

“MANEJO DE ESTRUCTURAS Y SUBRUTINAS EN ARDUINO”

Informe N°4

Laboratorio de Sistemas Embebidos

Melanny Dávila

Ingeniería en Telecomunicaciones
Facultad de Eléctrica y Electrónica
Quito, Ecuador
melanny.davila@epn.edu.ec

2nd Jonathan Álvarez

Ingeniería en Telecomunicaciones
Facultad de Eléctrica y Electrónica
Quito, Ecuador
jonathan.alvarez@epn.edu.ec

Abstract—Las funciones y subrutinas son de gran importancia a medida que los bloques de código empleados se vuelven mas extensos y complicados. En el presente informe se tratará acerca de sus ventajas, Las diferencias que presentan con las librerías existentes en Arduino, el concepto de sobrecarga y se tratará un ejemplo práctico de automatización usando el sensor HC-SR04.

Index Terms—Arduino, sketch, subrutina, función, estructura.

I. INTRODUCCIÓN

Si tenemos un conjunto de instrucciones que hemos de escribir repetidas veces en diferentes partes de nuestro sketch la creación de funciones es una forma de mantener nuestro código compacto. Una función es un trozo de código al que se le identifica con un nombre. De esta forma, se puede ejecutar todo el código incluido dentro de ella simplemente escribiendo su nombre en el lugar deseado de nuestro sketch. Al crear nuestras propias funciones escribimos código mucho más legible y fácil de mantener. Segmentar el código en diferentes funciones permite crear piezas modulares de código que realizan una tarea definida. Además, una función la podemos reutilizar en otro programa, de manera que con el tiempo podemos tener una colección muy completa de funciones que nos permitan escribir código muy rápida y eficientemente.

II. OBJETIVOS

- Relacionar al estudiante con el uso y manejo de las diferentes estructuras y subrutinas de control.
- Diseñar e implementar segmentos de código que permitan al estudiante explorar esquemas de automatización [2].

III. CUESTIONARIO

A. Consultar las ventajas del uso de funciones en Arduino.

Las principales ventajas del uso de funciones en los sketches de Arduino son:

- Ayuda a tener organizado el código.
- Es más sencillo reutilizar código en otros sketches.
- Aumenta la legibilidad del código [3].
- Reduce la probabilidad de errores al modificar el código.

- El tamaño del sketch sea menor porque el código de la función es reutilizado.
- Reducción considerable del tamaño del sketch porque el código de la función es reutilizado [4].
- Una función codifica una tarea en un lugar del sketch, así que la función solo debe ser pensada una sola vez.
- La modificación se vuelve más fácil con la ayuda de funciones en un sketch [3].

B. Consultar las diferencias entre librerías y funciones en Arduino.

- **Librerías:** Son funciones integradas, es decir, están pre-definidas en la biblioteca de Arduino. Se utilizan para realizar las operaciones más comunes como cálculos, actualización, etc. Algunas de las funciones de la biblioteca o librería son printf, scanf, sqrt, etc. Para usar estas funciones en el programa, el usuario tiene que usar el archivo de encabezado asociado a la función correspondiente en el programa [4].
 - Estas funciones no las crea el usuario como propias.
 - Las funciones de librería se almacenan en un archivo de biblioteca especial. En esto, si el usuario desea usar una función de biblioteca en particular, entonces el usuario debe agregar la biblioteca particular de esa función en el archivo de encabezado del programa.
 - La ejecución del programa no comienza desde la función de biblioteca [4].
- **Funciones:** Las funciones permiten a un programador dividir un código específico en varias secciones, y cada sección realiza una tarea en particular. Las funciones se crean para realizar una tarea varias veces en un programa. La función es un tipo de procedimiento que devuelve el área de código desde la que se llama [5].
 - Las funciones que se van a crear no están pre-definidas en el compilador.
 - Estas funciones son creadas por el usuario según sus propios requisitos.
 - Las funciones definidas por el usuario no se almacenan en el archivo de biblioteca [4].

- No existe tal tipo de requisito para agregar una biblioteca en particular.
- La ejecución del programa comienza desde la función definida por el usuario [5].

El Arduino tiene dos funciones comunes `setup()` y `loop()`, que se llaman automáticamente en segundo plano. El código a ejecutar está escrito entre llaves dentro de estas funciones.

