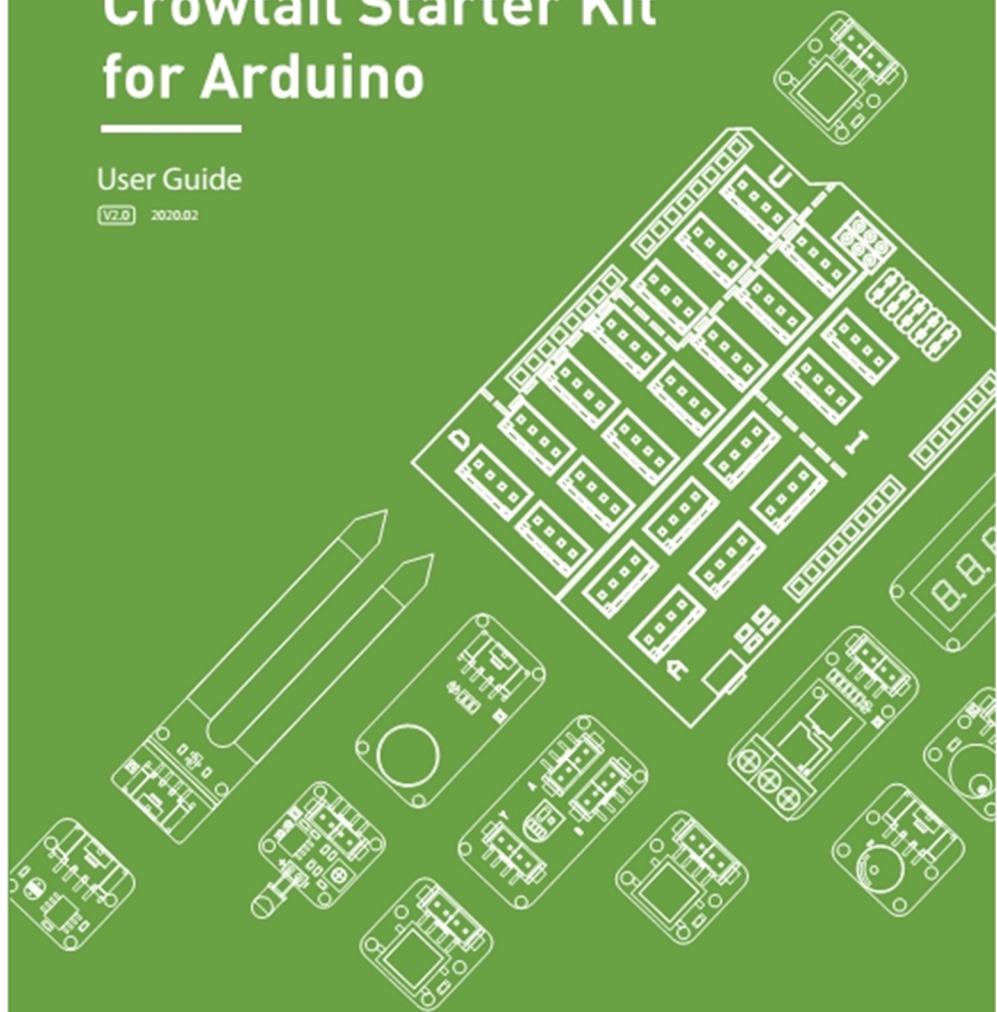


Crowtail Starter Kit for Arduino

User Guide

V2.0 2020.02



Contenido

Instrucciones	04
Crowduino Uno	05
Crowtail	05
Listado de módulos	07
Empezando	08
• Instalando el IDE	11
• Cómo cargar un programa	11
• Añadir librerías y abrir el Monitor Serie	12
Lecciones	14
• Lección 1 – Control de un LED	14
• Lección 2 – Control de un LED con un pulsador	15
• Lección 3 – Control de un LED con un interruptor	17
• Lección 4 – Sensor al tacto	18
• Lección 5 – Alarma por movimiento	20
• Lección 6 – Alarma de incendios	21
• Lección 7 – Regulación LED	23
• Lección 8 – Monitor de humedad	25
• Lección 9 – Sensor de sonido	26
• Lección 10 – Alumbrado inteligente	28
• Lección 11 – Alarma por calentamiento	29
• Lección 12 – Función lógica NOT	31
• Lección 13 – Función lógica OR	32
• Lección 14 – Función lógica AND	34
• Lección 15 – Sistema automático de control	35
• Lección 16 – Un número sobre display de 4 dígitos	37
• Lección 17 – Semáforo	39
• Lección 18 – Luz inteligente de pasillo	40
• Lección 19 – Sistema de riego	42
• Lección 20 – Ajustar brillo con display	44

Bienvenido a la guía de usuario del Crowtail Starter Kit para Arduino. Vamos a empezar con algunos de los sensores básicos y la tarjeta Crowduino 100% compatible con Arduino. El kit contiene 20 ejemplos interesantes, organizados de menor a mayor complejidad y explicados paso a paso para familiarizarnos con los módulos electrónicos, ejercitar el pensamiento lógico y desarrollar nuestras ideas a través de la programación. Desde la instalación de Arduino hasta la explicación de cada módulo electrónico y su programación, cada paso es tan sencillo que, aun siendo principiante, rápidamente podrás adquirir los conceptos básicos de la programación de Arduino.

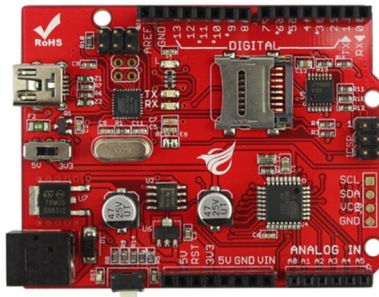
El Crowtail Starter Kit para Arduino incluye 22 módulos electrónicos con sus características y funciones concretas. Cada módulo se ha seleccionado para todos los que empiezan con Arduino, como son los módulos con funciones lógicas pensados para introducirnos en el pensamiento de lógico de la programación, o el módulo sensor de luz que nos muestra métodos para el control de iluminación. En resumen, con este kit se mostrarán los conceptos básicos y empleo de sensores, captura de señales digitales y analógicas, conversión analógico/digital, lógica de programación y algunos módulos electrónicos algo más complejos. Lo más importante de todo es apreciar la “*magia*” de la electrónica y divertirse programando en Arduino tus propias ideas y proyectos.

En lo que ha programación se refiere usaremos el entorno IDE de Arduino. Se trata de una plataforma de código abierto, fácil de usar y de las más populares del mundo, incluyendo diferentes modelos de tarjetas controladoras. Es la mejor opción para personas que desean aprender a programar y adquirir conocimientos de hardware.

Crowduino UNO

La tarjeta de control Crowduino UNO-SD es la tarjeta microcontroladora 100% compatible con Arduino UNO. Está basada en el chip Atmega328P, el mismo que se emplea en Arduino UNO original y en otras tarjetas compatibles.

La tarjeta UNO-SD tiene 14 patillas de entrada/salida digital (6 de las cuales pueden usarse como salidas PWM), 6 entradas analógicas, un resonador cerámico de 16MHz, conector USB, jack de alimentación, conector ICSP, botón de reset y un cable mini USB para conectar con el ordenador. Se puede alimentar desde el propio puerto USB, mediante un adaptador AC/DC o bien mediante una batería externa.



Comparada con la Arduino UNO original, la tarjeta Crowduino UNO presenta las siguientes mejoras:

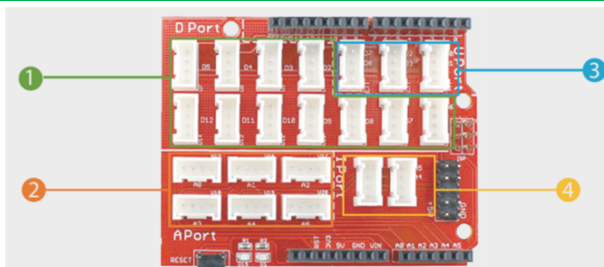
1. El botón Reset se encuentra en un borde de la tarjeta, lo que facilita resetear el sistema.
2. Circuito mejorado de alimentación. Mediante un conmutador se puede seleccionar el trabajar a 3.3V o a 5V, lo que hace la tarjeta UNO-SD sea compatible con más sensores que usan niveles lógicos con tecnología de 3.3V
3. Emplea un conector mini USB
4. Incluye zócalo para tarjetas SD de memoria. Crowduino UNO-SD es ideal para aplicaciones de captura de datos y monitorización de entornos.

Crowtail

Crowtail es una serie de módulos desarrollados para facilitar la conexión de circuitos electrónicos. Parte de una tarjeta o base shield y varios módulos con sensores y actuadores que facilitan el montaje de circuitos de forma rápida y sencilla

Crowtail – Base Shield

El Crowtail-Base Shield es una tarjeta estándar para la expansión de las patillas de E/S de Arduino. Emplea conectores de 4 vías normalizados para conectar en ellos cualquiera de los módulos Crowtail. Están agrupados en: Analógicos (A), Digitales (D), Comunicaciones UART (U) y Comunicaciones I2C (I).

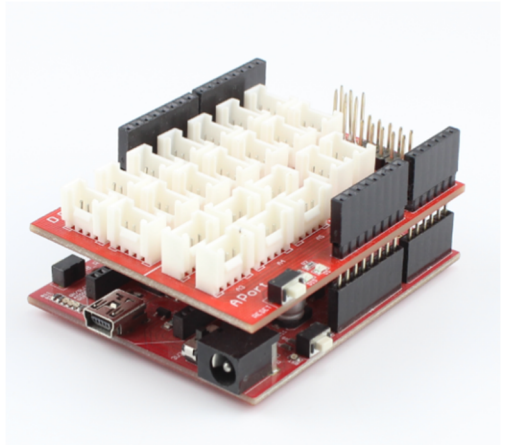
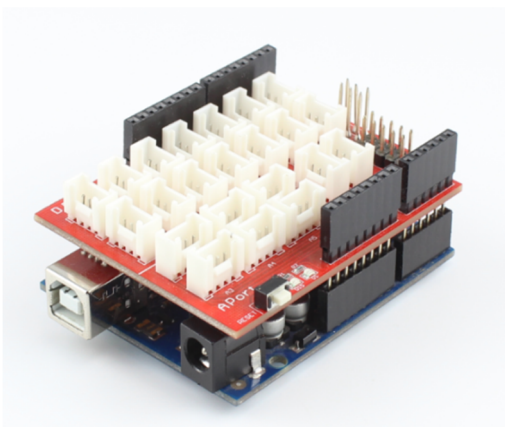


1. Conectores de acceso a 11 patillas digitales (D2-D12). Se usan para leer y controlar periféricos digitales como los de los módulos Crowtail tipo LED, Pulsador, Interruptor, etc...Algunas de estas patillas se usan como salidas PWM (modulación de anchura del pulso).
2. Acceso a las 6 entradas analógicas (A0-A5) del controlador. En ellas podemos conectar módulos Crowtail analógicos como son el sensor de luz o el potenciómetro.
3. Conectores que acceden a las parejas de señales D0/D1, D2/D3 y D7/D8. Especialmente pensados para conectar módulos Crowtail que se comunican con el controlador vía serie tipo UART (Wifi, Bluetooth, etc...)
4. Dos conectores que dan acceso a las señales A4/A5 del Arduino y que también se usan en comunicaciones I2C. Se pueden conectar simultáneamente dos módulos Crowtail que empleen este protocolo.

También disponen de acceso a la alimentación de 5V y GND para uso de propósito general. La idea es que los usuarios puedan conectar, mediante unos sencillos cables, cualquier módulo de forma rápida y segura

En comparación con otros sistemas para realizar montajes electrónicos, Crowtail permite conectar múltiples periféricos de forma rápida y segura. El usuario se centra solo en programar ya que los módulos son totalmente funcionales.

Conexión de la tarjeta de adaptación Base Shield con Arduino UNO original y con nuestra controladora Crowduino UNO.



