

PROSERQUISA^{de G.V.}

EQUIPO DE LABORATORIO DIDÁCTICO

“Excelencia en la experimentación científica”

Curso de Arduino

Tutorial 6: Funciones



Tel.: (503) 2273-2018
Fax: (503) 2273-4770
gerencia@proserquisa.net

Reparto y Calle Los Héroes No. 26-A,
San Salvador, El Salvador, Centroamérica

Tutorial 6: ¿Qué es una función? y como utilizarla

```
int nuestra_función()  
{  
  return 5;  
}
```

Objetivo

En la siguiente lección desarrollaremos el tema de las funciones en Arduino, para poder aplicarlo en las practicas siguientes ya que será necesario el saber utilizarlas para crear nuestros propios programas.

¿Qué es una función?

Es un segmento de código que está separado del código principal del programa, que realiza una tarea específica y regresa a la zona del programa donde fue llamada.

```
void setup() {  
  
    // CODIGO DE LA FUNCION  
  
}  
  
void loop() {  
    // CODIGO DE LA FUNCION  
  
}  
void ejemplo(){  
  
    //CODIGO DE LA FUNCION  
}
```

Luego de declarar las funciones principales void setup y void loop podemos agregar más funciones luego, pero la que Arduino correrá será void loop. En este caso podemos llamar a la función “ejemplo” desde void loop para correrla y que esta ejecute su código interno podría ser una operación matemática como ejemplo.

Utilidades de las funciones.

Ayudan al programador a ser mucho más organizado separando por nombres las funciones haciendo que el código sea legible.

Al utilizar funciones podemos hacer más compacto nuestro código organizándolo por bloques y llamando a las funciones con solo utilizar su nombre para correr su código interno esto nos permite que nuestro código sea más ligero y más fácil de entender.

Las funciones codifican una acción en un lugar solo deben ser depuradas de errores una vez

Declarando funciones

```
}  
int cincuenta_porcientos(int valor){  
  
    int resultado;  
  
    resultado = valor *0.5;  
  
    return resultado;  
  
}
```

La función debe ser declarada fuera de cualquier otra función, por lo tanto puede antes o después de la función loop.

Partes de una función.

Tipo de dato: int, double, void. (Tipo void, si no regresa ningún valor).

Nombre de la función: El nombre no puede llevar espacios puedes usar guion bajo “ejemplo_1” y no puedes utilizar palabras reservadas del lenguaje de programación.

Parámetros: es una variable que envía la función loop, se utiliza para realizar algún cálculo o tarea. Se dejan vacíos si no requiere ningún parámetro.

Código: El código que escribiremos debe estar entre las llaves y cada línea se cierra con punto y coma.

Retorno: “return” es el valor que regresa la función, debe ser del mismo tipo del parámetro que se escribió.

En el ejemplo la función “cincuenta_porcientos” del tipo int, va a calcular el 50% de una determinada cantidad, vamos a recibir un parámetro que es esa cantidad, si suponemos que es 100, en el código de la función tengo un int llamado resultado que será el valor 100 en este caso * 0.5 para calcular el 50% ahora mi variable resultado tendrá un valor de 50, entonces con return devolverá este valor a la función void loop.

Llamar la función.

Para invocar la función en este caso para funciones que regresan un valor. Está compuesta por una variable que va almacenar el valor de retorno, en este caso la variable se llama "res" ahora esta variable vale 50, seguido pondremos el nombre de la función "cincuenta_porcentaje (100)". Básicamente este es el código con el que se manda a llamar una función.

```
int res = cincuenta_porcentaje(100);
```

Declaración de funciones tipo Void.

```
}  
void cincuenta_porcentaje() {  
  
    int resultado;  
  
    resultado = 100 *0.5;  
  
    Serial.print(resultado);  
  
    delay(1000);  
}
```

La manera de declararla es parecida a la anterior, en este caso no quiero que regrese ningún valor usaremos void, luego el nombre de la función y cuando no se necesita que regresen valores los paréntesis se dejan vacíos. Esta función también calcula el 50% de un valor, pero aquí mismo se le asigna y aquí mismo lo muestra, ya que no está recibiendo o enviando valores a la función loop. Declaro mi variable llamada resultado será igual a $100 * 0.5$ que será 50, esto se mostrara en el monitor serial. Por medio de Serial.print, y el delay es para darle un tiempo a la respuesta del cálculo. No regresa ningún valor.

