



GUIA # 04

FUNCIONES Y VECTORES

INTRODUCCION

En la programación, juega un papel importante la simplicidad con la que un código es realizado, entre más simple sea, más óptimo es. Para esto existe una alternativa, esta alternativa permite realizar ciertas rutinas y después llamarlas ya sea en el cuerpo principal del programa o desde otro sitio, esto es lo que conocemos como funciones.

FUNCIONES

Recapitulando, hemos visto como se realiza un programa en arduino, como se utilizan estructuras de selección y estructuras repetitivas, todo esto dentro de la estructura loop, sin embargo, no es la única forma de hacer un llamado, existen las funciones.

Una *función* es grupo de instrucciones con un objetivo particular y que se ejecuta al ser llamada desde otra función o procedimiento. Una función puede llamarse múltiples veces e incluso llamarse a sí misma (función recurrente).

Para entender aún más claro el concepto de función, imagínese una licuadora, usted adiciona agua y fruta, pone las revoluciones de la licuadora y al terminar obtiene un jugo.

Tipos de función

Existen varios tipos de funciones las cuales son:

- *Funciones sin argumento:* Son funciones que no tienen parámetro de entrada, y no devuelven algún tipo de valor.

Ejemplo: Realice un programa que imprima la siguiente información:





Nombre: Juan
Apellido: García
Edad: 34 Años.

En este programa, solo es imprimir esta información, entonces la función que llamaremos *Información*, en lenguaje natural sería:

```
void información(){  
    imprima ('Juan');  
    imprima ('García');  
    imprima ('34');  
}
```

En el cuerpo principal, el llamado de la función sería:

```
INICIO  
    Información();  
FIN
```

- *Funciones con paso de parámetros como valor:* Para este tipo de funciones, se le agrega uno o más parámetros (variables o constantes), a la función, esta no devuelven ningún tipo de dato.

Ejemplo: Supongamos el programa explicado en el ejemplo anterior, ahora bien, para este caso, le pasamos los datos del nombre y apellido variables de tipo Cadena (para arduino el tipo Sting), y la edad de tipo entero (int), entonces la función quedaría:

```
void información(string n, string a, int e){  
    imprima (n);  
    imprima (a);  
    imprima (e);  
}
```

ACLARACION: El nombre de los parámetros no necesitan ser los mismos, ya que las funciones ocupan un lugar diferente en memoria. Como se ve en el cuerpo principal, lo que debe tener en cuenta es la forma en que pasa dichos parámetros:

INICIO

```
Declare Nombre      =      'Juan';  
Declare Apellido    =      'García';  
Declare Edad        =      34;  
Información(Nombre, Apellido, Edad);
```

FIN

- *Funciones que devuelven un valor:* Son funciones que reciben parámetros, pero devuelven algún tipo de valor.

Ejemplo: Realice un programa, que calcule el factorial de un número.

El factorial se define como la multiplicación de todos sus números enteros positivos, es decir, el factorial de 3 sería $1 \times 2 \times 3 = 6$. El factorial de 0 y 1 es 1 (las únicas excepciones del factorial). Para realizar la función Factorial, va a devolver un número entero, y recibe como parámetro un número entero, esta función sería de esta forma:

```
int factorial (int n){  
    int r=1;  
    if(n>=1){  
        for(int i=1; i>=n;i+1){  
            r=r*i;  
        }  
    }else{  
        r=r*1;  
    }  
    return r;  
}
```



En el cuerpo principal, quedaría de la siguiente forma:

```
INICIO
  Declare Numero      =      6;
  Declare Resultado;
  Resultado= factorial(Numero);
FIN
```

Se puede apreciar, que, en este caso, el valor que arroja la función , es guardado en una variable, a diferencia de las anteriores, en las cuales no había necesidad. Para estas funciones siempre hay que tener en cuenta EL TIPO DE DATO QUE NECESITA QUE ESTA FUNCIÓN RETORNE, ya que esto garantiza que la función haga lo que se le solicita.

- *Funciones Librerías:* Son paquetes predefinidos de funciones, que facilitan la programación de ciertos procedimientos. De esto, se hablará en una guía posterior a esta.