Módulos Crowtail

Son una serie de circuitos electrónicos formados por sensores, displays, módulos de E/S, analógicos, digitales, de comunicación I2C y UART, etc.... Todos los módulos tienen un conector común y se conectan, mediante unos cables incluidos en el kit sobre la tarjeta Base Shield, y esta a su vez sobre la controladora Crowduino, Hacemos las conexiones de forma rápida y segura.

Listado de los módulos incluidos en el kit

- 1 x Controladora Crowtail UNO-SD
- 1 x Crowtail Base shield
- 1 x Módulo Crowtail con un pulsador
- 1 x Módulo Crowtail con un interruptor
- 1 x Módulo Crowtail con un Zumbador
- 1 x Módulo Crowtail con un LED (rojo)
- 1 x Módulo Crowtail con un LED (Verde)
- 1 x Módulo Crowtail con un LED (amarillo)
- 1 x Módulo Crowtail con una función lógica AND
- 1 x Módulo Crowtail con una función lógica OR
- 1 x Módulo Crowtail con una función lógica NOT
- 1 x Módulo Crowtail con un sensor de llama
- 1 x Módulo Crowtail con un sensor de humedad
- 1 x Módulo Crowtail con un sensor táctil
- 1 x Módulo Crowtail con un relé
- 1 x Módulo Crowtail con un sensor de luz
- 1 x Módulo Crowtail con un sensor de temperatura
- 1 x Módulo Crowtail con un motor de vibración
- 1 x Módulo Crowtail con un Potenciómetro lineal
- 1 x Módulo Crowtail con un sensor de movimiento
- 1 x Módulo Crowtail con un sensor de sonido
- 1 x Módulo Crowtail con un display de 4 dígitos

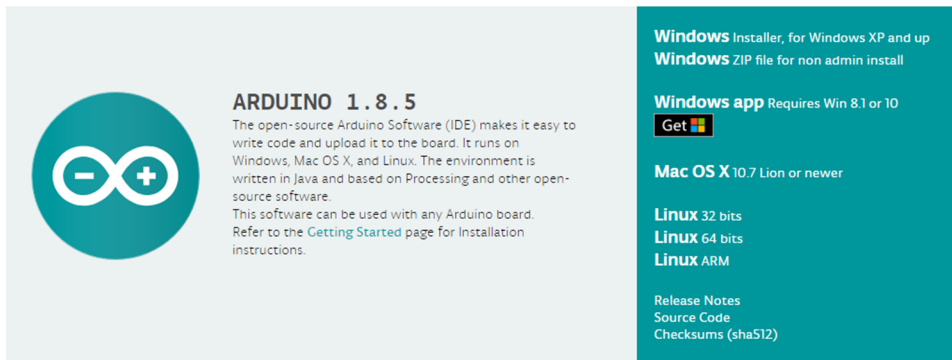
Vamos a empezar

Instalando el IDE

Introducción

“Arduino” no es solo el nombre de la tarjeta controladora, sino también el nombre del entorno de programación conocido como IDE y desarrollado en C/C++. Una vez que tenemos una tarjeta Aduino, o compatible como Crowduino, debemos instalar este IDE según la versión del Sistema Operativo que tengamos. Está disponible para Windows, Mac y Linux. Vamos a ver cómo instalar el IDE en nuestro ordenador.

PASO 1: Ve a <https://www.arduino.cc/en/Main/Software> y busca la zona de descarga.



ARDUINO 1.8.5
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
[Get](#)

Mac OS X 10.7 Lion or newer

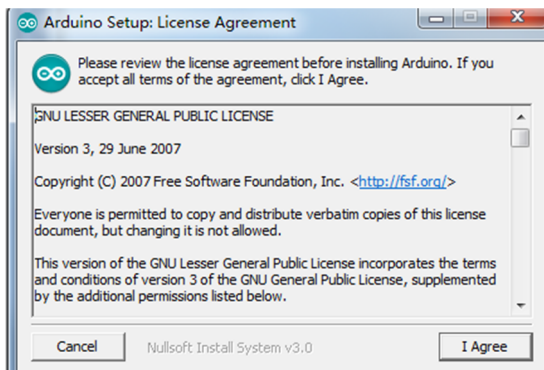
Linux 32 bits
Linux 64 bits
Linux ARM

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

En este sitio web encontrarás la versión más actual del IDE. No coincidirá con la mostrada en la imagen.

PASO 2: Descargar el software de acuerdo con tu Sistema Operativo, por ejemplo Windows.

- Selecciona “**Windows Installer**” → “**Download**”
- Pulsar “**I Agree**”, “**Next**” y luego “**Browse**” para elegir la ruta o directorio donde realizar la instalación.



- Pulsar **“Install”** para iniciar la instalación como cualquier otra aplicación Windows.



Una vez finalizado, el icono de Arduino aparecerá en el escritorio.

Conectar Arduino/Crowduino al PC

Conectar la tarjeta controladora Arduino/Crowduino al ordenador usando el cable USB.

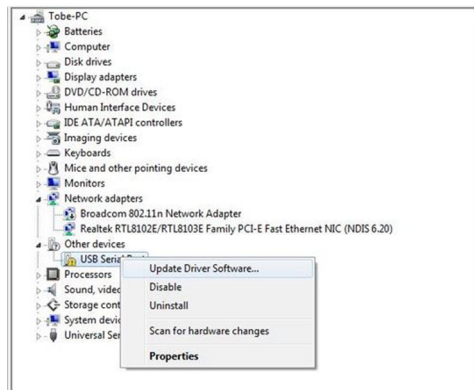
Instalar los Driver

Instalando drivers en Windows para Crowduino

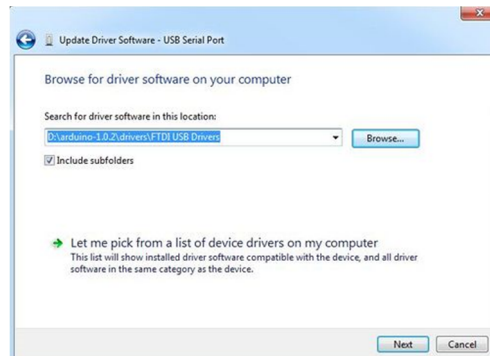
Al conectar Crowduino Windows tratará de instalar el controlador automáticamente en aproximadamente 1 minuto. Si no fuera así sigue los siguientes pasos

PASO 1: Abre el administrador de dispositivos. Mira en Puertos COM y LPT. Debe aparecer un puerto **“Puerto serie USB”**.

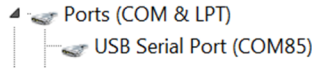
PASO 2: Con el botón derecho selecciona **“Actualizar software del controlador”**



PASO 3: Navega por el explorador en busca del archivo **“FTDI USB Driver”** situado en la carpeta **“Drivers”** del Arduino IDE.



Pulsa “**Siguiente**”. Una vez finalizada la instalación vuelve al administrador de dispositivos y comprueba que aparece un nuevo puerto serial.



Instalando Arduino (Linux)

Usa el comando “**Make install**”. Si estás empleando Ubuntu, se recomienda instalar Arduino IDE desde el centro de software de Ubuntu.

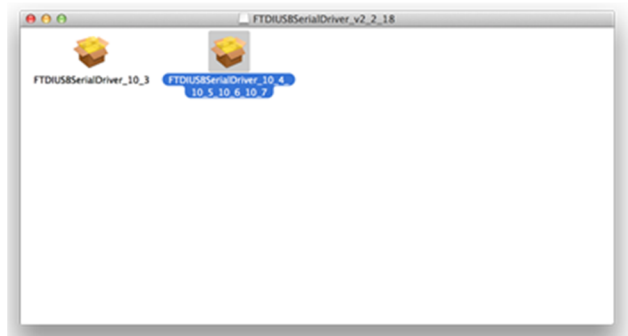
Instalando Arduino (MAC OS X)

Descarga y descomprime el archivo ZIP. Haz click en “Arduino.app” para comenzar con la instalación. Si lo pide, deberás instalar la biblioteca en tiempo de ejecución de Java. Una vez finaliza la ejecución ya puedes comenzar a trabajar con el IDE de Arduino.

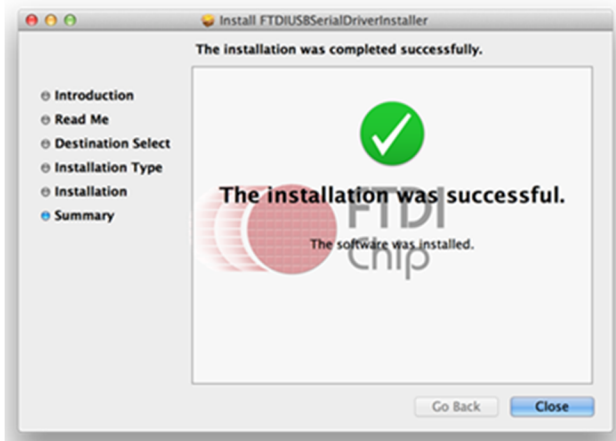
Instalar los Driver de Crowduino para Mac OS

Desde:

<http://www.ftdichip.com/Drivers/VCP.htm> descarga la versión correcta para tu ordenador de los drivers para Mac OS X. Abre el archivo recién descargado, **FTDIUSBSerialDriver_10_4_10_5_10_6_10_7.mpkg**, haciendo doble click y seleccionando la opción “**Continue**”.



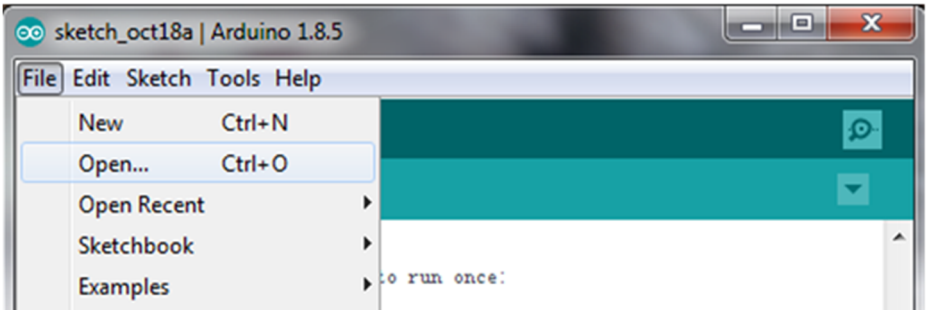
Si todo va correctamente deberás ver una ventana como la de la figura.



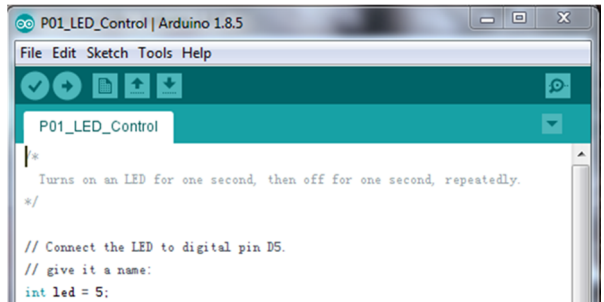
Cómo cargar un programa

Para cargar/grabar un programa en la memoria de Arduino seguimos los siguientes pasos:

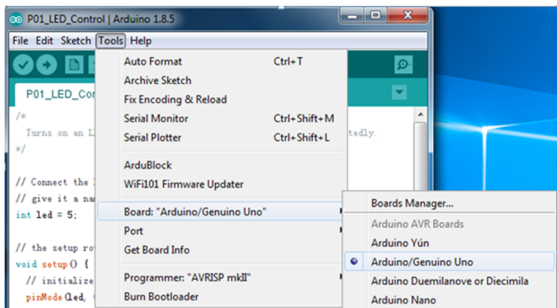
PASO 1: Ejecutar el IDE de Arduino y seleccionar "File" → "Open"



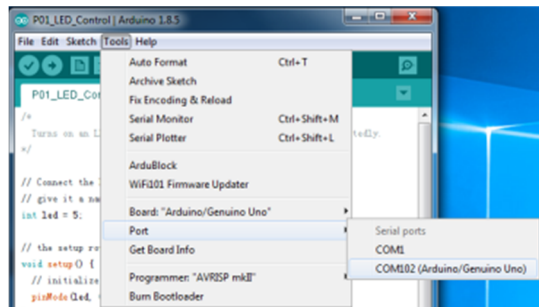
PASO 2: Buscar el directorio donde se encuentra el programa o script a cargar y abrirlo. Deberás ver el contenido de ese programa o script.



