Modelamiento de Sistemas II

“Uso de Arduino”

20-07-2023

OBJETIVO

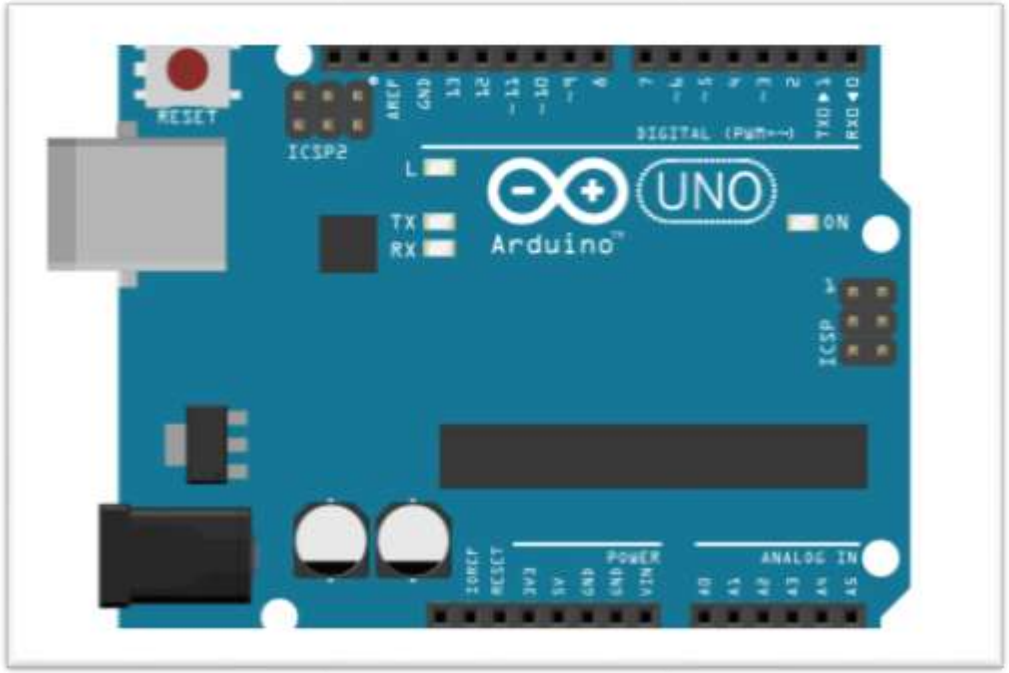
- Conocer la codificación de ARDUINO.

- Aplicar los códigos de programación, para elaboración de proyectos.

ARDUINO

•Es una **plataforma de desarrollo** basada en una **placa electrónica de hardware libre** que incorpora un microcontrolador re-programable y con una serie de puertos disponibles.

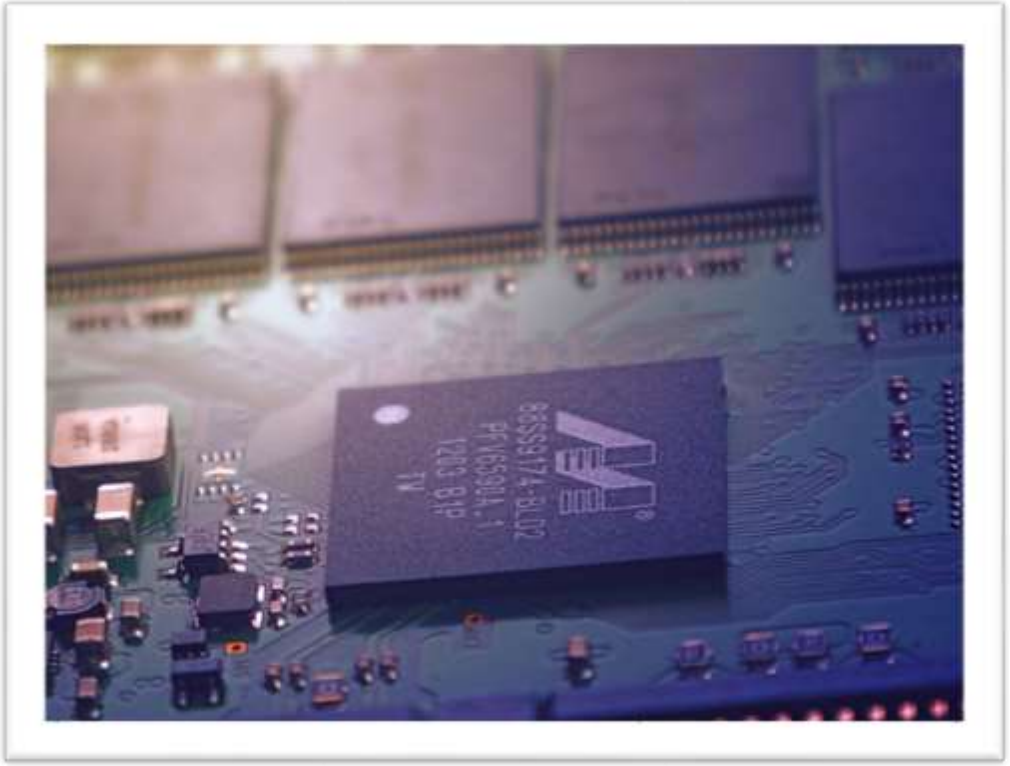
•Estos permiten **establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores** de una manera muy sencilla para elaborar proyectos y también MVP (producto mínimo viables) de manera rápida y efectiva.