Actividad: Realice las siguientes funciones en lenguaje natural:

- ✓ *Función que, dado dos números enteros, me muestre el mayor de ellos.*
- ✓ *Función que, dado un número de 3 cifras me sume sus dígitos ejemplo $123 = 1+2+3 = 6$.*
- ✓ *Función que entregándole dos valores, un porcentaje real y uno de prueba, calcule el margen de error (investigue la formula en internet)*

Forma de declarar funciones

Tomemos como ejemplo el primer código realizado (guía 1), ahora bien, el objetivo es convertirlo en una función la función se declara de la siguiente forma:



En cualquier parte del archivo .ino se va a declarar la función. (para llevar un orden, recomiendo después de la declaración de variables, antes del setup). La forma correcta de declarar una variable es:

```
void encender(int x){
```

Explique qué tipo de función es: _____

Cuantos parámetros recibe: _____

Como primera instancia el tipo de función para la cual requiera la función, para esto tenga en cuenta los tipos de datos, luego el nombre de la función el cual es libre de escoger y por último el (los) tipos de datos, llamados parámetros que requiere para realizar la función en este caso solo necesitamos un dato.

Aclaración: Como se habrá dado cuenta, el setup y el loop son funciones, de tipo *void*.

La función con nombre encender, recibe como parámetro un número entero, que para este caso es un pin, y la función debe encender y apagar el led. Entonces la función quedaría así:

```
void encender(int x){  
  digitalWrite(x,HIGH);  
  delay(5000);  
  digitalWrite(x,LOW);  
  delay(5000);  
}
```

Ahora bien, en el loop, la llamada de la función es:

```
void loop() {  
  encender(led);  
}
```



Tenga en cuenta, que el nombre del parámetro de la función no tiene que ser obligatoriamente igual al parámetro del llamado (loop), puede variar a capricho del programador.

Actividad

- a. Realice la prueba de este código, realice el montaje solo con un led conectado en el pin 13 (ya que es un puerto de prueba, y al lado hay una conexión a tierra). Pruébelo y compare con la versión anterior.*
- b. Realice el montaje del semáforo, declare las funciones que encienda los leds con forme está en el problema de la guía 01. Muestrelos a su tutor.*



ACTIVIDAD LUDICA

Problema 01: EL banco nacional solicita una alarma de movimiento, en la cual se encienda un color verde si no detecta movimiento, y un tono rojo y suene una alarma.

(realice los procedimientos para la elaboración de problemas, utilice vectores para declarar los pines del led funciones para cada una de las acciones. Enséñele a su tutor el pseudocódigo antes de realizar el montaje).