PASO 3: Mediante "Select Tools" → "Board" → Arduino/Genuino UNO seleccionamos el modelo de la tarjeta controladora (Crowduino es 100% compatible con Arduino)

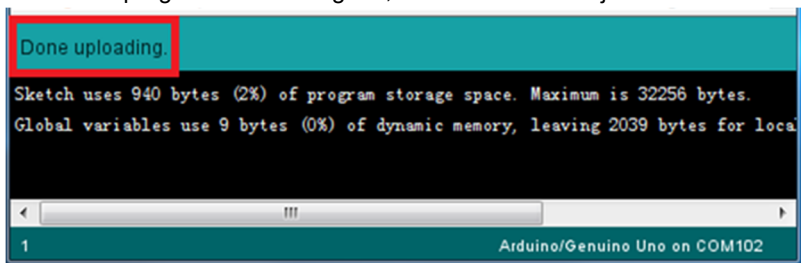


PASO 4: Seleccionamos el puerto de comunicaciones a emplear mediante "Select Tools" → Port → COMxx (xx es el canal COM que Windows asigne en cada caso).



PASO 5: Hacer click en el icono de “Grabar”  para grabar el programa.

PASO 6: Una vez el programa se ha cargado, recibirás un mensaje similar a este.



Añadir librerías y abrir el monitor serie

¿Qué son las librerías?

Las librerías son una colección de programas y funciones que facilitan el control de un sensor, pantalla, etc..., liberándote de tener que hacerlo tú. Hay cientos de librerías disponibles en Internet. Para usar una librería primero hay que instalarla. ¿Cómo se hace? Hay dos formas, una mediante el gestor de librerías del propio IDE y otra importando la librería en formato ZIP.

Importando una librería .ZIP

Algunas librerías se ofrecen en forma de un archivo o carpeta *.ZIP, cuyo nombre es el mismo que el de la librería. Dentro de la carpeta encontrarás archivos *.CPP, *.h, keywords.txt, carpeta con ejemplos y quizá otros archivos auxiliares.

Proceso

- Descomprimir el archivo *.ZIP y copiar la carpeta recién extraída en la carpeta C:\...\Arduino\Library

Nota: Los ejemplos no se verán en Archivo → Ejemplo del IDE hasta que éste se reinicie.

Usando el gestor de librerías

Para instalar una librería se puede usar el propio gestor incluido en el IDE de Arduino.

Proceso:

- Programa → Incluir librería → Administrar Bibliotecas
- En el administrador verás el listado de librerías instaladas o listas para instalar. Localiza y selecciona la que desees, así como su versión.
- Finalmente pulsa “Instalar”



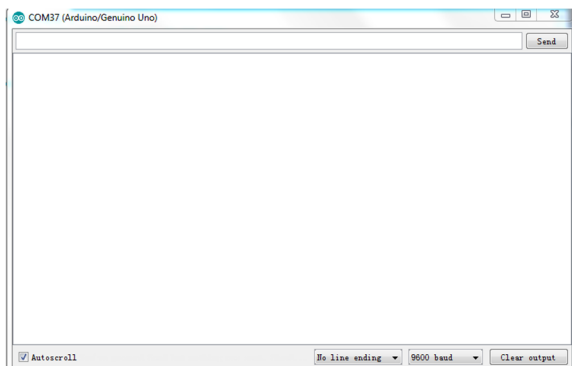
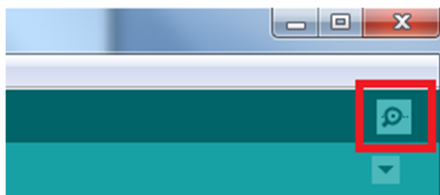
Monitor Serie de Arduino (Windows, Mac, Linux)

El IDE de Arduino es la herramienta software con la que vas a trabajar para desarrollar tus propios programas e incluye un terminal de comunicaciones o monitos serie.

El monitor serie es una de las herramientas más usadas del IDE. Te servirá para ver en pantalla los resultados que genera tu programa, o bien introducir mediante el teclado los datos que deben procesarse. Es una gran herramienta para la depuración y puesta a punto de las aplicaciones.

Cómo usar el monitor serie

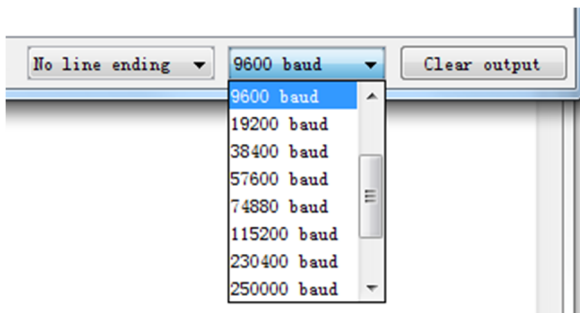
- Herramientas → Puerto, Seleccionar el puerto COM que haya asignado Windows. Es el mismo que el que aparece en el administrador de dispositivos.
- Hacer click sobre el icono del monitor serie.



- Verás la ventana propia del monitor serie.

Ajustes

Las posibilidades de configuración del monitor serie son limitadas pero suficientes para cubrir la mayor parte de necesidades. Una de ellas es el ajuste de la velocidad en baudios. Basta con seleccionar el valor apropiado.



También puedes activar o no el desplazamiento, Autoscroll, de la pantalla, así como seleccionar la forma de transmitir una línea: con retorno, avance, ambos o ninguno.

Lecciones

Lección 1 – Control de un led

Introducción

El LED es el dispositivo más sencillo para manejar las patillas de E/S. Conectaremos un LED a uno de los puertos digitales del Base Shield y luego grabaremos el programa en Arduino. Su empleo es muy sencillo. Podemos hacer que el LED parpadee a la frecuencia deseada y por tanto regular su brillo. En este Kt dispones de LED de colores rojo, verde y amarillo. En esta primera lección haremos una sencilla intermitencia.



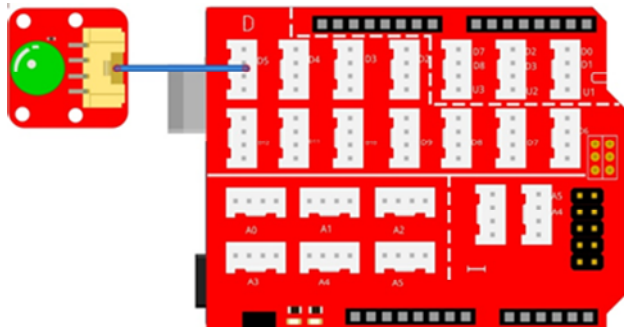
Necesitamos...

- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x LED
- 1 x Crowtail Cable
- 1 x Cable USB

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el módulo Crowtail LED en el puerto D5 como puedes ver en la figura. Abrir el programa de ejemplo [P01_Control_de_un_led.ino](#) con el IDE de Arduino y grabarlo sobre la tarjeta controladora.



Qué vas a ver

El LED se iluminará durante 1 segundo, se apagará durante otro segundo y el ciclo se repetirá indefinidamente. Si no fuera así revisa la conexión del módulo LED y la tarjeta Base Shield.

Descripción del programa

```
1 int led = 5;
2 // Turns on an LED for one second, then off for one second, repeatedly.
3 /*
4
5 // Connect the LED to digital pin D5.
6 // give it a name:
7 int led = 5;
8
9 // the setup routine runs once when you press reset:
10 void setup() {
11 // initialize the digital pin as an output.
12 pinMode(led, OUTPUT);
13 }
14
15 // the loop routine runs over and over again forever:
16 void loop() {
17 digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
18 delay(1000); // wait for a second
19 digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
20 delay(1000); // wait for a second
21 }
```

Comments: //This is a comment
Comments are a great way to leave notes in your code explaining why you wrote it the way you did. You'll find many comments in the examples that further explain what the code is doing and why. Comments can be single line using //, or they can be multi-line using /* */.

Integer Variables
A variable is a placeholder for a value that may change in your code. Variables must be introduced or "declared" before using variables. Here, we declare a variable called "led" of type int(integer) and assign it a value of 5, that is, led corresponds to pin 5. Don't forget that variable names are case sensitive!

code to run once
The code inserted between the setup() braces will run once.

Input or Output
Before using one of the digital pins, you need to tell Arduino whether it is an input (INPUT) or an output (OUTPUT). We use a built-in "function" called pinMode() to make the pin corresponding to the led a digital output.

code to run forever
The code inserted between the loop() braces will run repeatedly until the Arduino resets or powers off.

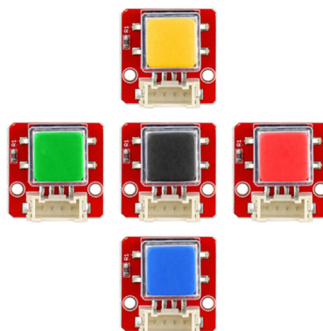
Digital Output
When you're using a pin as an OUTPUT, you can command it to be HIGH (output 5 volts) or LOW (output 0 volts).

Delay
Causes the program to wait on this line of code for the amount of time in between the brackets. After the time has passed, the program will continue to the next line of code.

Lección 2 – Control de un led con un pulsador

Introducción

Este pulsador produce una señal lógica de nivel ALTO cuando se pulsa y de nivel BAJO cuando se suelta. Estos niveles lógicos son detectados por Arduino y, según tu programa, hacer lo que quieras a partir de ellos. En esta lección usaremos el pulsador para encender y apagar el LED.



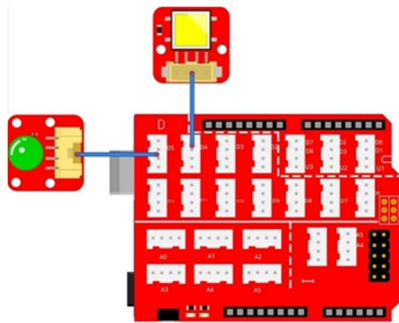
Necesitamos...

- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x LED
- 1 x Pulsador
- 2 x Crowtail Cable
- 1 x Cable USB

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el módulo Pulsador en el puerto D4 y el LED en el puerto D5 como puedes ver en la figura. Abrir el programa de ejemplo **P02_Control_de_un_led_con_un_pulsador.ino** con el IDE de Arduino y grabarlo sobre la tarjeta controladora.



Qué vas a ver

Cuando accionas el pulsador el LED se enciende y se mantiene encendido mientras el pulsador esté accionado. Al dejar de pulsar el LED se apaga.

Descripción del programa

Variables enteras: `int buttonState=0;`

Una variable almacena datos que cambian durante la ejecución del programa. Deben declararse antes de ser usadas. En el ejemplo se declara la variable “buttonState” de tipo entero y le asigna el valor inicial 0 que representa el estado del pulsador.

Digital Input: `buttonState=digitalRead(buttonpin);`

La función `digitalRead()` lee el valor de una entrada digital. Comprueba si esa entrada está a nivel alto-HIGH (5V) o bajo-LOW (0V). Devuelve TRUE (1) o FALSE (0) según ese estado.

Sentencia if/else: `if(condición) {si la condición es cierta se ejecuta este código} else {si la condición es falsa se ejecuta este código}`

La sentencia `if/else` ejecuta el código cerrado entre llaves si se cumple la condición. En caso contrario, `else`, se ejecuta el otro código cerrado también entre llaves. En el ejemplo, si el pulsador está activado el LED se enciende en caso contrario el LED se apaga.

Is equal to: `buttonState == HIGH`

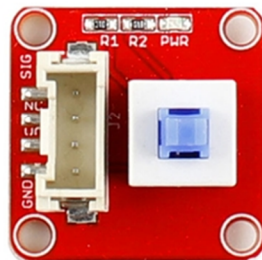
Operador lógico de igualdad (`==`). Determina si ambos valores son iguales. No confundir con el signo `'=`' que es un operador de asignación y no de comparación.

