Universidad Internacional de las Américas

Escuela de Ingeniería Informática.

Carrera:

Ingeniería de Software

Tema:

Arduino

Estudiante:

Christian Fuentes Oviedo

Profesor:

Ing. Leonardo Delgado Arroyo

San José, Costa Rica

Abril, 2017

# Programación en Arduino.

En este pequeño documento se planea explicar algunas funciones con respecto al lenguaje de programación, es por ello que se asume que aquellos estudiantes que lean dicho documento han concluido satisfactoriamente el curso de programación I de la UIA, por tanto, contarán con un nivel básico en Java, y se usará como leguaje guía y se realizarán similitudes con el fin de comprender su programación.

El IDE de Arduino, se encuentra basado en el lenguaje de programación C++, sin embargo, como ya se mencionó la similitud con Java es suficiente para un manejo optimo del Arduino, es por ello que no se mencionaran referencias a C++, aquellos que quieran utilizar C++ como lenguaje de programación para el Arduino, es posible importando unas librerías y utilizando Visual Studio Code, el cual es Open Source, como plataforma de desarrollo.

## Declaración de variables.

Es posible declarar variables tanto globales como locales, las globales son aquellas que es posible utilizarse desde cualquier parte del código mientras que las segundas son solo para trabajar de manera interna en los métodos.

Estas variables además pueden establecerse un pin que enviara datos:

pinMode (3, OUTPUT);

O un pin que recibirá datos:

pinMode(9,INPUT);

Los pines pueden adoptar un nombre, esto con el fin de ayudar en su programación:

**int** Contor\_In1 **=** 7;

En el caso anterior el pin numero 7 será de tipo entero y se llamará Control\_In1

## Uso de Métodos.

De igual manera que la programación común en Java, en el arduino es posible trabajar utilizando métodos, entendiendo por método bloques o secciones de código que permiten realizar una determinada función, por ejemplo, mover un motor, encender un led, entre muchos otros. Esto permite tener un código más limpio, así como una modularización, permitiendo llamar a los métodos en el momento que sea necesario.

## IDE Arduino

Al abrir el IDE de Arduino lo primero que salta a la vista es que contiene por defecto dos métodos, setup () y loop (). Estos métodos son de suma importancia en la funcionalidad que tomará el arduino en momento de ejecución, el primero permite establecer como funcionaran las variables, salidas o entradas de datos dependiendo de la necesidad, la declaración de estas ocurre como se mencionó en el apartado de **uso de variables.** El segundo método mencionado es un bucle infinito, es decir se repetirá constantemente mientras el arduino esté en funcionamiento. Se puede decir que es el método principal, como lo es el Main para Java, es aquí donde inicia la ejecución del código, este ciclo le permite al Arduino ejecutar sus funcionamientos una y otra vez. Dentro de este ciclo pueden ejecutarse métodos, condicionales, variables. Se dice que es infinito ya que una vez llegue a la llave de cierre volverá a empezar su ejecución desde el inicio del mismo.

**void** loop(){

  digitalWrite(8,LOW);

  delay(1000);

  digitalWrite(8, HIGH);

  delay(10000);

}

En el ejemplo anterior se muestra como en el método principal el pin 8 se apaga teniendo una duración de un segundo posteriormente es encendido la misma cantidad de tiempo. En un método común la ejecución terminaría, sin embargo, al ser el loop(), el ciclo se repite pagando nuevamente el pin. La palabra reservada delay permite a un pin ejecutar lo que se le pida por un rango de tiempo determinado siendo 1 = 10-6 es decir, 1 X 1000 equivale a 1 segundo. Si se desea trabajar con microsegundos de más de dos dígitos bastara con poner la palabra reservada Microseconds, ejemplo:

delayMicroseconds(10);

## Condicionales:

El uso de condicionales que alteran la ejecución del código como lo es if o for, cumplen una sintaxis y funcionamiento igual a Java por lo que su comprensión es inmediata, utilizar un for para apagar y encender un mismo Led es una manera óptima de trabajarlo:

**void** CicloEjemplo(){

**for**(**int** i **=** 0 ; i**<=** 10 ; i**++**){

      digitalWrite (12, HIGH);

      delay(200);

      digitalWrite (12, LOW);

      delay(200);

  }

## Llamadas:

Es buena práctica en programación crear bloques de código que cumplan determinadas funciones, como ya se mencionó en el apartado de métodos. Sin embargo, se debe comprender que dichos métodos no podrán cumplirse o ejecutarse si no son llamados desde el loop () o algún otro método creando una cadena de métodos, todo en función de un funcionamiento óptimo. Como manera de ejemplo usaremos el mismo ejemplo donde se encendía y apagaba el pin 8, suponiendo que representa una luz led:

**void** loop(){

  Ejemplo();

}

**void** Ejemplo(){

  digitalWrite(8,LOW);

  delay(1000);

  digitalWrite(8, HIGH);

  delay(10000);

}

La ejecución será la misma a la explicada anteriormente ejecutando primeramente el loop(), sin embargo, la primera línea de ejecución llama al método: Ejemplo(); por tanto, el orden de ejecución se altera, leyendo como siguiente línea de código lo que contenga dicho método.

Una vez comprendidos estos conceptos, es posible comprender que, al confeccionar un robot en Arduino, a la hora de implementar la lógica de funcionamiento, es posible implementar multitud de métodos y llamarlos según sea necesario, además pueden ser utilizados multitud de veces lo que simplifica el código. Por ejemplo, para el desplazamiento de un vehículo se pueden crear diversos métodos que controlen su movimiento, acelerar, frenar, doblar, entre otros según se necesiten.

**void** Run(){

  digitalWrite(Contor\_In1, HIGH);

  digitalWrite(Contor\_In2, LOW);

  digitalWrite(Contor\_In3, LOW);

  digitalWrite(Contor\_In4, HIGH);

}*//FIN DEL METODO RUN*

**void** Stop(){

  digitalWrite(Contor\_In1, LOW);

  digitalWrite(Contor\_In2, LOW);

  digitalWrite(Contor\_In3, LOW);

  digitalWrite(Contor\_In4, LOW);

}*//FIN DEL METODO STOP*

En ambos ejemplos se está utilizando un puente H para controlar dos motores y desplazar un vehículo, si se desea comprender mejor el funcionamiento a nivel de hardware en el documento escrito del proyecto final se encuentran los detalles.

Como se ha podido conocer, la programación en Arduino es sencilla, se debe tomar en cuenta que en este documento no se habla de la sección de hardware, un funcionamiento óptimo se software conlleva una estructura correcta de hardware.

Para más información, descargar este y/o el documento escrito del proyecto final, conocer los principales módulos utilizados o el código fuente del robot, puede acceder a mi git hub personal: <https://github.com/lKanade/Robot.Car.with.Arduino>