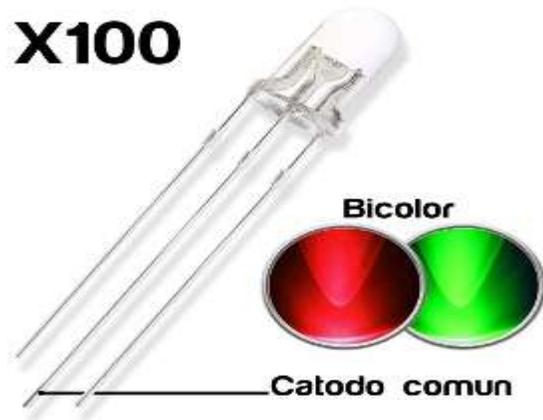
Montaje

Materiales

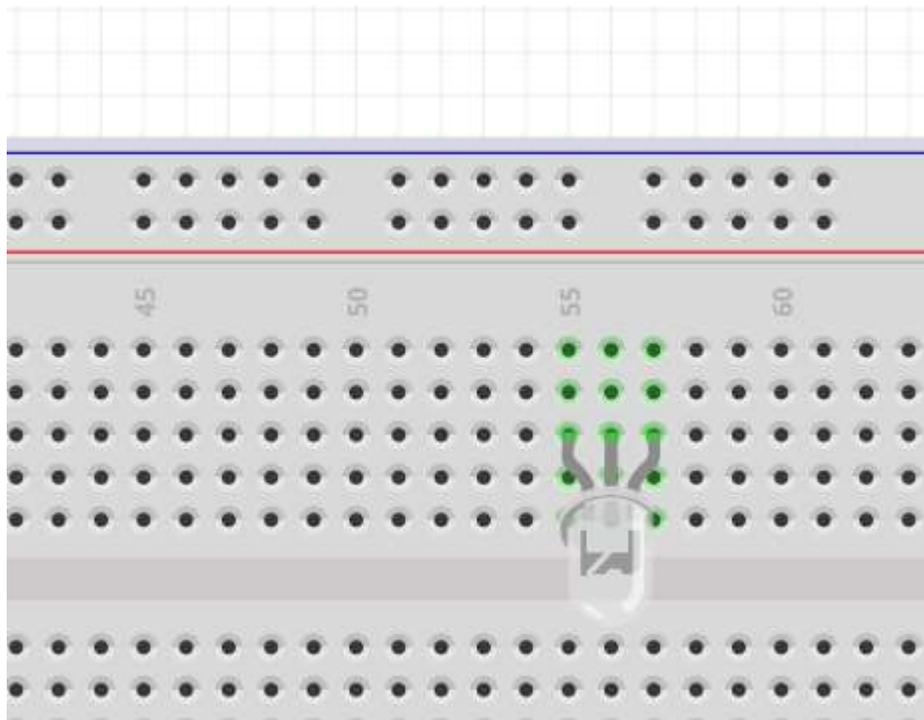
- 1 Arduino one
- 1 Protoboard
- 1 zumbador
- 1 resistencia de 220 OHM
- 6 cables conexión M/M
- 3 cables conexión M/H
- 1 sensor de movimiento HCSR501
- 1 LED bicolor

Paso 1: montaje del led Bicolor

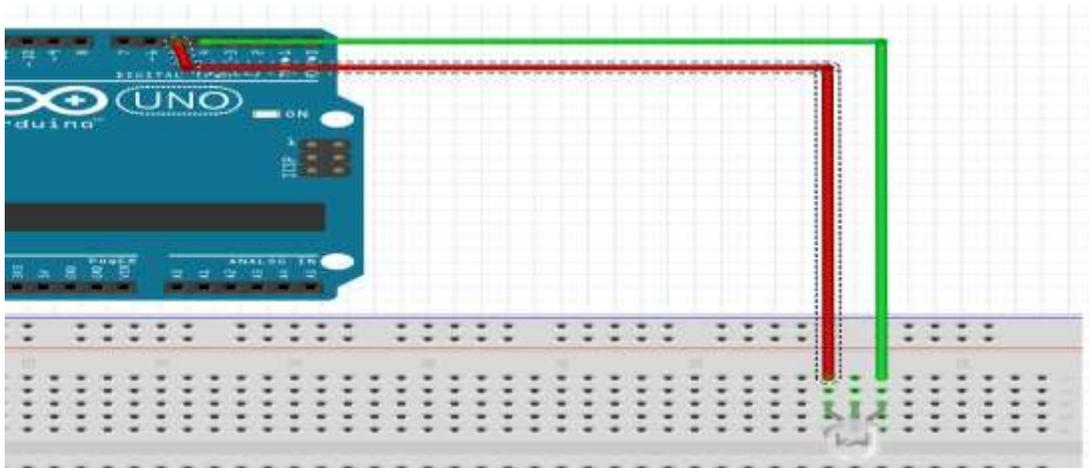
Un led bicolor es un led que enciende de dos colores, verde y rojo, a diferencia de los led utilizados anteriormente, este su cátodo está ubicado en el centro, y su pata más corta, representa un color, y la otra larga otro color.



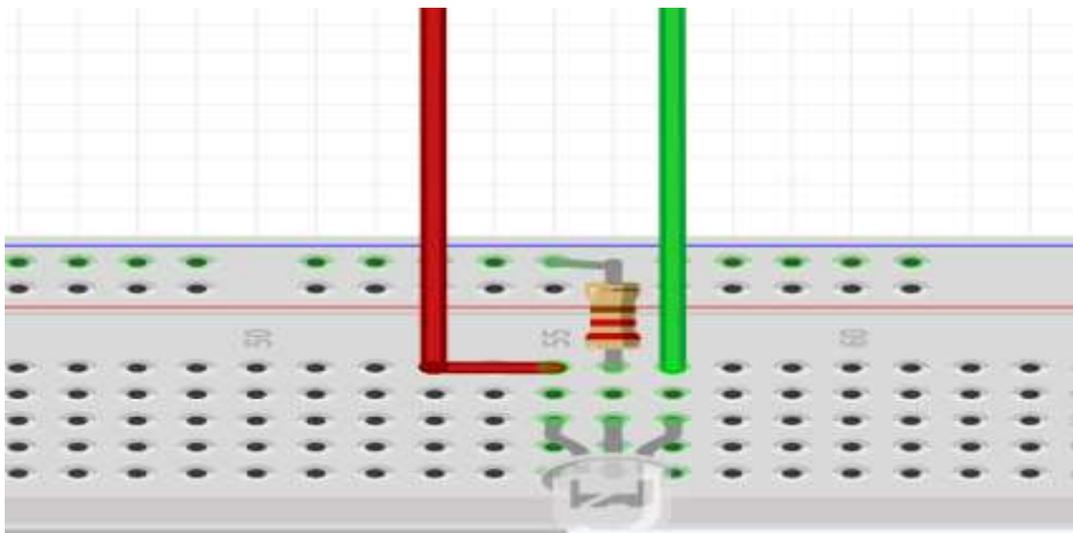
Como primer paso, ubique el LED en la protoboard.



