

Arduino setup loop serialEvent

Microprocesador, informática

El microprocesador (o simplemente procesador) es el circuito integrado central más complejo de un sistema informático; a modo de ilustración, se le suele llamar por analogía el «cerebro» de un ordenador. Es el encargado de ejecutar todos los programas, desde el sistema operativo hasta las aplicaciones de usuario; sólo ejecuta instrucciones en lenguaje binario, realizando operaciones aritméticas y lógicas simples, tales como sumar, restar, multiplicar, dividir, las lógicas binarias y accesos a memoria.

Raspberry Pi, ejemplo de microprocesador

La Raspberry Pi es una serie de ordenadores de placa reducida, ordenadores de placa única u ordenadores de placa simple (SBC) de bajo costo desarrollado en el Reino Unido por la Raspberry Pi Foundation, como clon directo del Apple Mac Mini, que se lanzó 7 años antes en enero de 2005, con el objetivo de poner en manos de las personas de todo el mundo el poder de la informática y la creación digital. Si bien el modelo original buscaba la promoción de la enseñanza de informática en las escuelas, este acabó siendo más popular de lo que se esperaba, hasta incluso vendiéndose fuera del mercado objetivo para usos como robótica. No incluye periféricos (como teclado y ratón) o carcasa.

Microcontrolador, tecnología

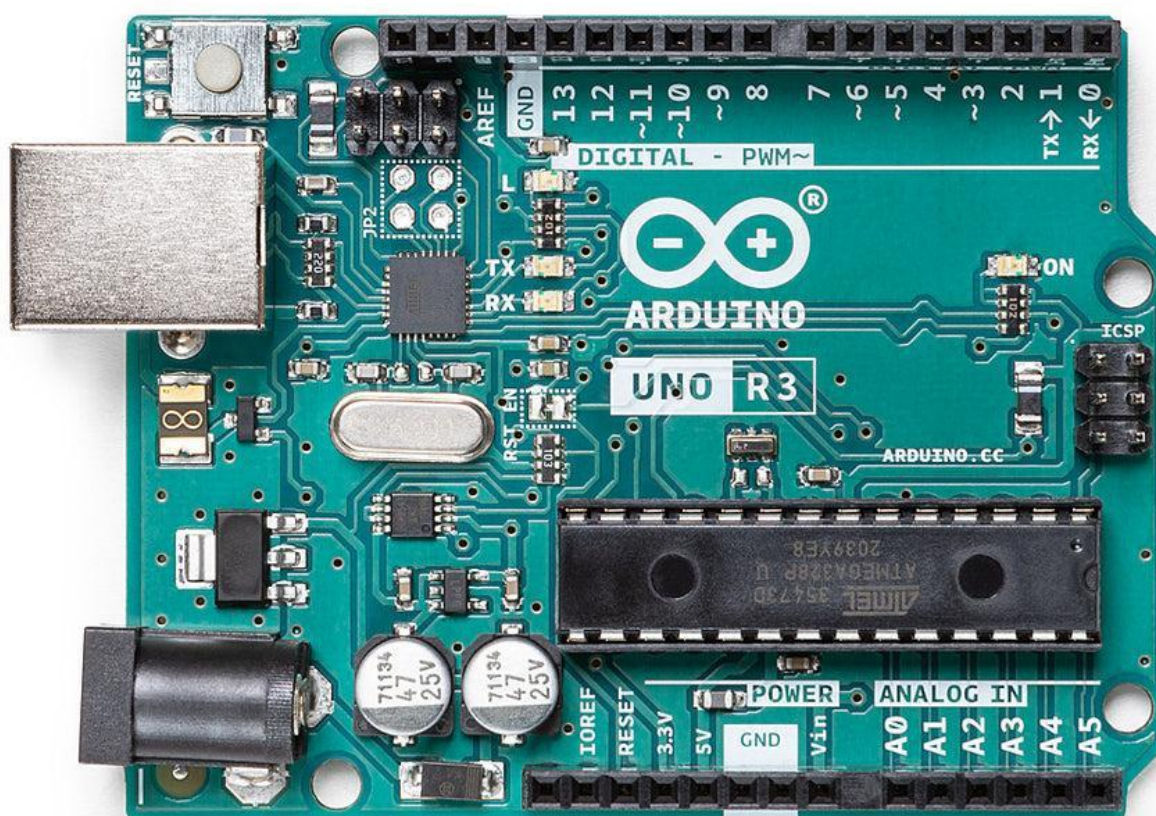
Un microcontrolador (abreviado μC , UC o mCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales que cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