MICROCONTROLADORES

•Es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria.

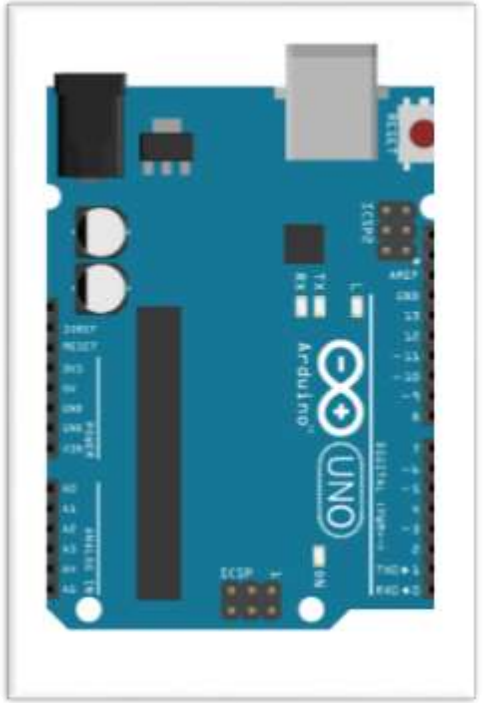
•En su interior contiene tres principales unidades funcionales de una computadora: unidad central de procesos, memoria y periféricos de entrada/salida.



ARDUINO Y APLICACIONES INDUSTRIALES

Existen multitud de entornos de aplicación de Arduino:

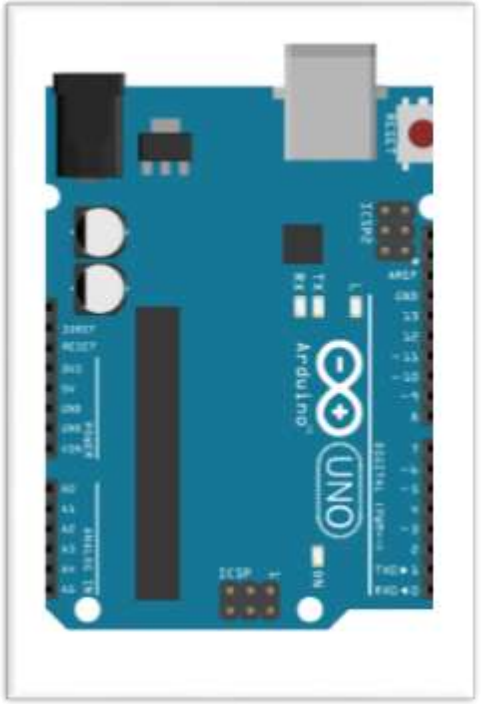
- Automatización industrial.
- Domótica (automatización de vivienda).
- Herramienta de prototipado.
- Plataforma de entrenamiento para aprendizaje de electrónica.
- Tecnología para artistas.
- Eficiencia energética.
- Monitorización.
- Adquisición de datos.
- DIY (Do it yourself).
- Aprendizaje de habilidades tecnológicas y programación, etc.



ARDUINO Y APLICACIONES INDUSTRIALES

También podemos ver a Arduino en lo siguientes tipos de usos:

1. Monitorización en Tiempo real.
2. Control remoto de instalaciones.
3. Eficiencia energética.
4. Automatización de procesos.
5. Automatización de informes/Cuadros de mando.
6. Mantenimientos Predictivos.