Llamando función loop.

```
cincuenta_por ciento();
```

Es un código más sencillo solo cuenta con el nombre de la función que se desea llamar, y por los parámetros que se le enviarán a la función, en este caso no enviaremos ningún parámetro por lo que los paréntesis se dejan vacíos. Y así de esta manera llamaremos a la función.

Ejemplo

Para poder explicar de una mejor manera pasaremos a ver un ejemplo programando un sencillo sensor de ejemplo, en este tutorial no veremos circuitos o como usarlo físicamente, pero lo usaremos como ejemplo en programación.

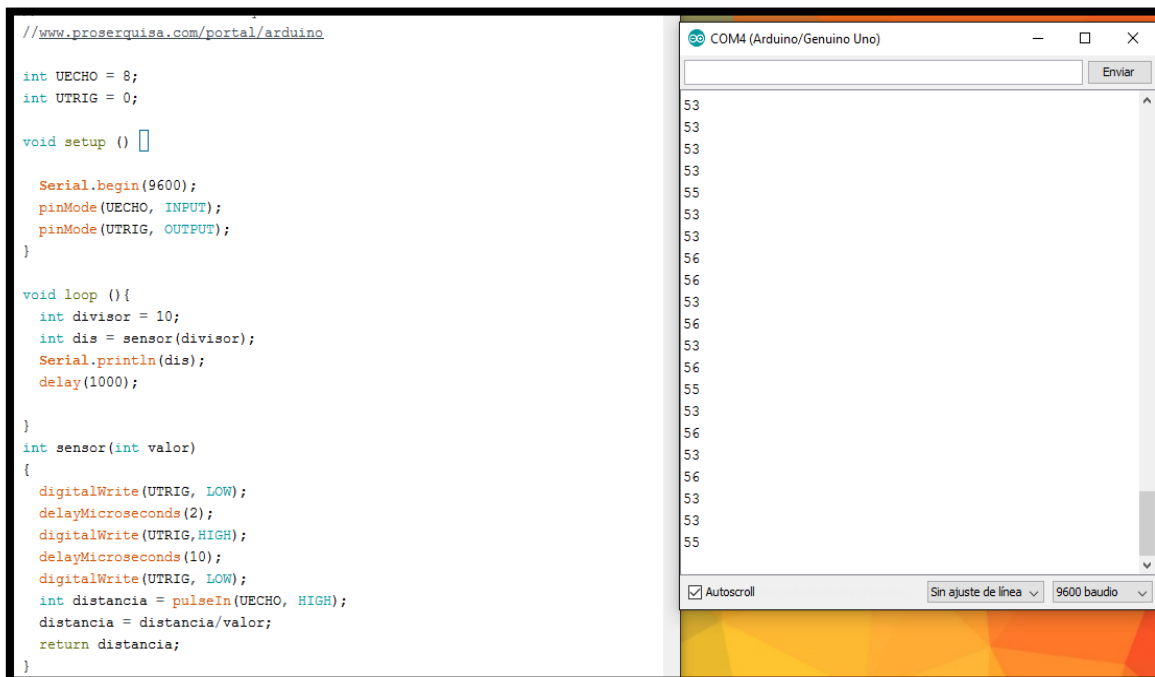
Utilizaremos un sensor ultrasónico de medición para detectar la distancia a la que están los objetos, para poner en prácticas las funciones. Con el siguiente código declaramos el nombre de los pines, y luego en el setup pondremos el modo del pin uno de entrada y el otro de salida, y usaremos la comunicación serial. Posteriormente el void loop, pero nosotros usaremos funciones así que abajo del void loop escribimos nuestra función de tipo int llamada sensor, y recibirá un parámetro de tipo entero llamado valor.

```
leccion6 §  
//Tutoriales Arduino Proserquisa.  
//www.proserquisa.com/portal/arduino  
  
int UECHO = 8;  
int UTRIG = 0;  
  
void setup () {  
  
    Serial.begin(9600);  
    pinMode(UECHO, INPUT);  
    pinMode(UTRIG, OUTPUT);  
}  
void loop  
{  
  
}  
int sensor(int valor){  
  
}
```