Lección 3 – Control de un led con un interruptor

Introducción

El interruptor es un dispositivo con enclavamiento. Cuando se acciona por vez primera produce una señal alta, HIGH, que se mantiene hasta que se vuelve a accionar por segunda vez. En ese momento la señal pasa a bajo, LOW. Es muy similar al pulsador excepto que el interruptor tiene un mecanismo de bloqueo que mantiene la señal sin necesidad de estar accionándolo constantemente.

Dado que el pulsador y el interruptor son muy similares, en este ejemplo usaremos el interruptor para que veas la diferencia.



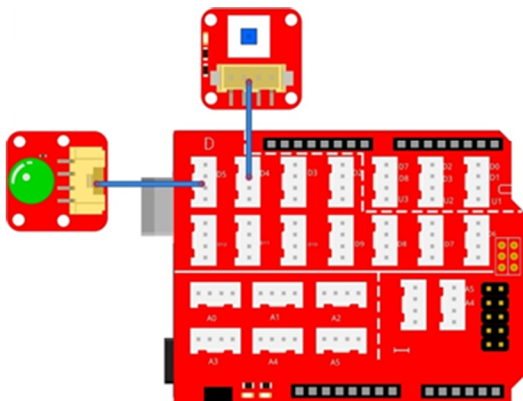
Necesitamos...

- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x LED
- 1 x Interruptor
- 2 x Crowtail Cable
- 1 x Cable USB

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el módulo Interruptor en el puerto D4 y el LED en el puerto D5 como puedes ver en la figura. Abrir el programa de ejemplo [P03_Control_de_un_led_con_un_interruptor.ino](#) con el IDE de Arduino y grabarlo sobre la tarjeta controladora.



Qué vas a ver

Al presionar el interruptor éste se enclava y el LED se enciende. Cuando lo accionamos nuevamente, el interruptor se desenclava y el LED se apaga. Si no fuera así revisa las conexiones del LED y el interruptor.

Descripción del programa

Variables enteras: `int switchPin=4; int ledPin=5;`

Una variable almacena datos que cambian durante la ejecución del programa. Deben declararse antes de ser usadas. En el ejemplo se declaran las variables “ledPin” y “switchPin” de tipo entero y se les asigna el valor 4 y 5, correspondientes a las patillas de entrada y salida respectivamente en las que los vamos a conectar.

Entrada o Salida: `pinMode(ledPin,OUTPUT); pinMode(switchPin, INPUT);`

Antes de usar cualquiera de las patillas digitales, se deben configurar quienes son entradas y quienes salidas. Esto lo hacemos con la función `pinMode()`. A la patilla del LED se le configura como salida y a la del interruptor como entrada.

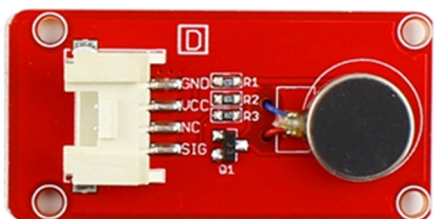
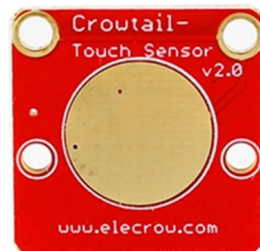
Loop: `void loop() {Código a ejecutar}`

En la función `loop()` se emplea `digitalRead()` para leer el estado del interruptor. Mediante la sentencia `if/else` se determina si está activado (HIGH) o no (LOW). Si está activado se enciende el LED y en caso contrario el LED se apaga.

Lección 4 – Sensor al tacto

Introducción

El sensor táctil, basado en el CI TTP223-B, detecta un toque por cambio de capacidad. Cuando esto ocurre se genera un nivel alto (HIGH). El módulo puede detectar la proximidad del dedo a unos 0-3 mm, sin llegar a tocarlo. Esto te permite colocarlo bajo una superficie no metálica como el vidrio o el papel, lo que sería idóneo en aplicaciones a la intemperie o con sensores ocultos.



Un vibrador está compuesto de un motor de CC sin núcleo de imán permanente que se activa cuando recibe una señal de nivel alto (HIGH). Según se controle puede producir una vibración más o menos notable y adecuada para aplicaciones como juguetería y alarmas. Por ejemplo, un invidente puede recibir una vibración a modo de señal de aviso.

Necesitamos...

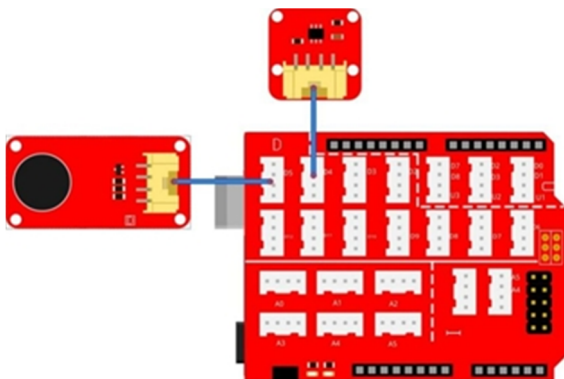
- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x sensor táctil
- 1 x vibrador
- 2 x Crowtail Cable
- 1 x Cable USB

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el sensor táctil en el puerto D4 y el motor de vibración en el puerto D5 como puedes ver en la figura.

Abrir el programa de ejemplo **P04_Sensor_al_tacto.ino** con el IDE de Arduino y grabarlo sobre la tarjeta controladora.



Qué vas a ver

En el momento de tocar el sensor táctil se produce un cambio de capacidad en el circuito y el vibrador se activará, Si no fuera así asegúrate que las conexiones estén bien realizadas y de que la distancia entre el dedo y el sensor es inferior a 3 mm.

Descripción del programa

Variables enteras: `int touchPin=4;`

Una variable almacena datos que se modifican con la ejecución del programa. Deben declararse antes de ser usadas. En el ejemplo se declara la variable “touchPin” de tipo entero y se le asigna el valor 4, correspondiente a la patilla de entrada.

Entrada o Salida: `pinMode(vibrationPin,OUTPUT); pinMode(touchPin, INPUT);`

Antes de usar cualquiera de las patillas digitales, se deben configurar quienes son entradas y quienes salidas. Esto lo hacemos con la función `pinMode()`. A la patilla del vibrador se le configura como salida y a la del sensor táctil como entrada.

Loop: void loop() {Código a ejecutar}

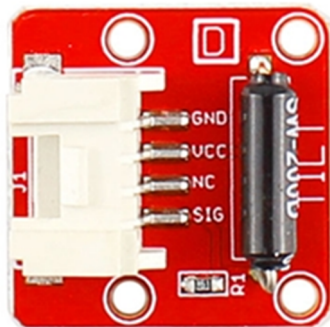
Puesto que el sensor táctil es un sensor digital, empleamos `digitalRead()` para leer el estado del mismo. Mediante la sentencia `if/else` se determina si lo estamos tocando (HIGH) o no (LOW). Si lo estamos tocando se activa el vibrador y en caso contrario se desactiva.

Lección 5 – Alarma por movimiento

Introducción

El sensor de movimiento es capaz de detectar su posición. Si se encuentra horizontal produce un nivel bajo (LOW) y nivel alto (HIGH) si se encuentra en posición vertical. En su interior hay dos bolas metálicas que hacen contacto con dos terminales cuando el sensor se pone verticalmente. En sentido horizontal no se produce contacto alguno.

¿Sabías que tu frigorífico no debe inclinarse? Esto afectaría al correcto funcionamiento. En esta lección vemos cómo detectar la inclinación.



Necesitamos...

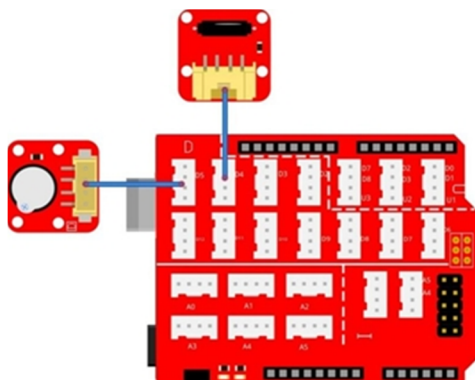
- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x sensor de movimiento
- 1 x zumbador
- 2 x Crowtail Cable
- 1 x Cable USB

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el sensor de movimiento en el puerto D4 y el zumbador en el puerto D5 como puedes ver en la figura.

Abrir el programa de ejemplo **P05_Alarma_por_movimiento.ino** con el IDE de Arduino y grabarlo sobre la tarjeta controladora.



Qué vas a ver

Cuando el sensor de movimiento se coloca en posición vertical se produce un nivel alto (HIGH) y el zumbador se activa. Cuando está en posición horizontal se produce un nivel bajo (LOW) y el zumbador se desactiva.

Descripción del programa

Variables enteras: `int tiltPin=4; int buzzerPin = 5; int tiltState = 0;`

Se declaran tres variables: `tiltPin`, `buzzerPin` y `tiltState` a las que se les asigna sus correspondientes valores. Esto permite conectar al sensor de movimiento en el pin D4, al zumbador en el pin 5 y poner el valor inicial del sensor a 0.

Entrada o Salida: `pinMode(buzzerPin,OUTPUT); pinMode(tiltPin, INPUT);`

En la función `setup()` se configura al sensor de movimiento como entrada y al zumbador como salida.

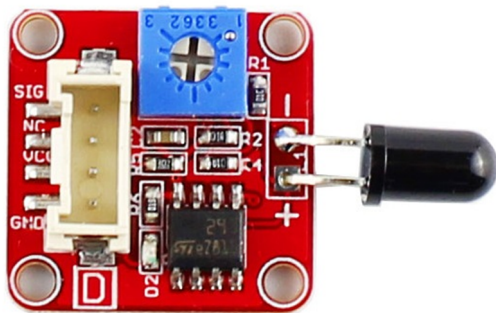
Sentencia if/else: `if(condición) {si la condición es cierta se ejecuta este código} else {si la condición es falsa se ejecuta este código}`

Mediante `if/else` se ejecuta el código necesario en función de si el sensor detecta inclinación, en cuyo caso se activa el zumbador, o no.

Lección 6 – Alarma de incendios

Introducción

El sensor de llama se puede usar para detectar incendios u otras fuentes de luz con una longitud de onda entre 760 y 110 nm. Consiste del fototransistor de silicio YG1006 de alta velocidad y sensibilidad a la radiación infra roja. En un sistema contra incendios este tipo de sensor juega un papel muy importante. El fuego representa una seria amenaza para la vida y la propiedad, por lo que la prevención de incendios es importante. En esta lección haremos una simulación de detección de incendio.



Necesitamos...

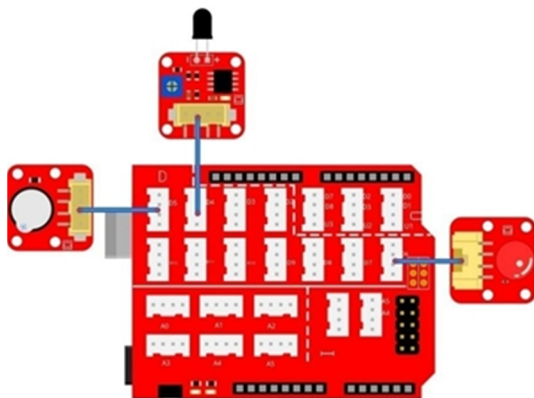
- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x sensore llama
- 1 x LED (rojo)
- 1 x zumbador
- 3 x Crowtail Cable
- 1 x Cable USB

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el sensor de llama, el zumbador y el LED en los pines D4, D5 y D6 respectivamente, como puedes ver en la figura.

Abrir el programa de ejemplo **P06_Alarma_de_incendios.ino** con el IDE de Arduino y grabarlo sobre la tarjeta controladora.



Qué vas a ver

Al acercarse una llama al sensor, el zumbador se activará y el LED producirá una intermitencia a modo de alarma. Cuando la llama se aleja la situación de alarma desaparece.