Arduino, ejemplo de microcontrolador

Es una compañía de desarrollo de software y hardware libres, así como una comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan detectar y controlar objetos del mundo real. Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios. Los productos que vende la compañía son distribuidos como Hardware y Software Libre, bajo la Licencia Pública General de GNU (GPL) y la Licencia Pública General Reducida de GNU (LGPL), permitiendo la manufactura de las placas Arduino y distribución del software por cualquier individuo. Las placas Arduino están disponibles comercialmente en forma de placas ensambladas o también en forma de kits, hazlo tú mismo (del inglés DIY: "Do It Yourself").

Arduino UNO R3

El Arduino Uno es una placa de microcontrolador de código abierto basado en el microchip ATmega328P y desarrollado por Arduino.cc. La placa está equipada con conjuntos de pines de E/S digitales y analógicas que pueden conectarse a varias placas de expansión y otros circuitos. La placa tiene 14 pines digitales, 6 pines analógicos y programables con el Arduino IDE (Entorno de desarrollo integrado) a través de un cable USB tipo B. Puede ser alimentado por el cable USB o por una batería externa de 9 voltios, aunque acepta voltajes entre 7 y 20 voltios.



Arduino UNO R3, hardware

Tinkercad → <https://www.tinkercad.com/>

Código fuente

El código fuente de un programa informático (o software) es un conjunto de líneas de texto con los pasos que debe seguir la computadora para ejecutar un cargador. El código fuente de un programa está escrito por un programador en algún lenguaje de programación. Hay diferentes lenguajes de programación, su uso depende del contexto y

del campo de aplicación.

Comentarios /* */ o //

En la programación de computadoras, un comentario es una explicación o anotación legible por el programador en el código fuente de un programa de computadora. Se agregan con el propósito de hacer que el código fuente sea más fácil de entender para los humanos y, por lo general, los compiladores e intérpretes los ignoran. La sintaxis de los comentarios en varios lenguajes de programación varía considerablemente.

TODO

Las etiquetas TODO son un tipo especial de comentario, que contienen información acerca de funcionalidades y características aún no implementadas en un programa informático.

Según la disponibilidad de tiempo y de recursos será probable que aparezcan en versiones futuras del programa, o en revisiones del proyecto.

Su nombre viene del inglés por To do ("por hacer").

Funciones

En informática, una función (también llamado bloque, procedimiento, subprograma, subrutina, rutina o método), como idea general, se presenta como un subalgoritmo que forma parte del algoritmo principal, el cual permite resolver una tarea específica. Algunos lenguajes de programación, como Visual Basic .NET o Fortran, utilizan el nombre función para referirse a subrutinas que devuelven un valor.

setup

En informática, la configuración es un conjunto de datos que determina el valor de algunas variables de un programa o de un sistema operativo. Estas opciones generalmente son cargadas en su inicio y en algunos casos se deberá reiniciar para poder ver los cambios, ya que el programa no podrá cargarlos mientras se esté ejecutando, si la configuración aún no ha sido definida por el usuario (personalizada), el programa o sistema cargará la configuración predeterminada.

loop

Un bucle o ciclo, en programación, es una secuencia de instrucciones de código que se ejecuta repetidas veces, hasta que la condición asignada a dicho bucle deja de cumplirse. Los tres bucles más utilizados en programación son el bucle while, el bucle for

y el bucle do-while.

serialEvent

Llamado al final de loop() cuando los datos están disponibles. Use Serial.read() para capturar estos datos.