- 1) `void setup()`: incluye la parte inicial del código, que se ejecuta solo una vez. Se llama bloque de preparación.
- 2) `void loop()`: incluye las declaraciones, que se ejecutan repetidamente. Se llama bloque de ejecución [5].

C. Consultar a que se hace referencia el término *function Overloading* o *sobrecarga de funciones* en Arduino.

A esta capacidad de definir varias funciones diferentes con el mismo nombre, se le llama *Function Overloading* o *sobrecarga de funciones* y dado que cada una de las funciones tienen igual nombre se dice que están *sobrecargadas* (*Overloaded*) [6].

D. En el circuito 5.4 utilizar un sensor HC-SR04 para obtener el nivel de agua del tanque, considerando que tiene una profundidad de 1m. Existe un sensor localizado en la parte superior del mismo, es decir que el sensor mide la distancia desde su ubicación al nivel de agua. Además, el circuito presenta una llave para controlar el ingreso de agua, simulada mediante un relé. El circuito debe responder a las especificaciones de la tabla 1.

Umbral	Estado	Distancia	Led	Llave
1	Tanque lleno	desde 0 cm a 10 cm	Encender led verde	Cerrar llave (relé)
2	Tanque a la mitad	desde 10 cm a 80 cm	Encender led amarillo	Abrir llave (relé)
3	Tanque vacío	desde 81 cm en adelante	Encender led rojo	Abrir llave (relé)

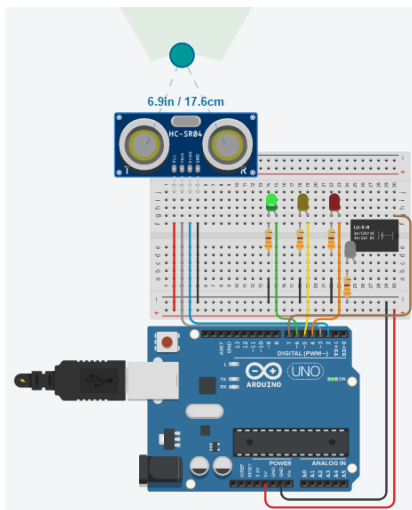


Fig. 1. Control de nivel del tanque con tanque lleno

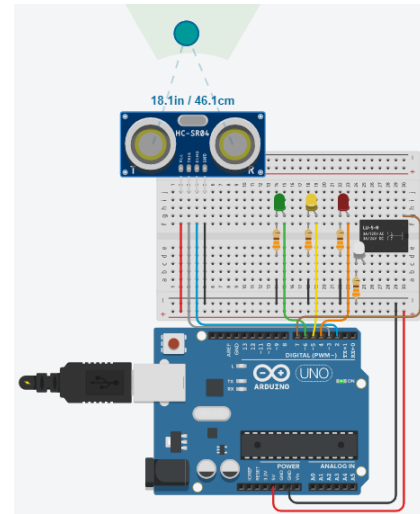


Fig. 2. Control de nivel del tanque con tanque a la mitad

```

1  /* Informe 4
2  Melanny Davila, Jonathan Alvarez
3  Usando el sensor HC-SR04 obtener el nivel de
4  agua en un tanque
5  Se representa por los siguientes valores:
6  Tanque lleno de 0 cm a 10 cm
7  Tanque a la mitad desde 11 cm a 80 cm y
8  Tanque vacío Desde 81 cm en adelante
9  Se encienden los LEDs verde, amarillo y rojo
10 dependiendo
11 del caso respectivamente. El rele se abrir
12 para las dos ultimas opciones
13 y cuando est lleno permanecerá cerrado
14 El rele sera representado por la luz blanca que
15 se enciende o se apaga
16 */
17 // Se inicializan variables
18
19 int cm = 0;
20 int triggerPin = 3;
21 int echoPin = 2;
22 float prof; // Variable que controla la
23 profundidad
24
25 void setup()
26 {
27     //Definicion de Pines
28     pinMode(4, OUTPUT);
29     pinMode(5, OUTPUT);
30     pinMode(6, OUTPUT);
31     pinMode(7, OUTPUT); // pin que controla el
32     rele
33 }
34 long readUltrasonicDistance(int triggerPin, int
35 echoPin){
36     pinMode(triggerPin, OUTPUT); // Define el pin
37     trigger
38     // Retardos para la medicion en el sensor
39     digitalWrite(triggerPin, LOW);
40     delayMicroseconds(2);
41     digitalWrite(triggerPin, HIGH);
42     delayMicroseconds(10);
43     digitalWrite(triggerPin, LOW);
44     pinMode(echoPin, INPUT);
45     return pulseIn(echoPin, HIGH); //El valor de la
46     funci n se ajusta para calcular
47 }