Descripción del programa

Variables enteras: `const int flamePin=4; const int buzzerPin = 5; const int redPin = 6; int flameState = 0;`

Inicialmente se declaran cuatro variables: `flamePin`, `buzzerPin`, `redPin` y `flameState` con sus correspondientes valores. En algunas de ellas se ha empleado el prefijo “const”, abreviatura de constante. En este caso esas variables solo se pueden leer y no se pueden modificar durante el programa. Es una buena práctica para declarar patillas o valores que no van a cambiar en el curso de la ejecución del programa. Resumiendo, se ha declarado que el sensor de llama se conecta con el pin D4, el zumbador con D5 y el led con D6. `flameState` se inicia con el calor 0

Entrada o Salida: `pinMode(flamePin, INPUT); pinMode(buzzerPin,OUTPUT); pinMode(redPin,OUTPUT);`

En la función setup() se configura al sensor de llama como entrada para detectar el fuego y al zumbador y al led como salidas que se activan y parpadean respectivamente cuando se detecta el incendio

Sentencia if/else: if(condición) {si la condición es cierta se ejecuta este código} else {si la condición es falsa se ejecuta este código}

Mediante if/else se ejecuta el código necesario en función de si el sensor detecta fuego, en cuyo caso se activa el zumbador y parpadea el led, o no hay fuego.

Lección 7 – Regulación LED

Introducción

El módulo potenciómetro emplea una resistencia variable de hasta 10 K Ω . Cuando se mueve el mando de un extremo a otro, el voltaje o tensión de salida varía de 0V a VCC. En este caso vamos a usar este potenciómetro lineal para controlar el brillo de un LED.



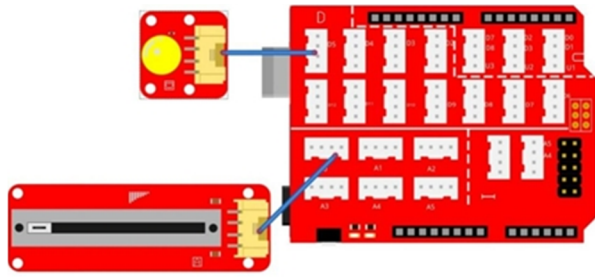
Necesitamos...

- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x Potenciómetro lineal
- 1 x LED (amarillo)
- 2 x Crowtail Cable
- 1 x Cable USB

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el módulo potenciómetro y el LED en los puertos de Crowtail Base Shield marcados como A0 y D5 respectivamente, como puedes ver en la figura.



Abrir el programa de ejemplo **P07_Regulacion_LED.ino** con el IDE de Arduino y grabarlo sobre la tarjeta controladora.

Qué vas a ver

AL deslizar el mando regularemos el brillo del LED. Si se mueve a la izquierda, hacia el conector, el brillo disminuye y al moverlo a la derecha el brillo aumenta.

Descripción del programa

Variables enteras: `int ledPin = 5; int pmeterPin = A0; int pmeterValue = 0;`

Lo primero siempre es declarar los pines y asignar valores. Lo nuevo es que en esta ocasión usaremos una entrada analógica para el potenciómetro. En esta lección declaramos el ledPin en D5 y pmeterPin en A0. El valor inicial de pmeterValue es 0.

Entrada o Salida: `pinMode(pmeterPin, INPUT); pinMode(ledPin, OUTPUT);`

En la función setup() configuramos el potenciómetro como entrada para leer su valor, y el pin LED como salida sobre la que escribiremos el valor leído desde el potenciómetro.

Entrada analógica: `pmeterValue = analogRead(pmeterPin);`

A diferencia de los pines digitales, debemos usar la función analogRead() para leer el valor de la entrada analógica que indiquemos, en este caso A0. Devuelve un número entre 0, si la entrada está a 0V, y 1023 si la entrada está 5V. Dicho valor se asigna a la variable pmeterValue.

Función matemática de Arduino map(): `int gapValue = map(valor, valor mínimo inicial, valor máximo inicial, valor mínimo final, valor máximo final)`

Pasa números de un rango a otro de forma proporcional. Un valor inicial comprendido entre un mínimo y un máximo es convertido a otro valor final también comprendido entre un mínimo y un máximo. En este ejemplo el valor de pmeterValue comprendido entre 0 y 1023 se convierte en otro proporcional comprendido entre 0 y 255 y se almacena en la variable gapValue.

La función analogWrite() produce una salida de señal PWM por la patilla indicada y con un ciclo útil determinado. Una señal PWM es una señal digital que transita de nivel bajo a nivel alto de forma constante a una frecuencia determinada. Sin embargo la duración del nivel alto puede variar entre 0 % (nivel bajo constante) y el 100% (nivel alto constante) del periodo de esa frecuencia. El ciclo útil se determina con un valor comprendido entre 0 (0%) y 255 (100%).

Lección 8 – Monitor de humedad

Introducción



Un sensor de humedad detecta la humedad del suelo y por lo tanto permite conocer si las plantas de un jardín necesitan o no que se les riegue. Este sensor emplea dos sondas que se introducen en el suelo y por las que se hace pasar la corriente a través de la tierra. Cuanto mayor sea la humedad del terreno mayor será la corriente (menos resistencia) y viceversa.

tienes plantas en casa? Ahora ya sabes que este sensor puede detectar la humedad de la tierra y mostrártela para que sepas cuándo debes regar.

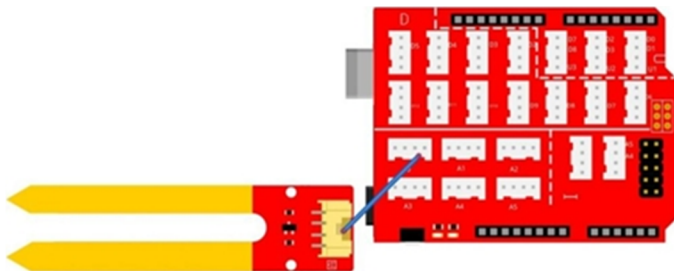
Necesitamos...

- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x Sensor de humedad
- 1 x Crowtail Cable
- 1 x Cable USB

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el sensor de humedad en el puerto de Crowtail Base Shield marcado como A0 como puedes ver en la figura.



Abrir el programa de ejemplo **P08_Monitor_humedad.ino** con el IDE de Arduino y grabarlo sobre la tarjeta controladora.

Qué vas a ver

Abre el monitor serie del IDE de Arduino. Si el sensor esta al aire verás que el valor medido es de 0 (máxima resistencia/mínima intensidad). Si tocas ambos electrodos del sensor medirás la humedad de tu cuerpo. Si los insertas en tierra medirás la humedad de la misma.

Descripción del programa

Variables enteras: `const int MoisturePin = A0; int sensorValue = 0;`

Se declaran dos variables. Una constante para la patilla del sensor de humedad, A0, y otra, sensorValue, para almacenar el valor leído.

Iniciar el puerto Serie: `Serial.begin(9600);`

El puerto serie se utiliza para enviar y recibir datos desde el ordenador. Esta función inicia ese puerto serie y establece que la velocidad de la comunicación entre el Crowduino y el ordenador será a 9600 baudios (bits/segundo). Tanto el Arduino como el ordenador deben estar configurados para trabajar a la misma velocidad.

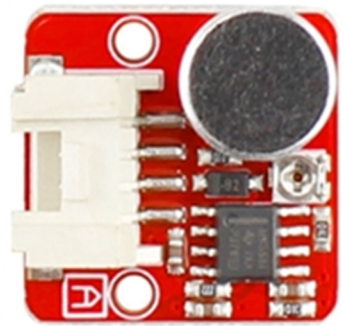
Salida serie: `Serial.print("sensor = "); Serial.println(sensorValue);`

La función `Serial.prin()` se usa para sacar por el monitor serie una cadena de caracteres o el valor de una variable. En el caso de una cadena esta debe ir cerrada entre “ ”, si es una variable basta con indicar su nombre. Observa que en un caso se ha empleado “`print`” y en otro “`println`”. En el segundo caso, además de imprimir lo que se desee visualizar por el puerto serie, también se añade el código de retorno de carro y avance de línea.

Lección 9 – Sensor de sonido

Introducción

El sensor de sonido puede detectar el nivel de ruido del medio ambiente. Se basa en un sencillo micrófono que recibe la señal de sonido, un filtro paso banda y un amplificador LMV358 que genera la señal que recibirá el controlador. Se trata de una señal analógica que varía en función del volumen. Dispone también de un potenciómetro para ajustar el umbral de ruido. Podemos usar este sensor para intentar no ser demasiado ruidoso y molestar a los demás.



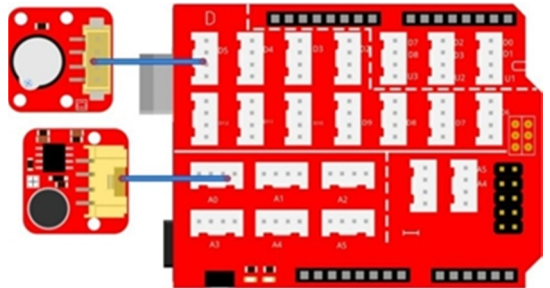
Necesitamos...

- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x Sensor de sonido
- 1 x Zumbador
- 2 x Crowtail Cable
- 1 x Cable USB

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el sensor de sonido en el puerto de Crowtail Base Shield marcado como A0 y el zumbador en el puerto 5 como puedes ver en la figura.



Abrir el programa de ejemplo **P09_Sensor_de_sonido.ino** con el IDE de Arduino y grabarlo sobre la tarjeta controladora.

Qué vas a ver

Si se detecta un volumen alto de ruido, el zumbador se activa durante 1 segundo. Si el ruido persiste el zumbador se mantiene activado, en caso contrario se desactiva.

Descripción del programa

Variables enteras: `int sounPin = A0; int buzzerPin = 5; int soundValue = 0;`

Declara las conexiones del sensor de sonido y del zumbador. Inicia con 0 la variable `soundValue`.

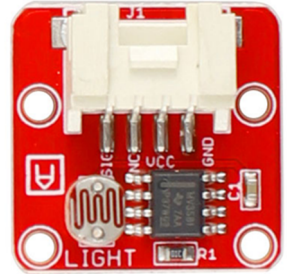
Sentencia if/else: if(condición) {si la condición es cierta se ejecuta este código} else {si la condición es falsa se ejecuta este código}

Si el valor del sonido es superior a 500 se activa el zumbador durante 1segundo y el ciclo se repite. Si el ruido se mantiene el zumbador también se mantiene activo. En caso contrario el zumbador se desconecta. Naturalmente que puedes cambiar el valor 500 por cualquier otro comprendido entre 0 y 1023 con objeto de ajustar el umbral de ruido deseado.

Lección 10 – Alumbrado inteligente

Introducción

El sensor de luz emplea una foto-resistencia GL5516 para detectar la luz visible que incide sobre ella. El valor de esa resistencia disminuye al aumentar la luz. El chip LV358 se emplea como seguidor de tensión que facilita una señal analógica en función de la luz del entorno. Se puede usar como alumbrado inteligente de la calle o del pasillo de tu casa. ¿Has visto las luces de la calle? Se encienden automáticamente cuando oscurece. Con esta lección puedes usar el sensor de luz para hacer tus propias farolas inteligentes.



Necesitamos...

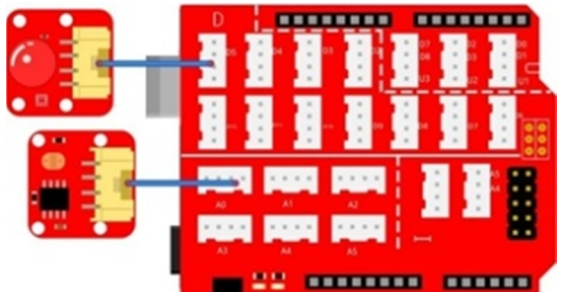
- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x Sensor de luz
- 1 x LED
- 2 x Crowtail Cable
- 1 x Cable USB

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el sensor de luz en el puerto de Crowtail Base Shield marcado como A0 y el led en el puerto 5 como puedes ver en la figura.

Abrir el programa de ejemplo **P10_Alumbrado_inteligente.ino** con el IDE de Arduino y grabarlo sobre la tarjeta controladora.