Lo que haremos será dentro de la función sensor, copiaremos el código que permite realizar la lectura del sensor, en este tutorial no explicaremos cómo funciona el código, porque la práctica son las funciones, pero luego veremos a detalle como programarlo, así que simplemente copiamos el código que hace funcionar el sensor y luego asignaremos una variable llamada “distancia” que es el valor que manda el sensor, y luego agregamos un valor para convertir a cualquier unidad de longitud este valor lo enviaremos desde el void loop, para finalizar la función pondremos un return llamado distancia que es el valor que la función sensor me va a regresar, ahora que tenemos lista la función procedemos a llamar la función en el void loop.

En el void loop crearemos una variable de tipo entera llamada “divisor” y le pondremos “10” este será el numero en el que se dividirá la distancia para convertir a una unidad, pero el 10 solo es un numero para dividir como ejemplo en este caso. Luego para mandar a llamar la función a una variable le asignaremos el valor de regreso se llamara “dis” será igual al nombre de la función y entre paréntesis pondremos el parámetro que enviara, en este caso “divisor” que vale 100, quedaría así: `int dis= sensor (divisor);`

Luego agregamos un `Serial.println (dis);` para poder leer los valores en el monitor serial y por ultimo agregamos un `delay` para tener una espera de 1 segundo entre la lectura de cada valor.



The image shows a screenshot of an Arduino IDE. The left pane displays the following code:

```
//www.proserquisa.com/porta/arduino

int UECHO = 8;
int UTRIG = 0;

void setup () {

  Serial.begin(9600);
  pinMode(UECHO, INPUT);
  pinMode(UTRIG, OUTPUT);
}

void loop () {
  int divisor = 10;
  int dis = sensor(divisor);
  Serial.println(dis);
  delay(1000);
}

int sensor(int valor)
{
  digitalWrite(UTRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(UTRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(UTRIG, LOW);
  int distancia = pulseIn(UECHO, HIGH);
  distancia = distancia/valor;
  return distancia;
}
```

The right pane shows a serial monitor window titled "COM4 (Arduino/Genuino Uno)". It displays a series of the number "53" being printed to the serial port. The monitor settings at the bottom are set to "Autoscroll", "Sin ajuste de línea", and "9600 baudio".