```

```

41 void loop(){
42   digitalWrite(4, LOW);
43   digitalWrite(5, LOW);
44   digitalWrite(6, LOW);
45
46   // Valor en cm ajustado
47   cm = 0.01723 * readUltrasonicDistance(
48     triggerPin ,echoPin);
49
50   // Rango de valores para cada estado
51   if (cm>=0 && cm<=40){
52     digitalWrite(6, HIGH);
53     digitalWrite(7, LOW);
54   }
55
56   if (cm>=41 && cm<=80){
57     digitalWrite(5, HIGH);
58     digitalWrite(7, HIGH);
59   }
60
61   if (cm>=81){
62     digitalWrite(4, HIGH);
63     digitalWrite(7, HIGH);
64   }
65   delay(100); // Retraso del programa
66 }

```

E. Conclusiones:

Jonathan Álvarez

- Las funciones nos permiten tener un código organizado ya que al realizar cierta acción nos puede referir a una parte del código donde nuestra función se encontrara en el lugar destinado para las mismas.
- Las funciones nos pueden ayudar a evitar errores por la repetición de código, ya que con la función solamente debemos escribirla una vez y esta es llamada a medida que se la necesite. Si se escriben las acciones de la función repetidas veces en el código en lugar de hacerlo de manera separada se puede cometer errores
- Las estructuras de control y condiciones nos permiten brindar a nuestros programas la capacidad de interactuar con variables de manera que se pueda determinar salidas dependientes de estos.

Melanny Dávila

- En muchas implementaciones es vital el uso de las funciones que permiten medir el tiempo con el fin de garantizar un correcto funcionamiento de la aplicación realizada.
- Una estructura de control permite implementar la toma de decisiones mientras el programa está siendo ejecutado, lo que permite aún una mejor interacción entre el usuario y la aplicación.
- El uso de subrutinas facilita una considerable reducción de las líneas de código dentro de un sketch; lo que en muchos caso puede disminuye el esfuerzo que debe realizarse dentro del mismo.

F. Recomendaciones:

Jonathan Álvarez

- Mantener una lista de funciones comunes debido a que hay acciones que se pueden requerir en múltiples aplicaciones y debido a la modularidad de las funciones se pueden adaptar al nuevo programa en el cual se está trabajando
- Colocar de ser necesario el tipo de valor que la función devolviera, en caso de que no exista retorno se puede usar void.

Melanny Dávila

- Se recomienda la creación de funciones siempre y cuando sea necesario y su uso sea constante con el fin de evitar tener un código cuya longitud es bastante larga.
- Analizar de forma minuciosa que tipo de dato devolverá una función, con el fin de evitar errores en el desarrollo del código.
- Se recomienda realizar el uso de estructuras de control de una forma ordenada y eficiente dentro del sketch.

REFERENCES

- [1] O. Lira, S. Hernández, y R. García, Manual de Programación Android-Arduino: Principios básicos de la programación móvil aplicados a entornos interactivos. Editorial Académica Española. OmniScriptum Publishing Group, 2017.
- [2] E. Tatayo, "MANEJO DE ESTRUCTURAS Y SUBROUTINAS EN ARDUINO". C.P. SISTEMAS EMBEBIDOS, Accedido: dic. 23, 2020. [En línea].
- [3] "Funciones definidas por usuario", Aprendiendo Arduino, nov. 16, 2016. <https://aprendiendoarduino.wordpress.com/2016/11/16/funciones-definidas-por-usuario-2/> (accedido dic. 23, 2020).
- [4] "Difference between user defined function and library function", Geeks-forGeeks, jun. 20, 2020. <https://www.geeksforgeeks.org/difference-between-user-defined-function-and-library-function-in-c-c/> (accedido dic. 24, 2020).
- [5] "Arduino Functions - JavaTpoint", [www.javatpoint.com](https://www.javatpoint.com/arduino-functions), <https://www.javatpoint.com/arduino-functions> (accedido dic. 24, 2020).
- [6] "Polimorfismo y Function Overloading — Tienda y Tutoriales Arduino". <https://www.prometec.net/polimorfismo-y-overloading/> (accedido dic. 24, 2020).