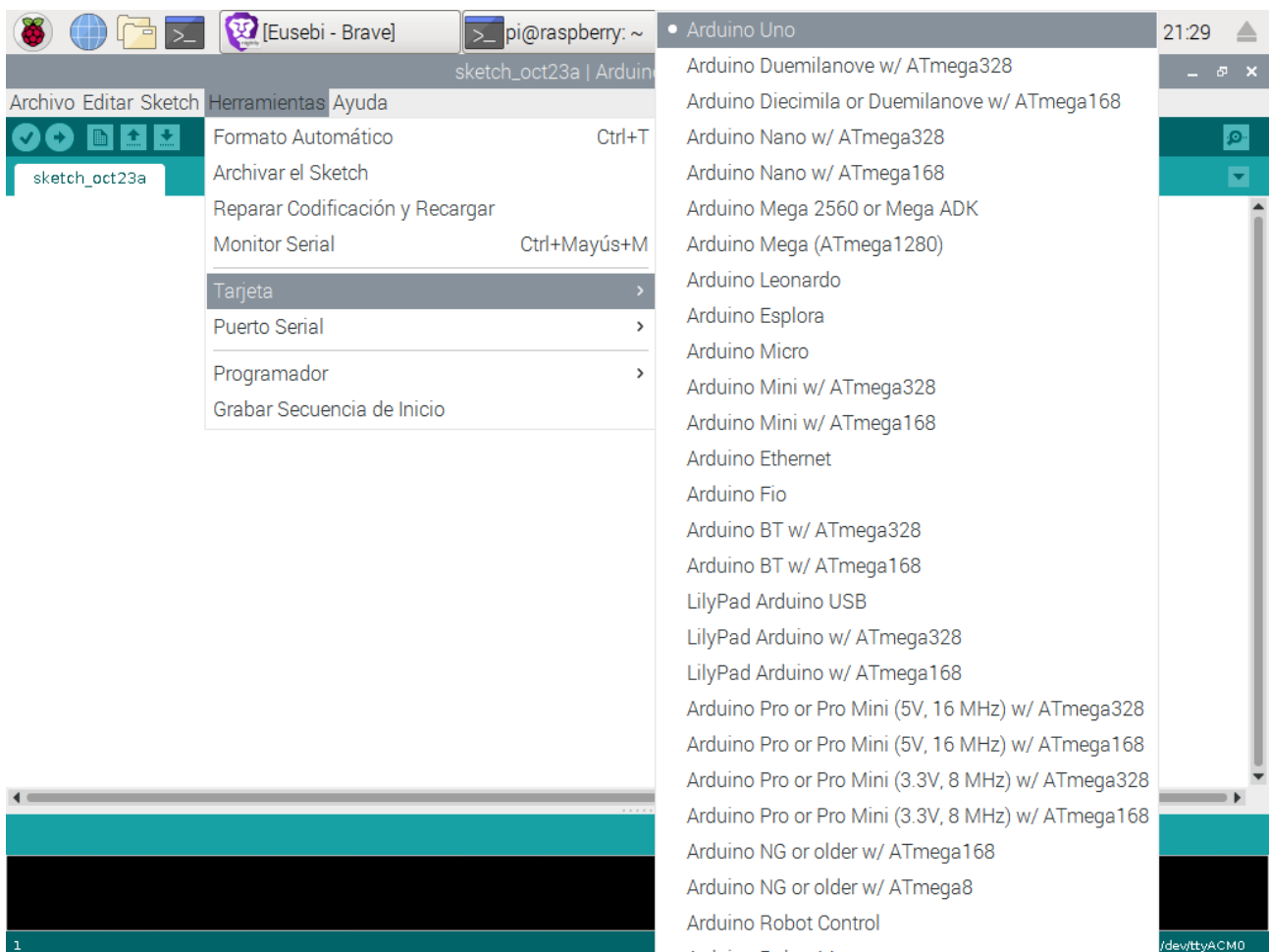
Arduino IDE, software

Sitio oficial → <https://www.arduino.cc/>