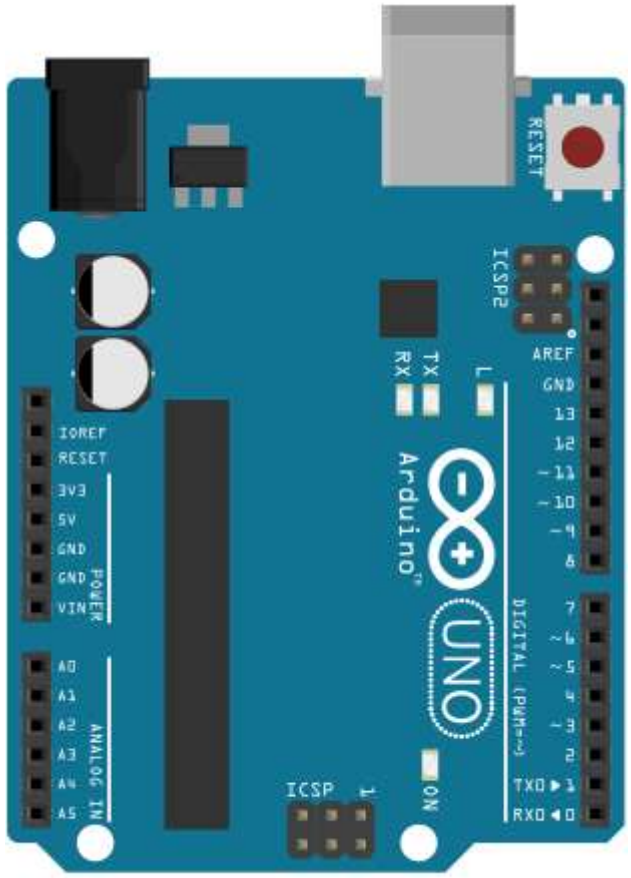


ARDUINO IDE

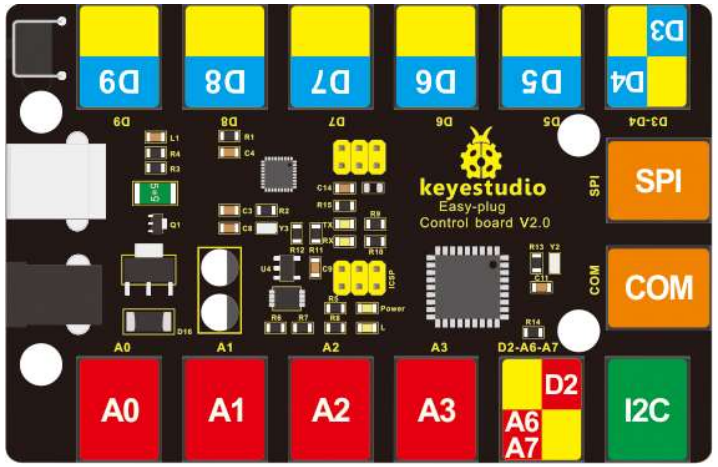
LINK de descarga:

