



# Escuela Industrial San Antonio

Establecimiento Particular Subvencionado de Financiamiento Compartido

Sostenedor: Fundación Educacional San Antonio – RUT.: 65.624.710-K

RBD: 2044-3

## Guía N° 4 SCD


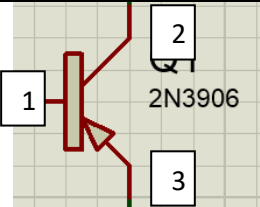
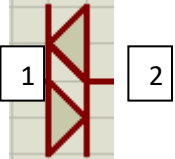
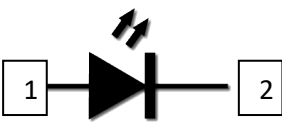
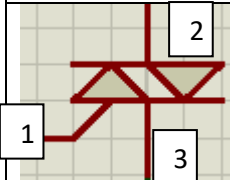
Nombre : .....  
 Profesor Sr. : José Ugalde Zamora.  
 Curso : ...3°D .....

22 junio al 26 de junio 2020, vía correo electrónico a [profesor.ugalde@gmail.com](mailto:profesor.ugalde@gmail.com) Indicando en el asunto: **apellido, nombre, curso, asignatura.**

-Si la entrega es en formato papel, debe ser entregado en la escuela industrial de San Antonio Av. Centenario 250.

Aprendizaje(s) Esperado(s):.....Conocer componentes electrónicos y programación básica arduino

### 1) Escriba el nombre del componente y el nombre de sus pines.

Símbolo	Nombre Componente Y nombre Pines	Símbolo	Nombre Componente Y nombre Pines
	1 2  Nombre		1 2 3  Nombre
	1 2  Nombre		1 2  Nombre
	1 2 3  Nombre		



## Escuela Industrial San Antonio

Establecimiento Particular Subvencionado de Financiamiento Compartido

Sostenedor: Fundación Educacional San Antonio – RUT.: 65.624.710-K

RBD: 2044-3

### Estructura

La estructura básica del lenguaje de programación de Arduino es bastante simple y se compone de al menos dos partes. Estas dos partes necesarias, o funciones, encierran bloques que contienen declaraciones, estamentos o instrucciones.

#### **void setup()**

```
{  
estamentos; “se declaran variables entradas y salidas a utilizar en el programa”  
}
```

#### **void loop()**

```
{  
estamentos; “se realizan funciones, se utilizan lo declarado en el Void Setup”  
}
```

### void setup()

En donde setup() es la parte encargada de recoger la configuración y loop() es la que contienen el programa que se ejecutará cíclicamente (de ahí el término loop –bucle-). Ambas funciones son necesarias para que el programa trabaje.

La función de configuración debe contener la declaración de las variables. Es la primera función a ejecutar en el programa, se ejecuta sólo una vez, y se utiliza para configurar o inicializar pinMode (modo de trabajo de las E/S), configuración de la comunicación en serie y otras.

La función bucle (loop) siguiente contiene el código que se ejecutara continuamente (lectura de entradas, activación de salidas, etc) Esta función es el núcleo de todos los programas de Arduino y la que realiza la mayor parte del trabajo.

La función setup() se invoca una sola vez cuando el programa empieza. Se utiliza para inicializar los modos de trabajo de los pins, o el puerto serie. Debe ser incluido en un programa aunque no haya declaración que ejecutar. **ejemplo**

```
void setup()  
{  
pinMode(pin, OUTPUT); // configura el 'pin' como salida  
}
```

### void loop()

Después de llamar a setup(), la función loop() hace precisamente lo que sugiere su nombre, se ejecuta de forma cíclica, lo que posibilita que el programa este respondiendo continuamente ante los eventos que se produzcan en la tarjeta. **ejemplo**

#### **void loop()**

```
{  
digitalWrite(pin, HIGH); // pone en uno (on, 5v) el 'pin'  
delay(1000); // espera un segundo (1000 ms)  
digitalWrite(pin, LOW); // pone en cero (off, 0v.) el 'pin'  
delay(1000);  
}
```



## Escuela Industrial San Antonio

Establecimiento Particular Subvencionado de Financiamiento Compartido

Sostenedor: Fundación Educacional San Antonio – RUT.: 65.624.710-K

RBD: 2044-3

### Punto y Coma ;

El punto y coma “;” se utiliza para separar instrucciones en el lenguaje de programación de Arduino. También se utiliza para separar elementos en una instrucción de tipo “bucle ejemplo

**for”.**

```
int x = 13; // declara la variable 'x' como tipo entero de valor 13
```

### Comentarios //

Una línea de comentario empieza con // y terminan con la siguiente línea de código. Al igual que los comentarios de bloque, los de línea son ignoradas por el programa y no ocupan espacio en la memoria.

```
// esto es un comentario
```

### HIGH , LOW

Estas constantes definen los niveles de salida altos o bajos y se utilizan para la lectura o la escritura digital para las patillas. ALTO se define como en la lógica de nivel 1, ON, ó 5 voltios, mientras que BAJO es lógica nivel 0, OFF, o 0 voltios.

Ejemplo:

```
digitalWrite(13, HIGH); // activa la salida 13 con un nivel alto (5v.)
```

### Input, Output

Estas constantes son utilizadas para definir, al comienzo del programa, el modo de funcionamiento de los pines mediante la instrucción pinMode de tal manera que el pin puede ser una entrada INPUT o una salida OUTPUT.

```
pinMode(13, OUTPUT); // designamos que el PIN 13 es una salida
```

### EJERCICIO

2) continuación realizaremos un programa en el cual se enciende un pin digital (6) de arduino, que como vimos en la guía anterior eran del 0 al 13 digital pin.

```
void setup()
```

```
{
```

```
// “se declara pin digital 6 como salida, es decir, tendremos un valor alto o bajo en el pin 6, según funcionamiento del programa.
```

```
}
```

```
void loop()
```

```
{
```

```
// se pone el pin digital 6 en alto, es decir, que por el pin 6 tendremos un valor alto o de 5Volts
```

```
}
```