Modulo_1:

Variables, Arrays y bucle For

FORO: https://nac-arduino.herokuapp.com/
REPOSITORIO: https://github.com/rody7val/nac-arduino/

Rodolfo Valguarnera, NAC - Pigüé

Contenidos:

1. Variables.
2. Desafío variable.
3. Arrays.
4. Desafío array.
5. Operadores lógicos.
6. Ciclo For.
7. Desafío for.

Variables

Programar consiste básicamente en decirle a tu Arduino lo que tiene que hacer.

Un programa (o 'sketch' en la jerga Arduino) consigue este objetivo mediante el procesamiento de datos y la transmisión de estos datos procesados a los actuadores.

Lo que llamamos variables es simplemente una manera de representar estos datos dentro del sketch para facilitar su manipulación.

Desde un punto de vista práctico, podemos considerar las variables como los cajones de un escritorio, cada uno tiene una etiqueta describiendo el contenido y dentro de él se encuentra el valor de la variable (el contenido del cajón).

Hay tantos tipos de variables como de datos: números de todo tipo representados de diferentes maneras (enteros, reales, binarios, decimales, hexadecimales, etc.), textos (de un solo o varios caracteres o líneas), matrices (arrays), constantes, etc.

Variables

Los tipos de variables mas conocidos:

TIPO	DESCRIPCIÓN	EJEMPLO
`int`	Almacena números enteros.	`int test = 28927;`
`float`	Almacena números reales (con decimales).	`float test = 3.56;`
`String`	Almacena cadenas de texto.	`String test = "Hola Mundo";`
`boolean`	Almacena sólo dos valores: true (verdadero) o false (falso).	`boolean test = true;`

Variables

Como utilizar una variable:

Cuando queremos utilizar una variable primero hay que declarar el tipo de variable de la que se trata (por ejemplo "int") y luego el nombre que le queremos dar a esa variable ("test" en los ejemplos de la tabla anterior).

Podemos dejar la variable sin inicializar (es decir, sin asignarle un valor de partida):

```
int comienzo;
```

o, asignarle un valor inicial:

```
int comienzo = 0;
```

Desafío Variable

Crear un nuevo prototipo Arduino en circuits.io para comprender el uso de las variables. El prototipo debe hacer parpadear un LED con intervalos de un segundo. Respetar los siguientes requisitos mínimos:

- Nombre: Parpadear-LED.
- Componentes: Arduino Uno R3, Resistor, LED.
- Sketch:

```
int pin = 6;  //la variable pin se inicializa a 6, es decir vamos a activar el pin 6

void setup(){
  pinMode(pin, OUTPUT);
}

void loop(){
  digitalWrite(pin, HIGH);
  delay(1000);
  digitalWrite(pin, LOW);
  delay(1000);
}
```

Arrays

Un "array" o arreglo, es una colección indexada de variables del mismo tipo (como un diccionario). En el caso de un array el índice no es una palabra como en el diccionario, sino simplemente un número (un número de orden de la variable concreta dentro del array).

Ejemplo para declarar un array con un tamaño de 6 índices:

```
int arr[6];
```

Si deseas inicializarlo al mismo tiempo:

```
int arr[6] = {1,2,3,4,5,6};
```

Para inicializar un array no es necesario especificar el tamaño del índice:

```
int arr[] = {1,2,3,4,5,6};
```

Pero no puedes declarar un array sin índices, sin inicializarlo. El siguiente ejemplo daría error:

```
int arr[];
```

Arrays

Acceso a los datos:

Para acceder a los datos almacenados dentro de un array es necesario invocar el nombre del arreglo junto al índice.

Importante!: El primer índice dentro del array es siempre 0 (no 1).

Ejemplo de selección de pines mediante un arreglo:

```
int pines[] = {6,13};  //en el array pines guardamos como referencia los pines 6 y 13

pinMode(pines[0], OUTPUT);  //configurar pin 6 como salida
pinMode(pines[1], OUTPUT);  //configurar pin 13 como salida
```

Desafío Array

Crear un nuevo prototipo Arduino en circuits.io para comprender el uso de los arrays. El prototipo debe hacer parpadear dos LED o mas con intervalos de un segundo. Respetar los siguientes requisitos mínimos:

- Nombre: Parpadear-LED-con-Arrays.
- Componentes: Arduino Uno R3, Resistor, LED.
- Sketch:

```
int pines[] = {9,13};       //array con dos índices
int ledRojo = pines[0];     //variable ledRojo equivale al pin 9
int ledVerde = pines[1];    //variable ledVerde equivale al pin 13

void setup(){
  pinMode(ledRojo, OUTPUT);
  pinMode(ledVerde, OUTPUT);
}

void loop(){
  digitalWrite(ledRojo, HIGH);   //encender pin 9
  digitalWrite(ledVerde, HIGH);  //encender pin 13
  delay(1000);
  digitalWrite(ledRojo, LOW);    //apagar pin 9
  digitalWrite(ledVerde, LOW);   //apagar pin 13
  delay(1000);
}
```

Operadores Lógicos

Programar también consiste en decirle a nuestro Arduino que es lo que queremos que haga cuando se confronte con una o varias opciones. En otras palabras, queremos que el Arduino pueda tomar decisiones en base a los datos disponibles en ese momento (el valor de las variables).