https://www.arduino.cc/en/Main/Donate

1.Hacer click en “Just download”
2. Instalar Arduino IDE
3. Instalar DRIVERS



EASYPLUG



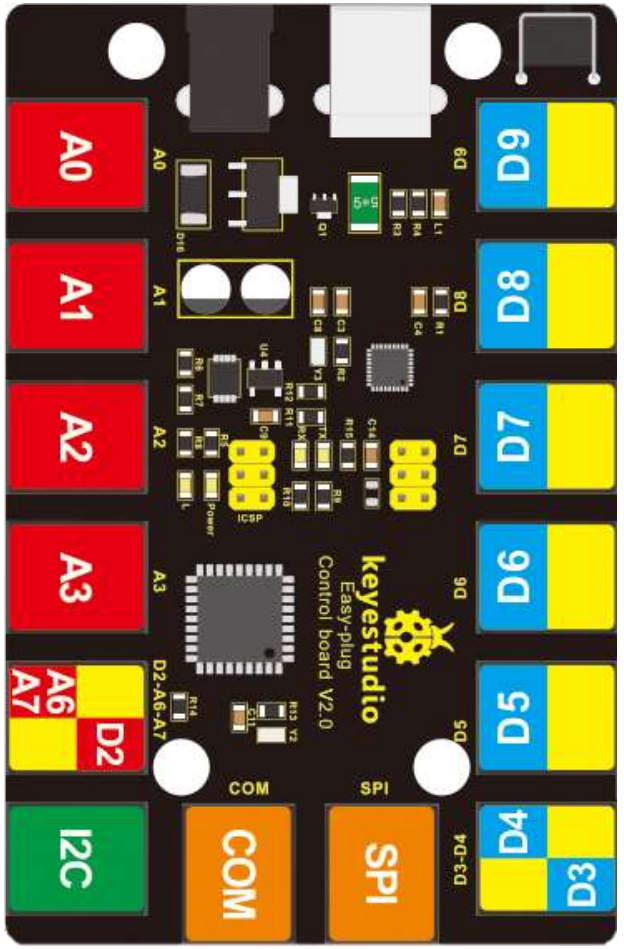
El Arduino EASYPLUG es una placa electrónica que a diferencia de Arduino, viene con **conectores tipo RJ11** lo que permite realizar aún mas rápido los circuitos por que sólo utilizan un cable de conexión, por lo que es muy útil a la hora de elaborar circuitos rápidos y sencillos, gracias a este tipo de conectores se eliminan los errores en circuitos mal conectados.

Por otro lado también tiene **colores** que diferencian los tipos de puertos para hacerlo mas intuitivo a la hora de trabajar.

Por último, se puede programar a través de **lenguaje de código** y también a través de **bloques.**

TIPOS DE PUERTOS.

- *Digital.*
- *Análogo.*
- *I2C: permite conectar dispositivos de baja velocidad como microcontroladores.*
- *SPI: Interfaz periférica Serial, diseñado, para que los microcontroladores se comuniquen entre sí.*
- *COM*



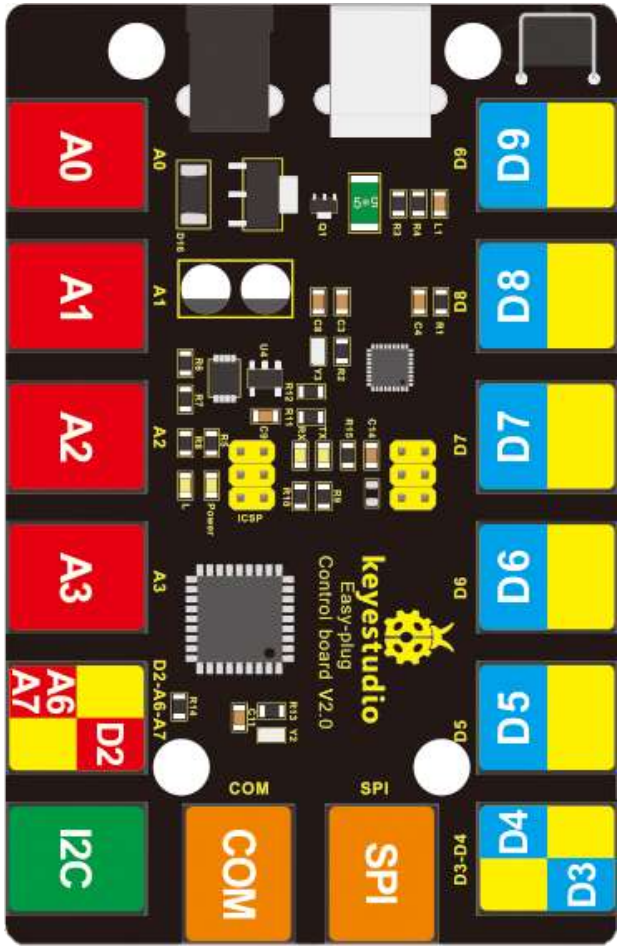
TIPOS DE PUERTOS.