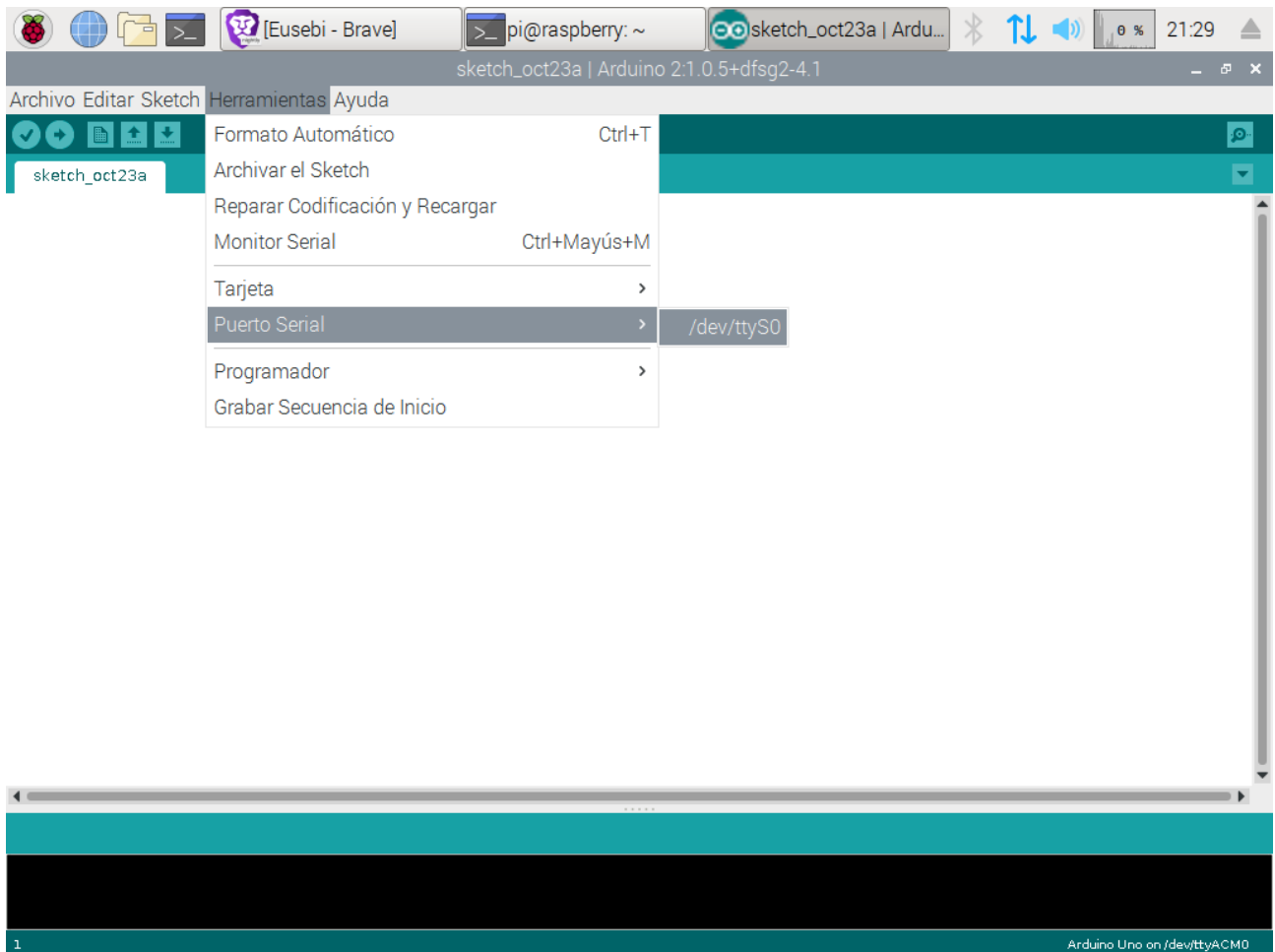
Editor online → <https://create.arduino.cc/editor>

