Actividades hogareñas Programación de dispositivos

* Objetivos: Hoy vamos a dar nuestros primeros pasos en programación. En nuestra materia vamos a utilizar un lenguaje de programación llamado C. A lo largo del año desarrollemos aplicaciones en C con la plataforma Arduino.

Cuando las clases sean retomadas, seguiremos trabajando sobre estos temas.

Si alguno tiene alguna duda o no entiende algo. ¡¡¡Por favor hagan preguntas!!!

* Para empezar necesitamos tener instalado en la PC que vamos a trabajar el IDE arduino. Adjunto el link para realizar la descarga:

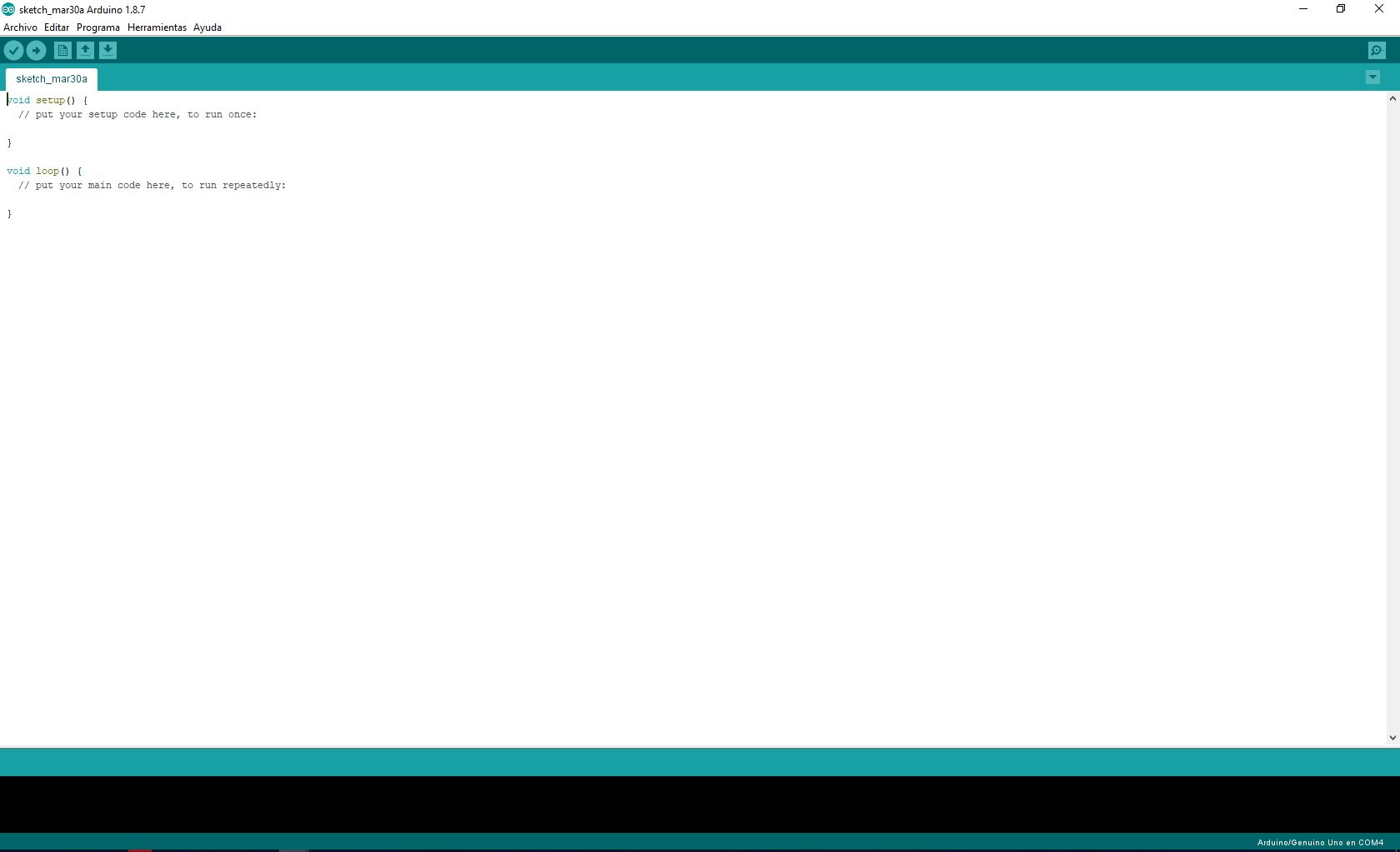
https://www.arduino.cc/download\_handler.php

Lo único que tienen que hacer es copiar y pegar el link en un navegador y seleccionar donde van a guardar el instalador. En cuanto a la instalación es la típica instalación de apretar siempre “siguiente” y cuando pide instalar algun driver solamente clickear “si”. Para el que no tiene mucha práctica sobre el tema de bajar archivos, instalar, etc… Les adjunto un link de Youtube para que puedan ver como instalar el ide paso a paso sin problemas.

<https://www.youtube.com/watch?v=8k3v63M6rEQ>

1. Una vez instalado y abierto el IDE. Nos aparecera algo asi

Esto se llama Sketch. Aca ira todo el codigo que mas luego será cargado en el micro.

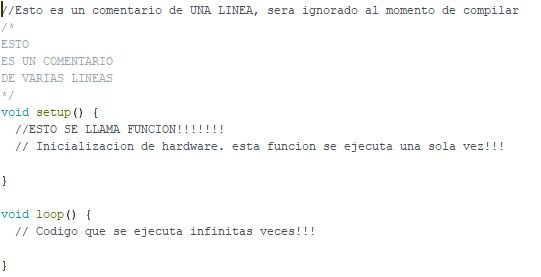


Veamos más en detalles qué es esto.