Actuador Digital

Sensor Digital

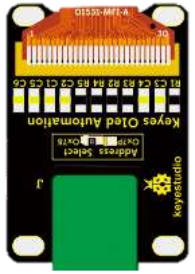
Sensor Análogo

Comunicación



Doble Canal Digital

Doble Canal Análogo

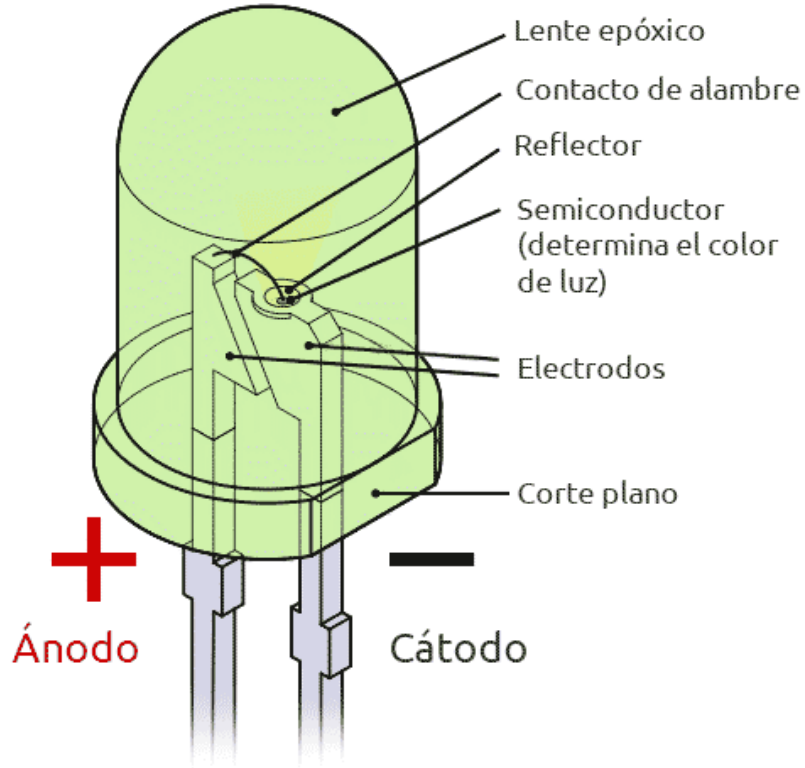


I2C

COMPONENTES: DIODO LED

•El diodo LED, esta compuesto por un ánodo (pata positiva) y un cátodo (pata negativa).

•Cuando conectamos con polarización directa el diodo led el semiconductor de la parte de arriba permite el paso de la corriente que circulará por las los terminales (cátodo y ánodo) y al pasar por el semiconductor, este semiconductor emite luz.



LEY DE OHM

Es la ley mas básica de la electrónica en cuanto a los conceptos de voltaje, corriente, y resistencia, la siguiente imagen grafica el comportamiento de esta ley.

FUNCIONES PRINCIPALES ARDUINO

```
void setup() {

}

void loop() {


}
```

Void Setup es una función en donde se ejecutan las configuraciones básicas solo una vez

Void Loop es nuestra función principal donde se realizarán las programaciones principales, es decir, donde se desarrolla el código y esta función no tiene término.

FUNCIÓN: PINMODE

Esta función nos permite definir como trabaja nuestro puerto digital, es decir, como entrada o como salida, las configuraciones son las siguientes:

```
pinMode(13, INPUT);
```

Se definió el puerto 13 como entrada

```
pinMode(8, OUTPUT);
```

Se definió el puerto 8 como salida

Es muy importante también saber que para trabajar con sensores, el puerto al cual se conectara el sensor debe definirse como **entrada**.

Para trabajar con dispositivos de control , es decir actuadores, el puerto se debe definir como **salida**.

FUNCIÓN: DIGITALWRITE

Esta función lo que nos permite hacer es escribir un estado alto “HIGH” o un estado Bajo “LOW” en un puerto seleccionado, con este tipo de funciones podemos controlar motores, válvulas, relés, etc.

La función esta compuesta de la siguiente manera:

```
digitalWrite(13, HIGH);
digitalWrite( 8, LOW);
```

Estamos escribiendo en el puerto 13 un estado alto (1)

Estamos escribiendo en el puerto 8 un estado bajo (0)

FUNCIÓN: DELAY

El **Delay** o también conocida como “retardo” es una función que hace que el procesador espere.

