de Adat 🕢		Verificar/Compilar Ctrl + R	
			Cargar Ctrl + U	
		sketch	Configura y Subir	Gestionar hibliotecas Ctrl + Mawis + I
		12	Cargar usando el programador Ctrl + Mayús + U	
	(二)	13	Exportar binario compilado Alt + Ctrl + S	Añadir biblioteca .ZIP
		14	Optimizar para depuración	Bibliotecas de Arduino
	17h	16	Manters and the Chatch Althe Chat a K	Arduino_BuiltIn
		17	Mostrar carpeta de Sketch Ait + Ctri + K	EEPROM
	0	18	Incluir biblioteca	Ethernet
	0	20	Añadir fichero	Firmata
		21	_ , 0, , , , , , , , , , , , , , , , , ,	HID
	Q	22	<pre>tcs.getRawData(&r, &g, &b, &c);</pre>	
		23	<pre>colorTemp = tcs.calculateColorTemper lum</pre>	keyboard
		24	<pre>iux = tcs.calculateLux(r, g, b);</pre>	LiquidCrystal
		Salida		Mouse
		Libr	ería instalada	SD
				Servo
				SoftwareSerial
				SPI
				Stepper
	8			TFT
				Wire n.
			- Auto dete bulge	Bibliotecas aportadas
	nduct.r	nd	nicer readme and tidy	Adafruit NeoPixel
				Adafniit TCS2/725
	erties		Bump version number	Auditur (C334723







Hay que instalar otra librería más ya que la librería del sensor utiliza I2C y es la librería

Adafruit BuslO

🔤 sketcl	h_feb22a	Arduino IDE 2.3.4			
Archivo	Editar	Sketch Herramientas Ayuda			
	€	Verificar/Compilar	Ctrl + R		
		Cargar	Ctrl + U		
	GEST	Configura y Subir			Wire h
	DUSIC	Cargar usando el programador	Ctrl + Mayús + U		WIT COMP
1_)	Tipo:	Exportar binario compilado	Alt + Ctrl + S		:Adafruit_TCS34725.h>
	Tema	Optimizar para depuración			ise with default values (int tim
山	Adaf	Mostrar carpeta de Sketch	Alt + Ctrl + K		TCS34725 sensor color = Adafruit
	Adafr	Incluir biblioteca		۲	Gestionar bibliotecas Ctrl + Mayús + I
0	This is away	Añadir fichero			Añadir biblioteca .ZIP







Al incluirla en nuestro scketch veremos que se añaden otras librerías más. Las necesita arduino para saber usar los sensores I2C



	⇒ 🚯	🖞 Arduino Nano 👻
	sketch_f	eb22a.ino
	1	<pre>#include <adafruit_busio_register.h;< pre=""></adafruit_busio_register.h;<></pre>
	2	<pre>#include <adafruit_genericdevice.h></adafruit_genericdevice.h></pre>
1_)	3	<pre>#include <adafruit_i2cdevice.h></adafruit_i2cdevice.h></pre>
	4	<pre>#include <adafruit_i2cregister.h></adafruit_i2cregister.h></pre>
nfla	5	<pre>#include <adafruit_spidevice.h></adafruit_spidevice.h></pre>
	6	
	7	<pre>#include <wire.h></wire.h></pre>
0	8	
	9	<pre>#include <adafruit_tcs34725.h></adafruit_tcs34725.h></pre>
	10	







- □ Vamos a intentar detectar algunos colores con nuestro sensor.
- Comenzaremos con el Rojo, Verde y Azul.
- Después añadiremos alguno más.

Mi programa es colores. Funciona y cambia los colores del led