Qué vas a ver

Cuando incide luz sobre el sensor verás que el led se apaga. Puedes usar una linterna o colocar el sensor de forma que incida luz solar sobre él. Si lo oscureces en un momento dado el led se activa durante 1 segundo. Si persiste la oscuridad el led se mantiene encendido.

Descripción del programa

Variables enteras: `int lighPin = A0; int ledPin = 5; int lighValue = 0;`

Declara las conexiones del sensor de luz y del led en A0 y D5 respectivamente. Inicia con 0 la variable lighValue.

Entrada analógica: `lighValue = analogRead(lighPin);`

En el bucle principal loop() se usa de forma repetida la función analogRead() para leer el valor del sensor de luz y almacenarlo en la variable lighValue.

Sentencia if/else: `if(condición) {si la condición es cierta se ejecuta este código} else {si la condición es falsa se ejecuta este código}`

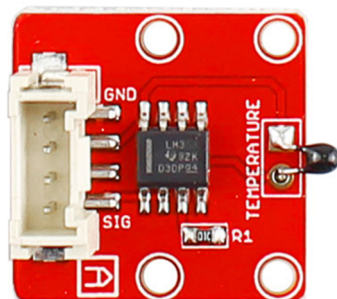
Si el valor de la luz recibida es inferior a 800 se activa el led durante 1 segundo y se vuelve a leer el valor del sensor. Si sigue siendo menor de 800 el led se mantiene encendido. Puedes modificar el valor 800 para hacer un control más o menos sensible de la luz.

Lección 11 – Alarma por calentamiento

Introducción

El sensor de temperatura ambiente emplea un termistor o resistencia NTC cuyo valor aumenta al disminuir la temperatura o viceversa. Crowduino recibe la tensión analógica que proporciona el sensor y calcula así la temperatura en un rango de -40° a 125°C con una precisión de $\pm 1.5^{\circ}\text{C}$. Para obtener la temperatura a partir de la resistencia del sensor se realiza un pequeño cálculo matemático según los datos que facilita el fabricante del termistor.

Muchos son los aparatos que se pueden averiar a causa de una temperatura elevada. Con esta lección aprenderás a detectar esos sobrecalentamientos.



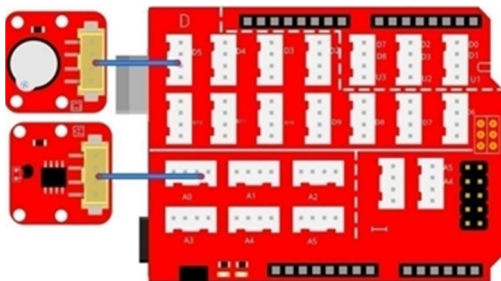
Necesitamos...

- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x Sensor de temperatura
- 1 x zumbador
- 2 x Crowtail Cable
- 1 x Cable USB

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el sensor de temperatura en el puerto de Crowtail Base Shield marcado como A0 y el zumbador en el puerto 5 como puedes ver en la figura.



Abrir el programa de ejemplo

P11_Alarma_por_calentamiento.ino con el IDE de Arduino y grabarlo sobre la tarjeta controladora.

Qué vas a ver

Cuando la temperatura supera los 35°C se activa el zumbador durante 1 segundo a modo de alarma por sobrecalentamiento.

Descripción del programa

Librería Math: `#include <math.h>`

Incluye la librería math.h con lo que podemos usar las funciones matemáticas incluidas en ella.

Variables Float: `float temperature;` `float resistance;`

Almacenan números en punto flotante con valores que pueden tener decimales. Estos valores son más precisos que los enteros. Por ejemplo podemos representar una temperatura como 35.58°C en lugar de 35°C.

Convertir la temperatura: `temperatura = 1/(log(resistance/10000)/B+1/298.15)-273.15;`

Se calcula según indica el datasheet del fabricante a partir del valor del termistor. La función `log()` contenida en la librería Math.h devuelve el logaritmo natural del valor contenido dentro de los paréntesis.

Lección 12 – Función lógica NOT

Introducción

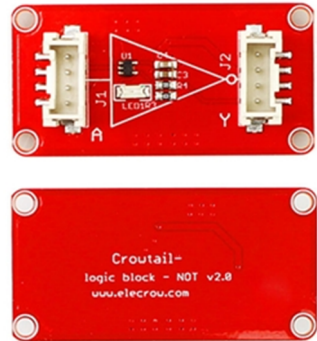
¿Qué es una puerta lógica?

En los circuitos digitales la puerta lógica es el componente básico. Actúa como un interruptor que en ciertas condiciones deja pasar o no a la señal. El nivel de salida es de nivel bajo o alto en función de los niveles de las entradas. Las puertas lógicas se pueden combinar para realizar operaciones lógicas más complejas y son la base de los circuitos digitales. Las tres puertas básicas son: NOT, AND y OR.

Vamos analizar el módulo Crowtail con la función NOT que tiene una entrada y una salida que vale lo contrario de la entrada. Esto es, si la entrada está a nivel bajo la salida pasa a nivel alto y viceversa. La función NOT se representa mediante esta tabla de la verdad.

ENTRADA	SALIDA
A	Y
0	1
1	0

que tiene una entrada y una salida que vale lo contrario de la entrada. Esto es, si la entrada está a nivel bajo la salida pasa a nivel alto y viceversa. La función NOT se representa mediante esta tabla de la verdad.



Necesitamos...

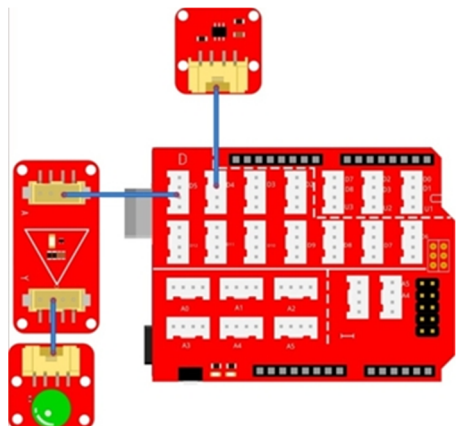
- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x Sensor Táctil
- 1 x Módulo con función NOT
- 1 x led
- 3 x Crowtail Cable
- 1 x Cable USB

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el sensor táctil y la entrada A de la función NOT con D4 y D5 respectivamente. Conectar el led con la salida Y de la función NOT, como puedes ver en la figura.

Abrir el programa de ejemplo **P12_Funcion_logica_NOT.ino** con el IDE de Arduino y grabarlo sobre la tarjeta controladora.



Qué vas a ver

Cargado el programa el led se enciende sin tocar el sensor. Puede parecer raro ya que en el programa se establece que el led se encienda cuando se toque el sensor (`touchState == 1`). Sin embargo mediante la acción de la función NOT, cuando sacamos un nivel bajo por D5, que la entrada A de la función NOT, a su salida tenemos un nivel alto y el led se enciende. Al tocar el sensor, por D5 aparece un nivel alto, pero a la salida de la función NOT tenemos un nivel bajo y el led se apaga.

Descripción del programa

Variables enteras: `int touchPin = 4; int ledPin = 5; int touchState = 0;`

Declara las conexiones del sensor táctil y del led en D4 y D5 respectivamente. Inicia con 0 la variable touchState..

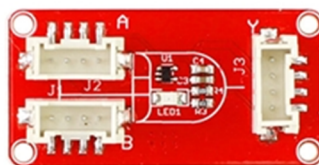
Sentencia if/else: `if(condición) {si la condición es cierta se ejecuta este código} else {si la condición es falsa se ejecuta este código}`

Si no se toca el sensor se saca un bajo por D5 que se aplica a la entrada A de la función NOT. Su salida será alta y el led se enciende. Si se toca el sensor ocurre justo lo contrario.

Lección 13 – Función lógica OR

Introducción

La función lógica OR tiene dos o más entradas y una salida. Esta es de nivel alto cuando una o varias entradas también están a nivel alto. Si todas las entradas están a nivel bajo, la salida también lo estará. Es una buena opción para aquellos proyectos en los que se desea controlar dos o más entradas sin importar cuál de ellas esté activada.



ENTRADAS		SALIDA
A	B	F = A + B
0	0	0
0	1	1
1	0	1
1	1	1

En esta lección se usarán dos entradas, pulsador e interruptor, para controlar el vibrador mediante la función OR.

Necesitamos...

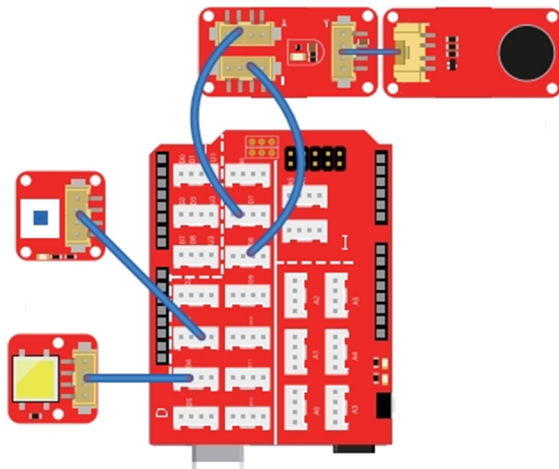
- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x Pulsador
- 1 x Interruptor
- 1 x Vibrador
- 1 x Función OR
- 5 x Crowtail Cable
- 1 x Cable USB

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el interruptor y el pulsador en las entradas D3 y D4. Las entradas A y B de la función OR con D7 y D8. La salida Y de la función OR con el motor vibrador, como puedes ver en la figura.

Abrir el programa de ejemplo **P13_Funcion_logica_OR.ino** con el IDE de Arduino y grabarlo sobre la tarjeta controladora.



Qué vas a ver

Cuando se activan el pulsador, o el interruptor o ambos, el vibrador se activa. Esto demuestra que en una función OR la salida es alta cuando cualquiera de las entradas también lo es.

Descripción del programa

Variables enteras: `int logicalA = 7; int logicalB = 8;`

Declara las variables `logicalA` y `logicalB` y les asigna los valores 7 y 8, que son pines D7 y D8 que conectamos con las entradas A y B de la función OR.

Sentencia if/else: `if(condición) {si la condición es cierta se ejecuta este código} else {si la condición es falsa se ejecuta este código}`

Las sentencias `if/else` se pueden usar tantas veces como sea necesario. En el ejemplo se emplean tres: para comprobar si el pulsador está activado, si está activado el interruptor o si ambos están activados.

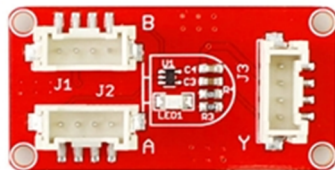
Lección 14 – Función lógica AND

Introducción

La función AND tiene dos o más entradas y una salida que será de nivel alto cuando todas las entradas también estén a nivel alto. Si cualquiera de las entradas está a nivel bajo, la salida también lo estará. Esta función, junto con las anteriores, permite resolver complejas situaciones o ecuaciones lógicas.

ENTRADAS		SALIDA
A	B	$F = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

En esta lección vamos a usar la lógica de la función AND para un control inteligente de alumbrado del pasillo de nuestra casa.



Necesitamos...

- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x sensor de sonido
- 1 x sensor de luz
- 1 x LED
- 1 x Función AND
- 5 x Crowtail Cable
- 1 x Cable USB

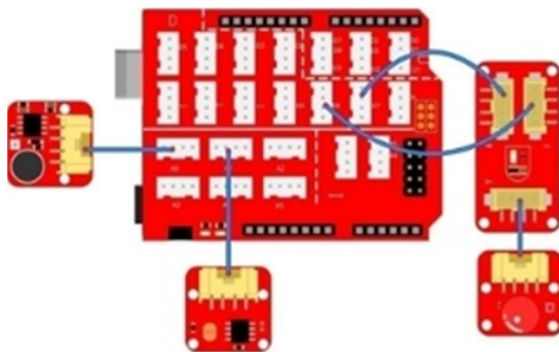
Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el sensor de sonido y el sensor de luz en las entradas A0 y A1. Las entradas A y B de la función AND con D7 y D8. La salida Y de la función AND con el LED, como puedes ver en la figura.

Abrir el programa de ejemplo

P14_Funcion_logica_AND.ino con el IDE de Arduino y grabarlo sobre la tarjeta controladora.