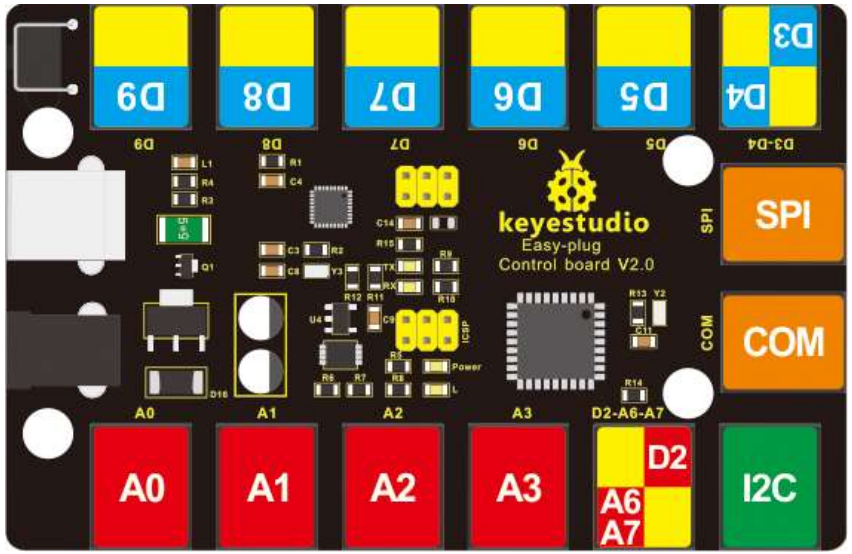
Podemos realizar pausas entre las ejecuciones del programa, como por ejemplo en un veamos el ejemplo en un sistema de aire acondicionado**:**

-Acción 1 (medir temperatura)
-Espera (1 segundo)
-Acción 2 (Subir o bajar la temperatura, dependiendo de como se haya configurado)
-Espera (2 segundos)

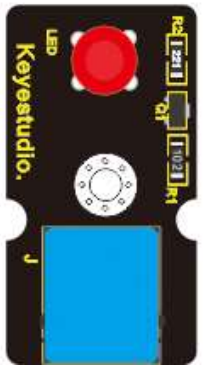
```
delay(1000);
```

Trabaja con valores enteros, en milisegundos y la relación es la siguiente:
1000 milisegundos = 1 segundo