Luego conecte dos cables, a los ánodos (extremos) hasta el arduino, utilice los pines que quiera.



Conecte una resistencia de 220 OHM desde el cátodo (pata central) a la línea negativa de la protoboard y alimente la protoboard.



Realice las funciones `encender_led`, que reciba como parámetros el pin del led y el tiempo de duración y `apagar_led` que reciba el parámetro el pin del led. utilícelas dentro del loop. Muéstrele a su tutor el funcionamiento antes de seguir con el montaje.

Paso 2: Montaje del zumbador

Realice el montaje del zumbador que se realizó en la práctica 2, añada la función sonido, modifique el loop de la siguiente forma:

- Encender verde
- Sonar
- Apagar verde
- Encender rojo
- Sonar
- Apagar rojo.

Enséñele a su tutor el código, el funcionamiento del montaje

paso 3: montaje del sensor de movimiento.

Anteriormente se hizo un montaje con el sensor de ultrasonido, para esa práctica se va a realizar con otro tipo de sensor, este sensor se denomina de detección de movimiento o sensor de presencia PIR.



1 Sensor PIR HCSR501

En los sensores de movimiento, el sensor PIR consta en realidad de 2 elementos detectores separados, siendo la señal diferencial entre ambos la que permite activar la alarma de movimiento. En el caso del HC-SR501, la señal generada por el sensor

ingresa al circuito integrado BISS0001, el cual contiene amplificadores operacionales e interfaces electrónicas adicionales.

Las funciones y ajustes complementarios del sensor de movimiento son:

- *Ajuste de parámetros: mediante 2 potenciómetros, el usuario puede modificar tanto la sensibilidad como la distancia de detección del PIR.*
- *Detección automática de luz (esta función no está disponible al adquirir el sensor de fábrica): por medio de una foto resistencia CdS (Sulfuro de Cadmio), se deshabilita la operación del sensor en caso que exista suficiente luz visible en el área. Esta función es utilizada en caso de sensores que enciendan lámparas en lugares poco iluminados durante la noche, y especialmente en corredores ó escaleras.*

Como se indicó anteriormente, el rango de detección de movimiento de los PIR es ajustable y generalmente funcionan con alcances de hasta 7 metros, y con aperturas de 90° a 110°, como se muestra en la figura. El montaje del PIR puede realizarse tanto en piso, muro o techo, según convenga a la aplicación.



2 Vista de componentes del sensor PIR

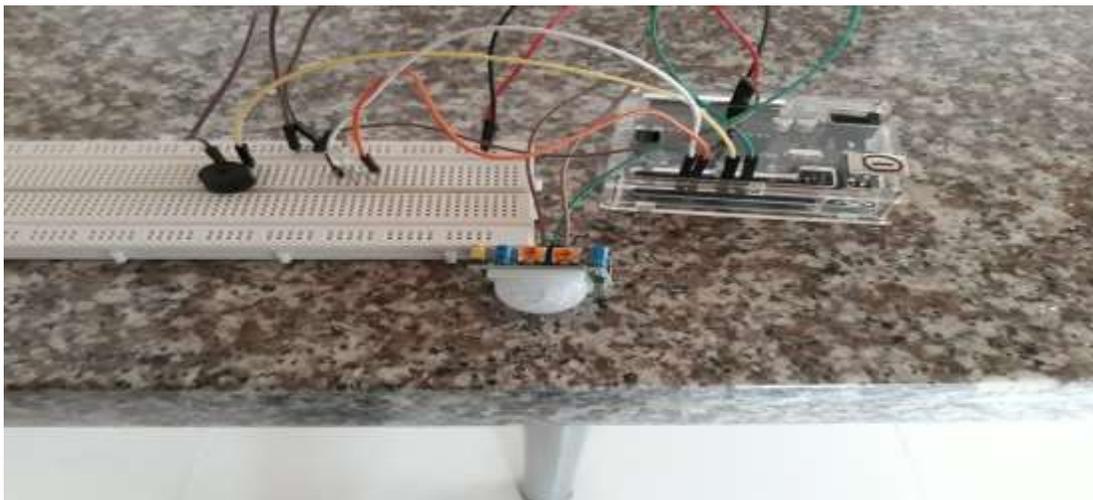
Este componente no se recomienda conectarlo directamente a la protoboard, para esto se utilizaran los cables M/H