Fijense que algunas palabras están marcadas de ciertos colores.

La línea que empieza con el carácter’ **//** ‘es un comentario. Los comentarios sirven para hacer notas que nos ayudan a entender mejor el código. Al momento de compilar (ya veremos más adelante que es esto en detalle) las líneas comentadas serán ignoradas por el compilador. Los comentarios pueden ser de una o varias líneas como se ve en el ejemplo. Si el comentario es de varias líneas, tiene que empezar con los caracteres ‘ **/\*** ‘ y terminar con los caracteres **’\*/**’ .

Una particularidad de los comentarios es que pueden ir en cualquier parte del sketch.



Luego de los comentarios tenemos una línea  esto nosotros lo llamaremos la función de **setup**. Fijense que tenemos este símbolo ‘**{**‘ al cual lo llamaremos **llave de apertura** y otra llave ‘**}**’ al cual llamaremos **llave de cierre.** Todo el código contenido en el setup, tiene que estar escrito entre la llave de apertura y de cierre.

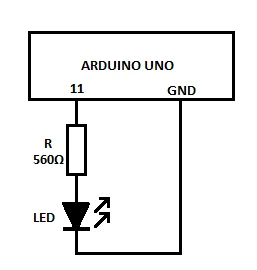
En el setup irán escritas todas las inicializaciones de hardware que vamos a utilizar. Tiene la particularidad que las instrucciones serán ejecutadas **una sola vez**.

Continuamos leyendo las líneas de abajo y vemos que aparece llamaremos la función de **loop**. Fijense que tenemos este símbolo ‘**{**‘ al cual lo llamaremos **llave de apertura** y otra llave ‘**}**’ al cual llamaremos **llave de cierre.** Todo el código contenido en el loop, tiene que estar escrito entre la llave de apertura y de cierre.

Acá estarán contenidas todas las líneas de código que formarán nuestra aplicación. Tiene la particularidad que las instrucciones serán ejecutadas **infinitas veces**.

Entonces resumiendo. La diferencia entre el **loop** y **setup**, es que loop ejecutara las líneas de código que contenga infinitas veces mientras que en el setup solo ejecutara las líneas de código una vez.

Ejemplo de aplicación. Se desea realizar el control de una baliza, cuyo tiempo de apagado sea de 1000 ms al igual que el tiempo de prendido.



***Circuito***

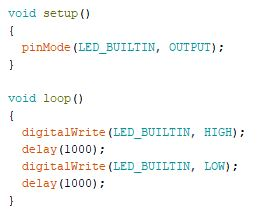
Entonces, ¿qué tenemos que hacer?. La respuesta haciendo hincapié en la actividad anterior es pensar paso a paso que voy a necesitar.

Por empezar, como se ve en el circuito, tengo solamente un LED que actúa representando la baliza. Un LED, es una **salida**. Las salidas es el comunicador por decirlo de alguna forma entre el mundo exterior y el microcontrolador.

Entonces en el setup debo especificar que voy a utilizar una salida como se en la imagen. La instrucción  sirve para determinar si un determinado pin va a ser **entrada** o **salida,** más luego hablaremos de que es una entrada. Lo que está entre paréntesis ‘**()**’, se llaman **parámetros**. En este caso  solo puede recibir dos parámetros separados uno del otro por una coma ‘**,**’. Analicemos qué puede hacer cada parámetro. Por empezar el primer parámetro, es el número de pin que yo voy a querer usar, en nuestro caso vemos el circuito y el pin es el número 11, da la casualidad que ese led está colocado en placa, por eso lo llamamos . El segundo parámetro indica si mi pin va a actuar como entrada o salida digital, en nuestro caso como ya lo aclaramos anteriormente es una salida por eso ponemos . Terminando, vemos que la instrucción  como todas las instrucciones, ***TERMINAN EN PUNTO Y COMA ; !!!***

***ESTO ES SUPER IMPORTANTE YA QUE SI NO SE PONE ; AL FINAL DE UNA INSTRUCCIÓN NOS DARÁ ERROR.***

La instrucción  solo será ejecutada una vez, ya que está en el setup y es la inicialización del hardware que vamos a utilizar. Siempre que utilicemos algún hardware nuevo, tendremos que inicializarlo primero.



Seguimos analizando nos encontramos con el loop. Como dijimos antes, todas las instrucciones que están aquí adentro serán ejecutadas secuencialmente infinitas veces.

Empezando por la sentencia , que sirve para poner un determinado pin en ‘1’ o ‘0’ lógico. El primer argumento que recibe es el número de pin que vamos a poner en ‘1’ o ‘0’, en nuestro caso es  que es el número de pin 11, luego sigue en qué estado va a estar, en este caso será  o es lo mismo que decir ‘1’. Después de haber prendido el led, tendremos una demora , entre paréntesis va el tiempo medido en milisegundos. Cuando el delay es ejecutado, el programa no hace nada, por decirlo de alguna forma, se queda perdiendo tiempo. Pasado el delay, la salida que contiene el led, se pondrá en ‘0’, seguido de otro delay de 1000 milisegundos. Estas instrucciones estan dentro del loop, lo que quiere decir es que cuando se termina de ejecutar el último delay, la secuencia vuelve al principio, osea se ejecutaría la sentencia  de vuelta.

Lo que nuestros ojos verían en la placa es como el led se prende y apaga cada 1 segundo infinitas veces