Descargas → <https://www.arduino.cc/en/software> → Just download

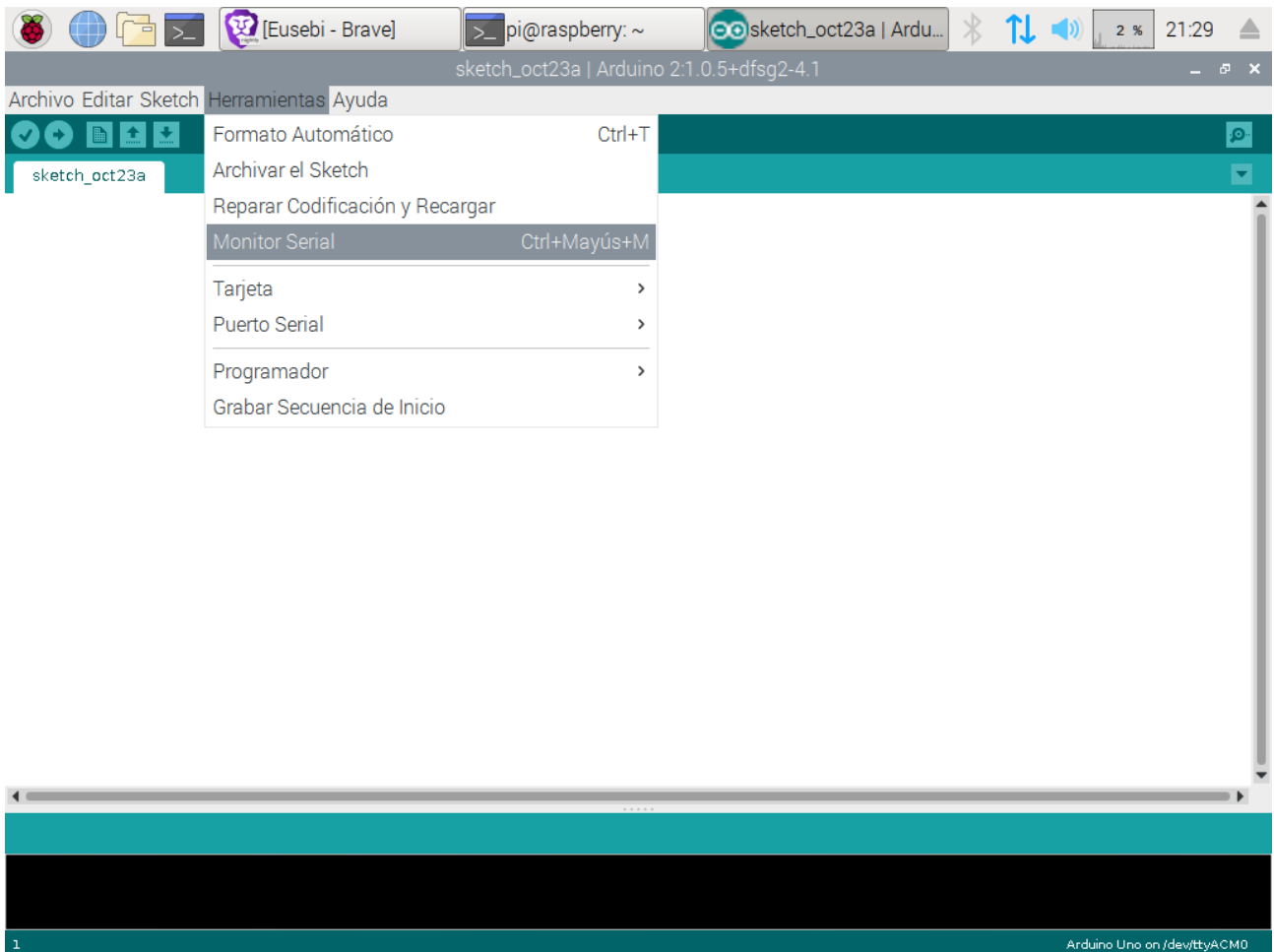
Arduino IDE, Tarjeta



Arduino IDE, Puerto Serial



Arduino IDE, Monitor Serial



Arduino IDE

```
/*
Serial Event example
When new serial data arrives, this sketch adds it to a String.
When a newline is received, the loop prints the string and clears it.
A good test for this is to try it with a GPS receiver that sends out
NMEA 0183 sentences.
NOTE: The serialEvent() feature is not available on the Leonardo, Micro, or
other ATmega32U4 based boards.
created 9 May 2011
by Tom Igoe
This example code is in the public domain.
https://www.arduino.cc/en/Tutorial/BuiltInExamples/SerialEvent
*/
//
// variables
//
String inputString = ""; // a String to hold incoming data
bool stringComplete = false; // whether the string is complete
//
unsigned long cronometro;
bool cronometrar = false, preciso = false, estado; // true
//
// setup
//
void setup() {
  pinMode(LED_BUILTIN,OUTPUT);
  // initialize serial:
  Serial.begin(9600);
  // reserve 200 bytes for the inputString:
  inputString.reserve(200);
  opciones_mostrar();
}
/*
```

```
*/  
//  
// loop  
//  
void loop() {  
  // print the string when a newline arrives:  
  if (stringComplete) {  
    // do something different depending on the range value:  
    Serial.println();  
    switch (inputString.charAt(0)){  
    case '?':  
      cronometro_iniciar();  
      estado_consultar();  
      cronometro_finalizar();  
      if (estado){  
        Serial.println("Encendido.");  
      }else{  
        Serial.println("Apagado.");  
      }  
      cronometro_mostrar();  
      break;  
    case '!':  
      cronometro_iniciar();  
      digitalWrite(LED_BUILTIN,!estado);  
      cronometro_finalizar();  
      Serial.println("Invirtiendo...");  
      cronometro_mostrar();  
      estado_consultar();  
      break;  
    case '+':  
      cronometro_iniciar();  
      estado_consultar();  
      cronometro_finalizar();  
      if (estado){  
        Serial.println("¡Encendido!");  
        cronometro_mostrar();  
      }else{  
        cronometro_iniciar();  
        digitalWrite(LED_BUILTIN,HIGH);  
        cronometro_finalizar();  
        Serial.println("Encendiendo...");  