3Cables de conexión M/H

Para la conexión del sensor, conecte los cables de alimentación M/H del sensor a la protoboard, los extremos a la fuente de alimentación de la protoboard (guíese por la figura 2 para identificar positivo y negativo) y el central a un pin digital del arduino.

Su montaje debe quedar de la siguiente forma:



Realice el código, pruebe su montaje y compárelo con el montaje del ultrasonido. ¿Cuál es más fiable para este caso y por qué?



Vectores

Un vector (array) es un arreglo unidimensional, en el cual se almacena un tipo único de datos y tiene un nombre común para todos los datos. Es una forma óptima para manejar una gran cantidad de datos. La representación de los vectores es:

$x[m]=\{\text{dato 1,dato 2,dato 3, dato n}\};$

donde m = longitud del vector, y n es el enésimo dato que hay en el vector.

Otra forma de representar un vector es:

$X=\{\text{dato 1, dato 2, dato 3, ..., dato n}\}.$

En esta forma no hay necesidad declarar la longitud del vector, esta es recomendada utilizarla siempre y cuando conozca todos los datos que se van a almacenar en el array.

Si se desea conocer el dato específico en un vector, se debe especificar la posición. Las posiciones de un vector comienzan en el número 0, hasta el último dato, por ejemplo:

$x=\{\text{Juan,José,Mariá,Pedro,Carlos,Martha,Camilo,Andres}\}$

entonces, la posición que ocupa cada dato dentro del vector es como se muestra en la siguiente tabla:

$x=$

Juan	José	Mariá	Pedro	Martha	Camilo	Andrés
0	1	2	3	4	5	6

La posición de un vector se determina por la ecuación $n-1$, como en este ejemplo, cuya longitud es de 7, su última posición es 6 ($n=7$; $7-1=6$).

Declaración de un vector en arduino





Un vector en arduino, se declara de la siguiente forma:

Int x[n];

Int x[]={dato 1, dato 2, dato 3, ..., dato n}

Tenga en cuenta que el tipo de dato (en este caso el int o entero) puede variar de acuerdo al tipo de dato que se requiera, y n representa la longitud de los datos que se ingresan:

Actividad

- a. *En lenguaje natural, realice las instrucciones para los siguientes casos:*
 - *Imprima un vector de datos.*
 - *Almacena los 5 primeros múltiplos de 3*
 - *Ordene los datos de un vector numérico de menor a mayor.*

- b. *Realice los montajes del semáforo (guía1) y el auto fantástico (guía 3), convierta la declaración de los pines en un arreglo. Enséñeselo a su tutor y explique por qué es una forma de optimización del código.*

- c. *Investigue que es una matriz, cual es su funcionalidad y su declaración en programación. El grupo debe realizar una breve exposición la siguiente sesión de máximo 2 minutos.*



AUTOEVALUACION

Coloque una X en las siguientes preguntas

1. Con el desarrollo de esta guía reafirmó el concepto de función.
SI ____ NO ____
2. Con lo desarrollado en esta guía, es capaz de diferenciar los tipos de funciones empleados en programación.
SI ____ NO ____
3. Con el desarrollo de esta guía reafirmó el concepto de vectores.
SI ____ NO ____
4. Con las guías realizadas anteriormente, ha fortalecido los conceptos que posee de programación
SI ____ NO ____
5. Con el desarrollo de esta guía ha comprendido aún más la programación de la plataforma arduino.
SI ____ NO ____