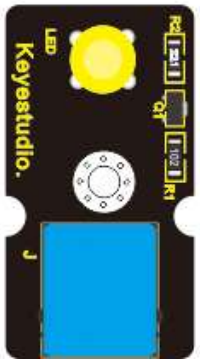
MATERIALES PARA LA SESIÓN



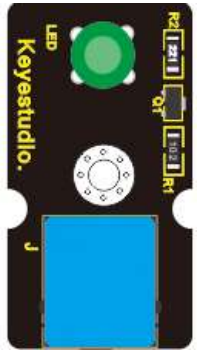
Easy-Plug



LED
rojo

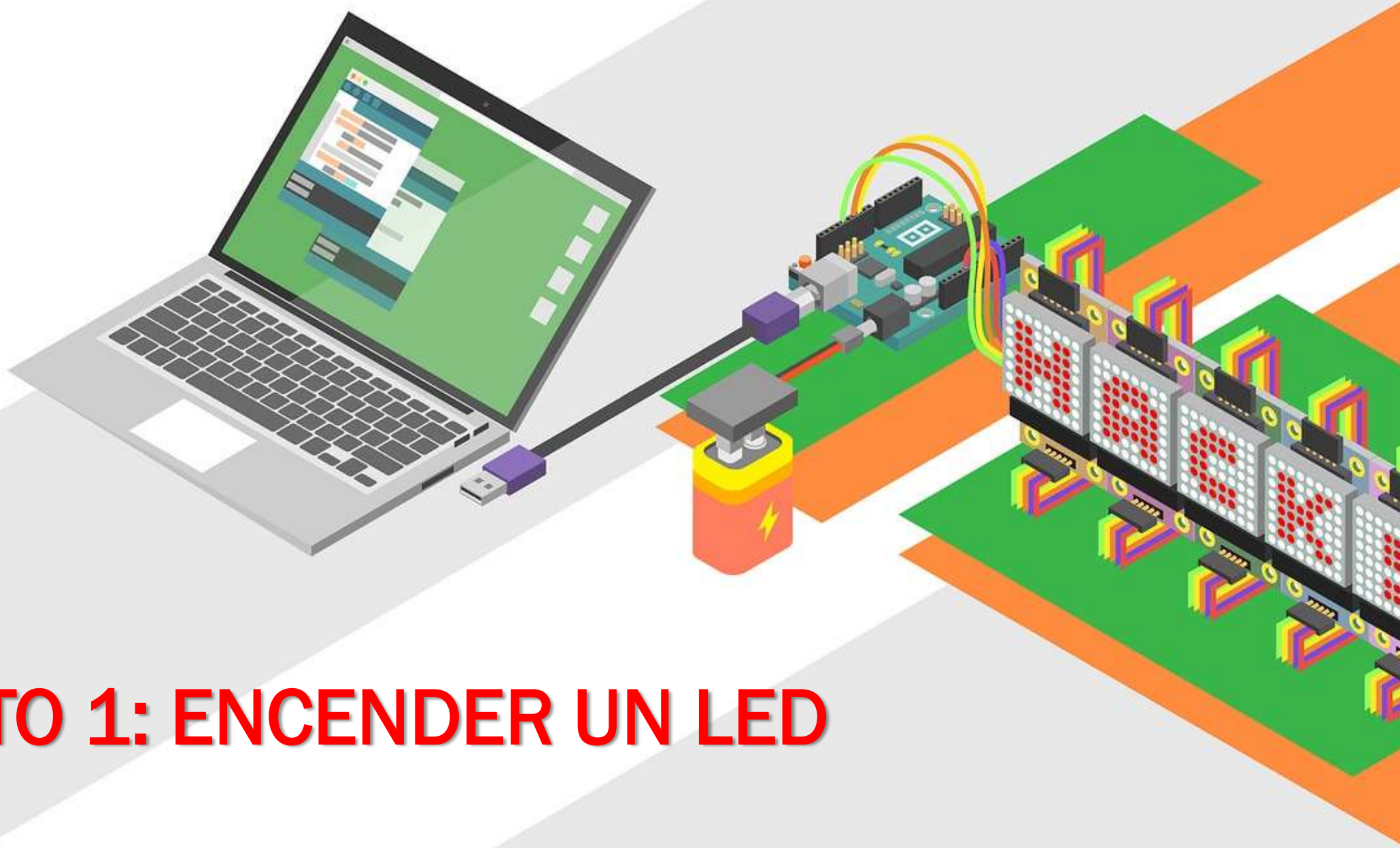


LED
Amarillo



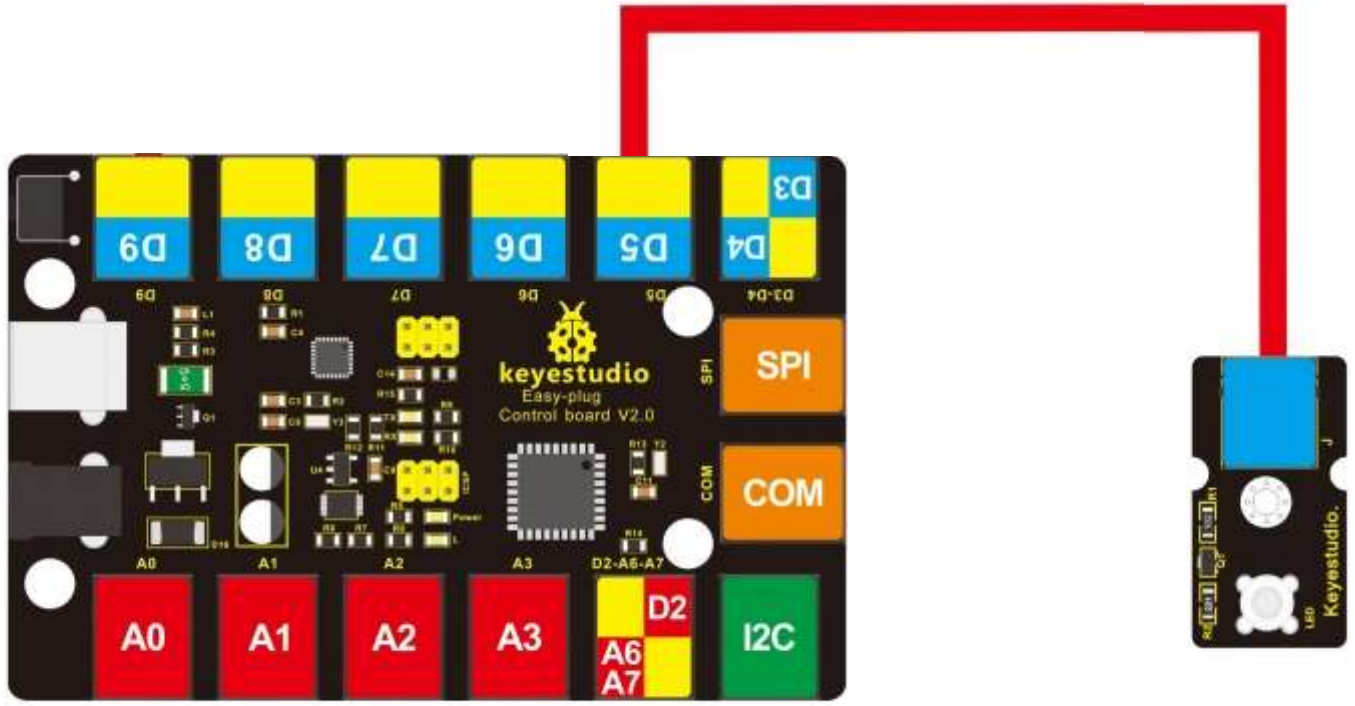
LED