        cronometro_mostrar();  
        estado_consultar();  
      }  
      break;  
    }  
  }  
  /*
```



```
*/
case '-':
cronometro_iniciar();
estado_consultar();
cronometro_finalizar();
if (estado){
cronometro_iniciar();
digitalWrite(LED_BUILTIN,LOW);
cronometro_finalizar();
Serial.println("Apagando...");
cronometro_mostrar();
estado_consultar();
}else{
Serial.println("¡Apagado!");
cronometro_mostrar();
}
break;
case '*':
cronometrar=!cronometrar;
if(cronometrar){
Serial.print("A");
}else{
Serial.print("Desa");
}
cronometro_iniciar();
Serial.println("ctivado.");
cronometro_finalizar();
cronometro_mostrar();
break;
case '/':
preciso=!preciso;
Serial.print("Mi");
if(preciso){
Serial.print("cro");
}else{
Serial.print("li");
}
cronometro_iniciar();
Serial.println("segundos.");
cronometro_finalizar();
cronometro_mostrar();
break;
/*
```

```
*/
default:
cronometro_iniciar();
Serial.println("Ayuda.");
cronometro_finalizar();
cronometro_mostrar();
Serial.println();
opciones_mostrar();
} // switch
// clear the string:
inputString = "";
stringComplete = false;
}
}
//
// opciones mostrar
//
void opciones_mostrar() {
Serial.println("Opciones");
Serial.print(" ? -> estado ");
if(estado){
Serial.println("encendido");
}else{
Serial.println("apagado");
}
Serial.println(" ! -> invertir");
Serial.println(" + -> encender");
Serial.println(" - -> apagar");
Serial.print(" * -> cronometrar ");
if(!cronometrar){
Serial.print("des");
}
Serial.println("activado");
Serial.print(" / -> precisión mi");
if(preciso){
Serial.print("cro");
}else{
Serial.print("li");
}
Serial.println("segundos");
}
/*
```

```
*/  
//  
// estado consultar  
//  
void estado_consultar() {  
  estado=digitalRead(LED_BUILTIN);  
}  
//  
// cronometro iniciar  
//  
void cronometro_iniciar() {  
  if(cronometrar){  
    if (preciso){  
      cronometro=micros();  
    }else{  
      cronometro=millis();  
    }  
  }  
}  
//  
// cronometro finalizar  
//  
void cronometro_finalizar() {  
  if (preciso){  
    cronometro=micros()-cronometro;  
  }else{  
    cronometro=millis()-cronometro;  
  }  
}  
//  
// cronometro mostrar  
//  
void cronometro_mostrar() {  
  if(cronometrar){  
    Serial.println(cronometro);  
  }  
}  
/*
```