Qué vas a ver

Si colocas este circuito en un entorno con mucha luz y muy silencioso verás que el led no se enciende. Tampoco se enciende si lo colocas en un entorno oscuro pero silencioso. Con la función AND se deben dar ambas situaciones, oscuridad y ruido, para que se active el led. Es decir, las dos entradas deben de estar a nivel alto.

Descripción del programa

Variables enteras: `int soundPin = A0; int lighPin = A1;`

Declara las variables `sounPin` y `lighPin` con los pines `A0` y `A1`. Son entradas analógicas ya que los sensores de sonido y de luz también lo son. Según los valores de ambas magnitudes se activa o no el led.

Analog Input: `lightValue=analogRead(lightPin); soundValue=analogRead(soundPin);`

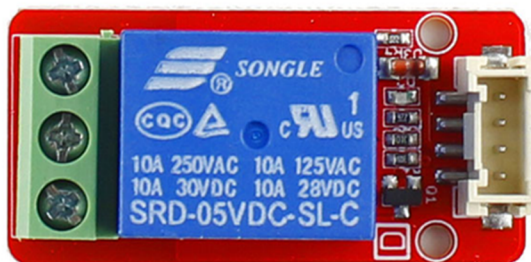
Lee los valores de los sensores de luz y sonido y los almacena en las variables `lightValue` y `soundValue` respectivamente.

Sentencia if/else: `if(condición) {si la condición es cierta se ejecuta este código} else {si la condición es falsa se ejecuta este código}`

Las sentencias `if/else` se pueden usar tantas veces como sea necesario. En el ejemplo se emplean dos: para comprobar si el sonido > 500 , y otra para comprobar si la luz está por debajo de 800 . Si se producen ambas situaciones el led se enciende.

Lección 15 – Sistema automático de control

Introducción



El relé es uno de los dispositivos más usado en aplicaciones reales como es en la automatización del hogar, cuando es necesario controlar aparatos de gran consumo como aire acondicionado, calentador, etc...

El módulo del relé tiene un borne con tres conexiones: Común (COM), normalmente cerrado (NC) y normalmente abierto (NO). Se comporta como un conmutador. La conexión entre COM y NC se produce cuando el relé está desactivado, no hay conexión entre COM y NO. Por el contrario, cuando el relé se activa, la conexión entre COM y NC se abre y ahora se cierra la conexión entre COM y NO.

La corriente máxima que soportan los contactos cuando están cerrados respecto a COM es de 30A a 240 VAC o bien de 20A a 30VDC. Esto es suficiente para controlar un buen número de electrodomésticos diferentes.

En esta lección usaremos el sensor de temperatura y el relé para activar una luz indicadora que se activa automáticamente cuando se eleva la temperatura.

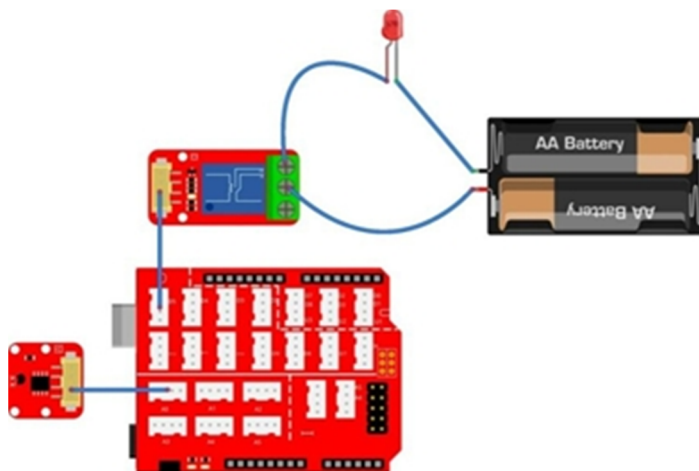
Necesitamos...

- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x sensor de temperatura
- 1 x Módulo con relé
- 1 x led
- 2 x Crowtail Cable
- 1 x Cable USB
- 1 x Portapilas

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el relé y el sensor de temperatura en las patillas D5 y A0. Conectar el positivo de las pilas (cable rojo) en COM del relé, el GND de las pilas (cable negro) en la patilla corta del led y la pata larga de este en NO del relé. Mira la figura.



Abrir el programa de ejemplo [P15_Sistema_automático_de_control.ino](#) con el IDE de Arduino y grabarlo sobre la tarjeta controladora.

Qué vas a ver

Si la temperatura ambiente no supera los 30°C el relé se mantiene desconectado y el led, que actúa como carga a controlar, desactivado. Al superar los 30°C el relé se activa, se cierra la conexión entre los terminales COM y NO y el led se enciende con la tensión que aporta el portapilas. En lugar de un led se podría conectar otro tipo de carga como un ventilador a 220 VAC.

Descripción del programa

Librería Math: `#include <math.h>`

Incluye la librería math.h con lo que podemos usar las funciones matemáticas incluidas en ella.

Variables Float: `float temperature;` `float resistance;`

Almacenan números en punto flotante con valores que pueden tener decimales. Estos valores son más precisos que los enteros. Por ejemplo podemos representar una temperatura como 35.58°C en lugar de 35°C.

Lección 16 – Un número sobre display de 4 dígitos

Introducción

Para controlar una pantalla de 4 dígitos harían falta 12 patillas, pero este módulo Crowtail utiliza el chip TM1650 que emplea un interface I2C, por lo que solo emplearemos dos patillas de Crowduino para visualizar cualquier contenido e incluso regular el brillo de la pantalla. Ideal para aplicaciones que requieran una visualización numérica. En esta lección mostraremos números e incluso puntos decimales.



Necesitamos...

- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x Display de 4 dígitos
- 1 x Crowtail Cable
- 1 x Cable USB

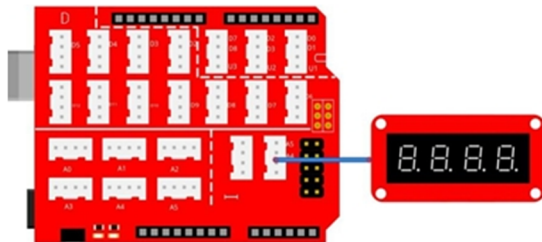
Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO
PASO 2: Conectar el módulo Crowtail con los 4 dígitos, en cualquiera de los dos puertos I con las señales A4 y A5. Mira la figura.

En la misma carpeta en la que se encuentra el ejemplo de esta lección, se encuentra la carpeta con la librería TM1650. Debes copiarla en la carpeta Library de Arduino. Ahora puedes abrir el programa de ejemplo **P16_Un_numero_sobre_display_de_4_digitos.ino** con el IDE y grabarlo sobre la tarjeta controladora.

Qué vas a ver

Sobre el display se visualizan los números del 1 al 4 y luego de 5 al 8 de forma repetitiva.



Descripción del programa

Importar librería: `#include "TM1650.h" #include <inttypes.h>`

"TM1650.h" contiene funciones muy completas para controlar el display de 4 dígitos. Necesitamos instalarla en la carpeta Library de Arduino si queremos hacer uso de esas funciones. La librería "Inttypes.h" nos permite definir tipo de enteros que contienen información sobre el tamaño de ese tipo.

Array: `static uint8_t TubeTab[] = {} static uint8_t TubeTabwithPoint[] = {}`

Los arrays son una forma de programación que permiten organizar un conjunto de elementos o datos del mismo tipo para facilitar el procesamiento. Aquí hemos creado dos arrays para almacenar los valores que decodifican los números que queremos mostrar. Si queremos obtener uno de esos valores solo tendremos que indicar el nombre del array y la posición que ocupa (empezando desde 0). La palabra "static" es una palabra clave. Las variables declaradas con ella no pueden ser repetidas. "uint8_t" es otra palabra clave para definir que cada elemento del array es del tipo entero sin signo que ocupa 8 bits (1 byte). Previamente se debe incluir la librería "inttypes.h".

Crear una instancia: `TM1650 DigitalLED(A5,A4);`

DigitalLED es una instancia que permite usar las funciones de la librería "TM1650.h". En este caso utiliza las señales A5 y A4 como señales I2C.

Borrar pantalla: `DigitalLED.clearDisplay();`

Es una de las funciones de la librería "TM1650.h". Borra los cuatro dígitos del display.

Digital Display: `DigitalLED.display(dígito, valor);`

Es otra de las funciones de la librería "TM1650.h", que visualiza un dígito sobre el display. Tiene dos parámetros. Con el primero indicamos el N° de dígito (0 – 3) y con el segundo el valor a visualizar en ese dígito.

Lección 17 – Semáforo

Introducción

Ya hemos usado todos los módulos del Crowtail Starter Kit, ahora trataremos de hacer algunos proyectos con ellos. En esta ocasión vamos a hacer un semáforo empleando el display y los tres leds, rojo, amarillo y verde. La secuencia de encendido es verde, amarillo y rojo. El display irá mostrando el tiempo restante en que permanece encendida cada luz.

Necesitamos...

- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x Display de 4 dígitos
- 1 x led rojo
- 1 x led amarillo
- 1 x led verde
- 4 x Crowtail Cable
- 1 x Cable USB

Conexiones

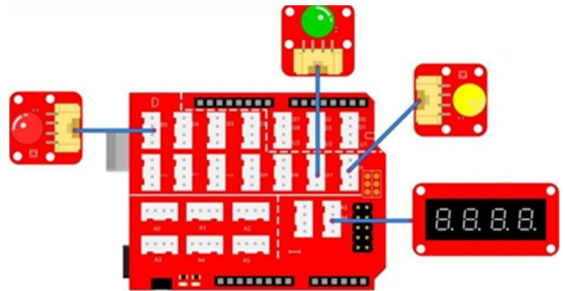
PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el módulo Crowtail con los 4 dígitos, en cualquiera de los dos puertos I con las señales A4 y A5. Conectar los leds rojo, amarillo y verde en D5, D6 y D7 respectivamente. Mira la figura.

Ahora abre el programa de ejemplo **P17_Semaforo.ino** con el IDE y grábalo sobre la tarjeta controladora.

Qué vas a ver

La led verde se enciende primero. El tiempo que debe permanecer encendido se va indicando en el display. Luego se apaga el led verde y se enciende el amarillo. En el display también se visualiza el tiempo restante. Finalmente se apaga el led amarillo, se enciende el rojo y nuevamente el display muestra el tiempo. El ciclo se repite indefinidamente.



Descripción del programa

Importar librería: `#include "TM1650.h" #include <inttypes.h>`

"TM1650.h" contiene funciones muy completas para controlar el display de 4 dígitos. Necesitamos instalarla en la carpeta Library de Arduino si queremos hacer uso de esas funciones.

Variables enteras: `int redPin = 5; int yellowPin = 6 int greenPin = 7; int redValue=5; int yellowValue=2; int greenValue=5;`

Se declaran las patillas para conectar los tres leds de colores, y el valor en segundos que cada led debe mantenerse encendido. Lo puedes modificar a tu gusto.

Digital Display: `DigitalLED.display(dígito, valor);`

Es una de las funciones de la librería “TM1650.h”, que visualiza un dígito sobre el display. Tiene dos parámetros. Con el primero indicamos el N° de dígito (0 – 3) y con el segundo el valor a visualizar en ese dígito.

Decrementar: `greenValue-=1; yellowValue-=1; redValue-=1;`

El contenido de cada variable se decrementa en una unidad. Generalmente se suele poner “variable = variable – 1”, pero de forma abreviada se puede expresar: “variable -= 1”

Sentencia While: `while(condición) {código a ejecutar si se cumple esa condición}`

Al igual que if/else la sentencia while también ejecuta instrucciones en forma de bucle condicionado. Todas las instrucciones contenidas entre las llaves se ejecutan mientras se cumple la condición. En el momento en que la condición sea falsa, finaliza el bucle y la ejecución continua tras la llaves.

Setup y Loop: `void setup(){se ejecuta una vez} & void loop(){ejecución indefinida}`

En la función setup() se configuran las patillas de los leds como salidas. En la función loop() se establece la secuencia de funcionamiento que se ejecuta de forma indefinida. Empieza con la luz verde, luego la amarilla y finalmente la roja. Todo ello se repite constantemente.