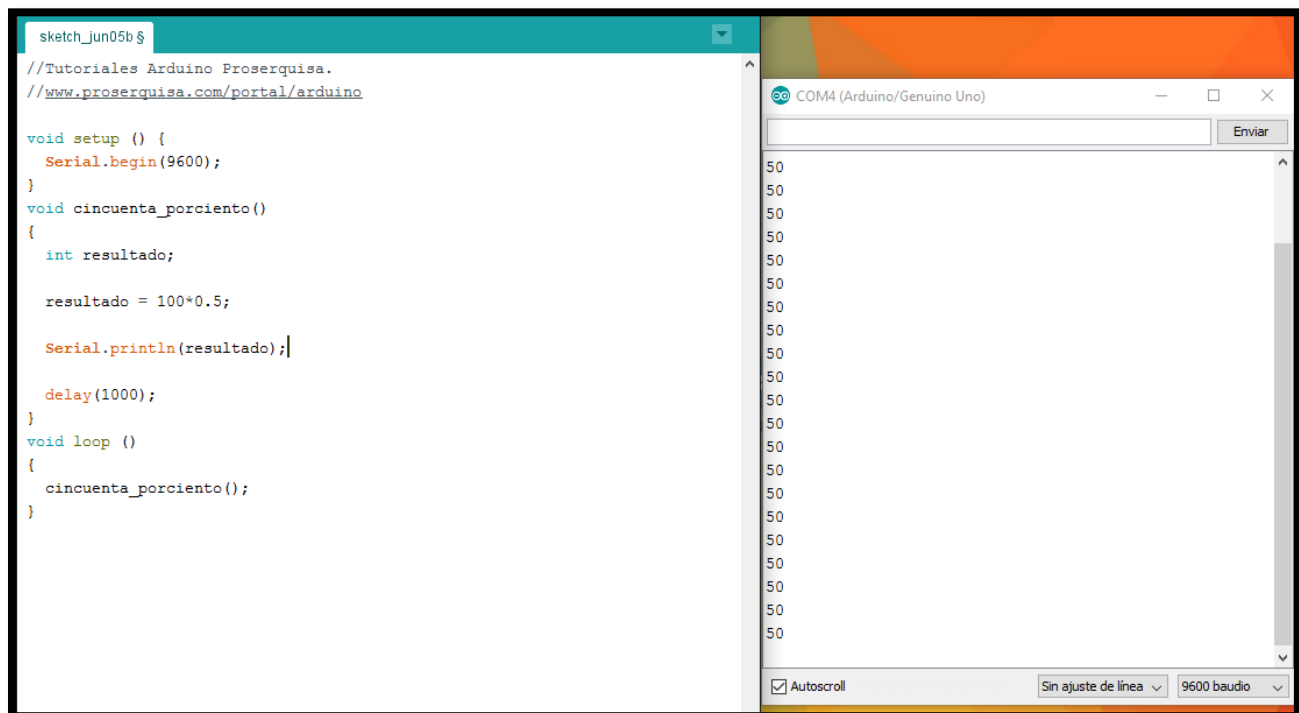
Si quisiéramos medir varias veces la distancia en el código, no hay que estar poniendo todo el código del sensor, simplemente con una línea podemos llamar todo el código.

Ejemplo de funciones Void

En el siguiente ejemplo usaremos el mismo código con el que vimos las partes de una función en el void setup declaramos únicamente la comunicación serial, luego creamos una función void cincuenta_porcentaje() y dentro de ella creamos la variable de tipo entero "resultado" que es igual a 100×0.5 , luego nos escribirá el resultado en el valor serial con un Serial.println y añadimos una espera de 1 segundo entre el resultado.

Por último en el void loop, llamamos a la función simplemente escribiendo el nombre de la función cincuenta_porcentaje(); y es todo.

De esta manera hemos aprendido a utilizar funciones, lo que nos servirá de mucho de aquí en adelante para continuar programando nuestros proyectos.



The screenshot shows the Arduino IDE interface. On the left, the sketch editor displays the following code:

```
sketch_jun05b $
//Tutoriales Arduino Proserquisa.
//www.proserquisa.com/portal/arduino

void setup () {
  Serial.begin(9600);
}
void cincuenta_porcentaje()
{
  int resultado;

  resultado = 100*0.5;

  Serial.println(resultado);

  delay(1000);
}
void loop ()
{
  cincuenta_porcentaje();
}
```

On the right, the serial monitor window is open, showing the output of the sketch. The output consists of a continuous stream of the number "50" on each line, indicating that the function is being called repeatedly in the loop. The serial monitor settings at the bottom show "Autoscroll" checked, "Sin ajuste de línea" selected, and "9600 baudio" selected.

Sección de preguntas.

1 ¿Para qué sirve Return en una función?

2 ¿Puedo crear funciones antes del void loop?

3 ¿Por qué las funciones loop no llevan parámetros en los paréntesis?

1 Sirve para devolver un valor a la función principal loop.

2 Si se puede, se pueden crear antes o después, ya que lo que se ejecutara es la función loop, y en esta se pueden llamar las funciones estén atrás o delante de ella.

3 Porque estas no devuelven ningún valor a loop, solo ejecutan su código interno cuando se les llama.