Estas decisiones se toman en base a el resultado de uno o varios tests lógicos ("boolean tests" en inglés). El resultado de estos test lógicos será true o false (verdadero/falso) y determinará la decisión elegida.

Ejemplos de test lógicos:

```
60 < 120;   //true
30 > 28;    //true
45 <= 32;   //false
20 == 21;   //false
```

Operadores lógicos y su descripción:

OPERADOR	DESCRIPCIÓN	OPERADOR	DESCRIPCIÓN
<	Menor que	>	Mayor que
<=	Menor o igual que	>=	Mayor o igual que
!=	Diferente	==	Igual que

Bucle For

Las estructuras en ciclos, en bucle o "loops" en inglés, realizan una tarea de manera repetitiva hasta que ésta se ha completado.

El tipo de bucle FOR se utiliza para ejecutar un bloque de código un cierto número de veces. En general se usan con un contador incremental que va aumentando hasta alcanzar un valor prefijado, momento en el que el bucle se da por terminado.

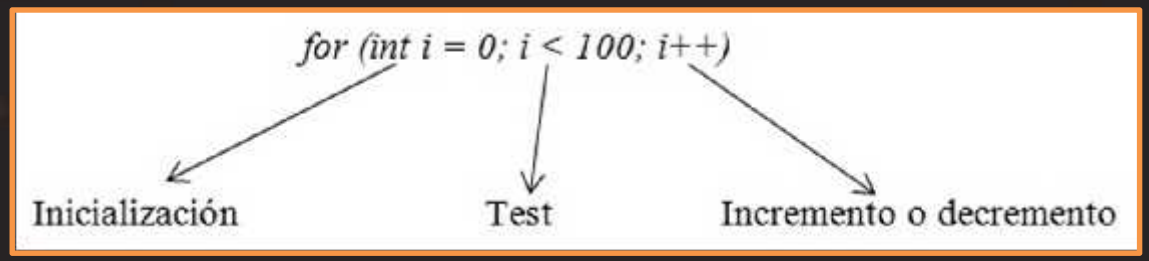
El ejemplo siguiente muestra el típico bucle FOR que imprime el valor del contador `i` de 0 hasta 99 hasta que se apaga el Arduino:

```
void setup(){
  Serial.begin(9600);
}

void loop(){
  for (int i = 0; i < 100; i++){    //ciclo for que se repite 100 veces
    Serial.print(i);
  }
}
```

Bucle For

Como funciona el bucle FOR ?



La variable i es inicializada con el valor 0. Al final de cada bucle la variable se incrementa en 1 (i++ es una manera abreviada de codificar i = i + 1).

El código en el interior del bucle se ejecuta una vez tras otra hasta que alcanza el valor 100. En ese punto el bucle finaliza y recomienza volviendo a poner la variable i a 0.

La primera línea del bucle es la instrucción “for”. Esta instrucción tiene siempre tres partes: inicialización, test y el incremento o decremento de la variable de control o contador.

La inicialización sólo sucede una vez al comienzo del bucle. El test se realiza cada vez que el bucle se ejecuta. Si el resultado del test es “verdadero” (true), el bloque de código se ejecuta y el valor del contador se incrementa (++) o decrementa (–-) tal como está especificado en la tercera parte del “for”. El bloque de código (o rutina) continuará ejecutándose hasta que el resultado del test sea “false” (es decir, cuando el contador i haya alcanzado el valor 100).

Desafío For

Crear un nuevo prototipo Arduino en circuits.io para comprender el uso del bucle For. El prototipo debe hacer parpadear dos LED o mas con intervalos de un segundo. Respetar los siguientes requisitos mínimos:

- Nombre: Parpadear-LED-con-For.
- Componentes: Arduino Uno R3, Resistor, LED.
- Sketch:

```
int pines[] = {7,9,13};            //array con 3 índices

void setup(){
  for(int i = 0; i < 3; i++){      //el bloque for se ejecutará 3 veces
      pinMode(pines[i], OUTPUT);   //suficiente para recorrer todo el array
  }
}

void loop(){
  for(int i = 0; i < 3; i++){
    digitalWrite(pines[i], HIGH);   //encender los 3 leds
    delay(1000);
  }
  for(int i = 0; i < 3; i++){
    digitalWrite(pines[i], LOW);    //apagar los 3 leds
    delay(1000);
  }
}
```

Fin del tema.

Muchas gracias!

FORO: https://nac-arduino.herokuapp.com/
REPOSITORIO: https://github.com/rody7val/nac-arduino/

Rodolfo Valguarnera, NAC - Pigüé