SerialEvent occurs whenever a new data comes in the hardware serial RX. This routine is run between each time loop() runs, so using delay inside loop can delay response. Multiple bytes of data may be available.

```
*/  
void serialEvent() {  
  while (Serial.available()) {  
    // get the new byte:  
    char inChar = (char)Serial.read();  
    // if the incoming character is a newline, set a flag so the main loop can do something  
    about it:  
    if (inChar != '\n') {  
      // add it to the inputString:  
      inputString += inChar;  
    }else{  
      stringComplete = true;  
    }  
  }  
}
```

Consigna

Subir a la página de la EIS, un documento con carátula, una breve explicación de las funciones setup, loop y serialEvent; cuyo nombre sea el/los apellido/s completo/s seguido de la/s inicial/es de su/s nombre/s, en formato .pdf.

Cibergrafía

Microprocesador

<https://es.wikipedia.org/wiki/Microprocesador>

Raspberry Pi

https://es.wikipedia.org/wiki/Raspberry_Pi

Raspberry Pi Modelos

https://es.wikipedia.org/wiki/Raspberry_Pi#Modelos

Microcontrolador

<https://es.wikipedia.org/wiki/Microcontrolador>

Arduino

<https://es.wikipedia.org/wiki/Arduino>

Arduino Tipos y especificaciones

https://es.wikipedia.org/wiki/Arduino#Tipos_y_especificaciones

Arduino Uno

https://es.wikipedia.org/wiki/Arduino_Uno

Arduino UNO R3

<http://store-usa.arduino.cc/products/arduino-uno-rev3>

Tinkercad

<https://www.tinkercad.com/>

Código fuente

https://es.wikipedia.org/wiki/C%C3%B3digo_fuente

Comentarios

[https://en.wikipedia.org/wiki/Comment_\(computer_programming\)](https://en.wikipedia.org/wiki/Comment_(computer_programming))

TODO

<https://es.wikipedia.org/wiki/TODO>

Funciones

<https://es.wikipedia.org/wiki/Subrutina>

setup

[https://es.wikipedia.org/wiki/Configuraci%C3%B3n_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Configuraci%C3%B3n_(inform%C3%A1tica))

loop

[https://es.wikipedia.org/wiki/Bucle_\(programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Bucle_(programaci%C3%B3n))

serialEvent

<https://www.arduino.cc/reference/en/language/functions/communication/serial/serialEvent/>

Arduino IDE

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