Lección 18 – Luz inteligente de pasillo

Introducción

La luz del pasillo se enciende automáticamente cuando no haya suficiente iluminación ambiental y se detecte, mediante el sonido, una presencia. Se apaga al de 5 segundos. Usaremos el sensor de luz y el de sonido para detectar esas situaciones. Además usaremos el sensor táctil para encender manualmente la luz cuando se detecte un toque en el mismo.

Necesitamos...

- 1 x Crowduino UNO-SD
- 1 x Sensor de sonido
- 1 x Crowtail Base Shield
- 1 x Sensor de luz

- 1 x Sensor táctil
- 1 x led verde
- 4 x Crowtail Cable
- 1 x Cable USB

Conexiones

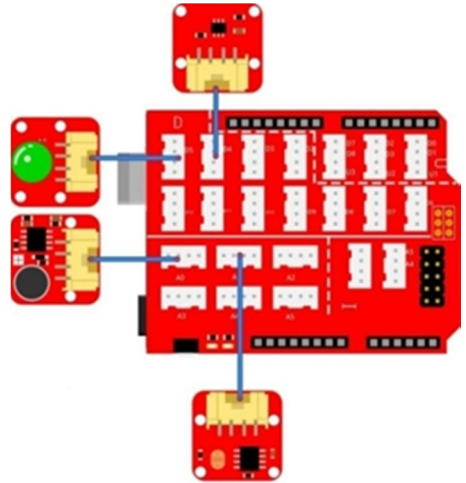
PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO

PASO 2: Conectar el sensor de sonido y el de luz a las entradas analógicas A0 y A1 respectivamente. El sensor táctil se conecta con la entrada D4 y el led en la salida D5. Mira la figura.

Ahora abre el programa de ejemplo **P18_Luz_inteligente_pasillo.ino** con el IDE de Arduino y grábalo sobre la tarjeta controladora.

Qué vas a ver

Cuando se oscurece el sensor de luz y se detecta un ruido, el led de alumbrado se enciende automáticamente durante 5 segundos. Ocurre exactamente lo mismo cuando se toca el sensor táctil. Si hay poca luz pero no hay ruido, o bien se detecta un ruido pero hay mucha luz, el led permanece apagado.



Descripción del programa

Variables enteras: `int soundPin = A0; int lightPin = A1; int touchPin = 4; int ledPin = 5; int soundValue = 0; int lightValue = 0; int touchValue = 0;`

Se declaran las patillas para conectar los sensores analógicos de sonido y luz, el sensor digital táctil, y el led de salida que simula el alumbrado. También se declaran las variables que contienen los valores leídos desde los tres sensores.

Entradas/Salidas: `pinMode(soundPin,INPUT); pinMode(lightPin,INPUT); pinMode(touchPin,INPUT); pinMode(ledPin,OUTPUT);`

En la función `setup()` se configuran las patillas que actúan como entradas y salidas

Sentencia if/else: `if(condición) {si la condición es cierta se ejecuta este código} else {si la condición es falsa se ejecuta este código}`

Si el sensor táctil está activado se enciende el alumbrado durante 5 segundos. En caso contrario se lee el valor de los sensores de sonido y de luz. Si ambos valores coinciden con los márgenes establecidos, también se activa el led de alumbrado. Si no se cumple ninguna de las condiciones, el alumbrado permanece desactivado.

Lección 19 – Sistema de riego

Introducción

¿Tienes una planta en casa? ¿Te olvidas de regarla? Con esta lección verás cómo conocer el grado de humedad para recordarte que tienes que regar. Mediante el sensor de humedad se lee el valor de ésta. Lo dividiremos en tres rangos y el display mostrará el valor actual. Mediante el zumbador y dos leds podrás conocer si el riego es o no urgente.

Necesitamos...

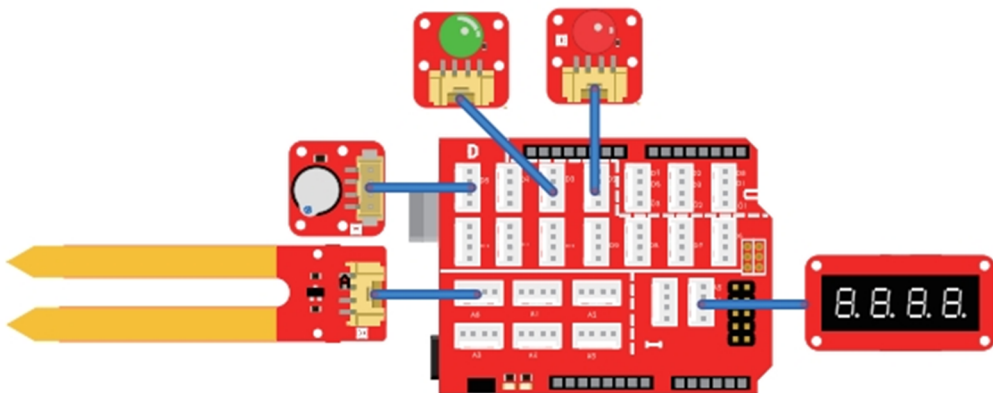
- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x Sensor de humedad
- 1 x display de 4 dígitos
- 1 x zumbador
- 1 x led rojo
- 1 x led verde
- 5 x Crowtail Cable
- 1 x Cable USB

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO.

PASO 2: Conectar el sensor de humedad a la entrada analógica A0, el display de 4 dígitos en uno de los conectores del puerto I (I2C) y el zumbador, el led verde y el led rojo en las salidas D5, D3 y D2 respectivamente. Mira la figura.

Ahora abre el programa de ejemplo **P19_Sistema_de_riego.ino** con el IDE de Arduino y grábalo sobre la tarjeta controladora.



Qué vas a ver

El valor de la humedad que facilita el sensor se divide en tres niveles o escalas que se visualizan sobre el display de 4 dígitos. El primer nivel es cuando la humedad está por debajo del 10%. Se activa el zumbador a modo de alarma para proceder a regar inmediatamente. Si la humedad se encuentra entre el 10% y el 20% se activa el led rojo para indicar que hay que regar próximamente. Cuando la humedad está por encima del 20% se activa el led verde. A la planta no le hace falta regar.

Descripción del programa

Importar librería: `#include "TM1650.h" #include <inttypes.h>`

"TM1650.h" contiene funciones muy completas para controlar el display de 4 dígitos. Necesitamos instalarla en la carpeta Library de Arduino si queremos hacer uso de esas funciones. La librería "Inttypes.h" nos permite definir tipo de enteros que contienen información sobre el tamaño de ese tipo

Variables enteras: `int d,t,h;`

Declara tres variables donde se almacena las unidades, decenas y centenas que representan la humedad y que serán visualizadas en los dígitos correspondientes del display.

Array: `static uint8_t TubeTab[] = {}`

Los arrays son una forma de programación que permiten organizar un conjunto de elementos o datos del mismo tipo para facilitar el procesamiento. Si queremos obtener uno de esos valores solo tendremos que indicar el nombre del array y la posición que ocupa (empezando desde 0). La palabra "static" es una palabra clave y la variable no se pueden repetir. "uint8_t" es otra palabra clave para definir que cada elemento del array es del tipo entero sin signo que ocupa 8 bits (1 byte). Previamente se debía incluir la librería "inttypes.h".

Divisor y resto: t = moistureValue/10; d = moistureValue%10;

“t” y “d” representan los dígitos de las decenas y las unidades del valor de la humedad. Si un valor de dos dígitos lo divides entre 10 obtienes las decenas (p.e. $52 / 10 = 5$). Si de ese mismo valor te quedas con el módulo o resto, obtienes las unidades (p.e. $52 \% 10 = 2$). Así pues tenemos los dígitos 5 y 2 para visualizar en el display.

Bucle loop() {código a ejecutar}

En el bucle principal se lee el sensor de humedad. Mediante sentencias if/else se determina el nivel de la misma y se actúa sobre el zumbador y los leds.

Lección 20 – Ajustar brillo con display

Introducción

¿Recuerdas la regulación del brillo de un led que hicimos en la lección 7? En esta ocasión haremos algo similar, pero el valor actual del brillo se visualizará sobre el display de 4 dígitos.

Necesitamos...

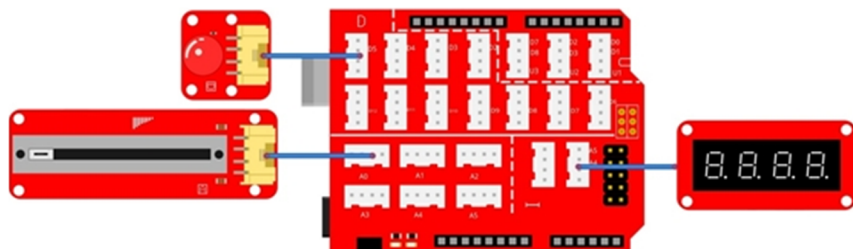
- 1 x Crowduino UNO-SD
- 1 x Crowtail Base Shield
- 1 x Potenciómetro
- 1 x display de 4 dígitos
- 1 x led rojo
- 3 x Crowtail Cable
- 1 x Cable USB

Conexiones

PASO 1: Insertar la tarjeta Crowtail Base Shield sobre la tarjeta controladora Crowduino UNO.

PASO 2: Conectar el potenciómetro a la entrada analógica A0, el display de 4 dígitos en uno de los conectores del puerto I (I2C) y el led rojo en la salida D5. Mira la figura.

Ahora abre el programa de ejemplo [P20_Ajustar_brillo_con_display.ino](#) con el IDE de Arduino y grábalo sobre la tarjeta controladora.



Qué vas a ver

Según desplazamos el potenciómetro de un extremo a otro, regularemos el brillo del led rojo de forma similar a como se hizo en la Lección 7. En esta ocasión el display visualizará un valor entre 0 y 255 que representa la señal PWM aplicada al led.

Descripción del programa

Importar librería: `#include "TM1650.h" #include <inttypes.h>`

"TM1650.h" contiene funciones muy completas para controlar el display de 4 dígitos. Necesitamos instalarla en la carpeta Library de Arduino si queremos hacer uso de esas funciones. La librería "Inttypes.h" nos permite definir tipo de enteros que contienen información sobre el tamaño de ese tipo

Función matemática de Arduino map(): `int gapValue = map(valor, valor mínimo inicial, valor máximo inicial, valor mínimo final, valor máximo final)`

Pasa números de un rango a otro de forma proporcional. Un valor inicial comprendido entre un mínimo y un máximo es convertido a otro valor final también comprendido entre un mínimo y un máximo. En este ejemplo el valor de `pmeterValue` comprendido entre 0 y 1023 se convierte en otro proporcional comprendido entre 0 y 255 y se almacena en la variable `gapValue`.

Divisor y resto: `h = gapValue/100; t = moistureValue/10%10; d = moistureValue%10;`


"h", "t" y "d" representan los dígitos de las centenas, decenas y las unidades de un valor de tres dígitos. Si a ese valor lo divides entre 100 obtienes las centenas (p.e. $521 / 100 = 5$). Si a ese valor lo divides entre 10 y luego calculas el resto, obtienes las decenas (p.e. $521 / 10 \% 10 = 2$). Por último, si calculas el resto de ese valor dividido entre 10, obtienes las unidades (p.e. $521 \% 10 = 1$). Así pues tenemos los dígitos 5, 2 y 1 para visualizar en el display. ¡¡Ya puedes visualizar cualquier valor de hasta 4 dígitos sobre ese display!!

Bucle loop() {código a ejecutar}

En el bucle principal se lee el valor del potenciómetro (0 – 1023) y se redondea o "mapea" a otro equivalente comprendido entre 0 y 255. Se genera la señal PWM para regular el brillo del led, y se calculan las centenas, decenas y unidades para visualizar sobre el display.



 info@elecrown.com

 +86 0755-23204330

 www.elecrown.com

 8th Floor, F-building, Fusen Industry
park, Gushu Town, Bao'an District,
Shenzhen, China.



Traducido por MK Electrónica (www.mkelectronica.com; info@mkelectronica.com)