Verde

RJ11



PROYECTO 1: ENCENDER UN LED

DIAGRAMA DE CONEXIÓN



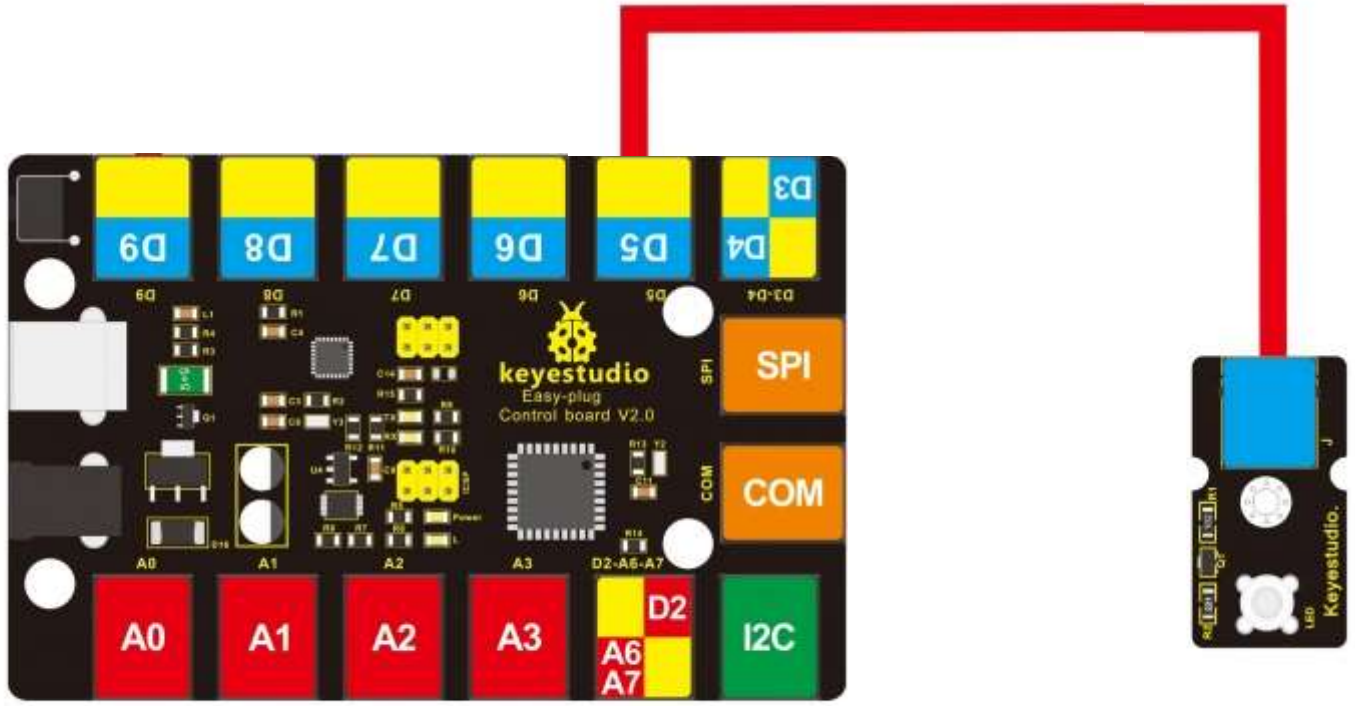
PROGRAMACIÓN

```
void setup() {
pinMode(5, OUTPUT);
}

void loop() {
  digitalWrite(5, HIGH);
}
```

PROYECTO 2: PARPADEO DE UN LED

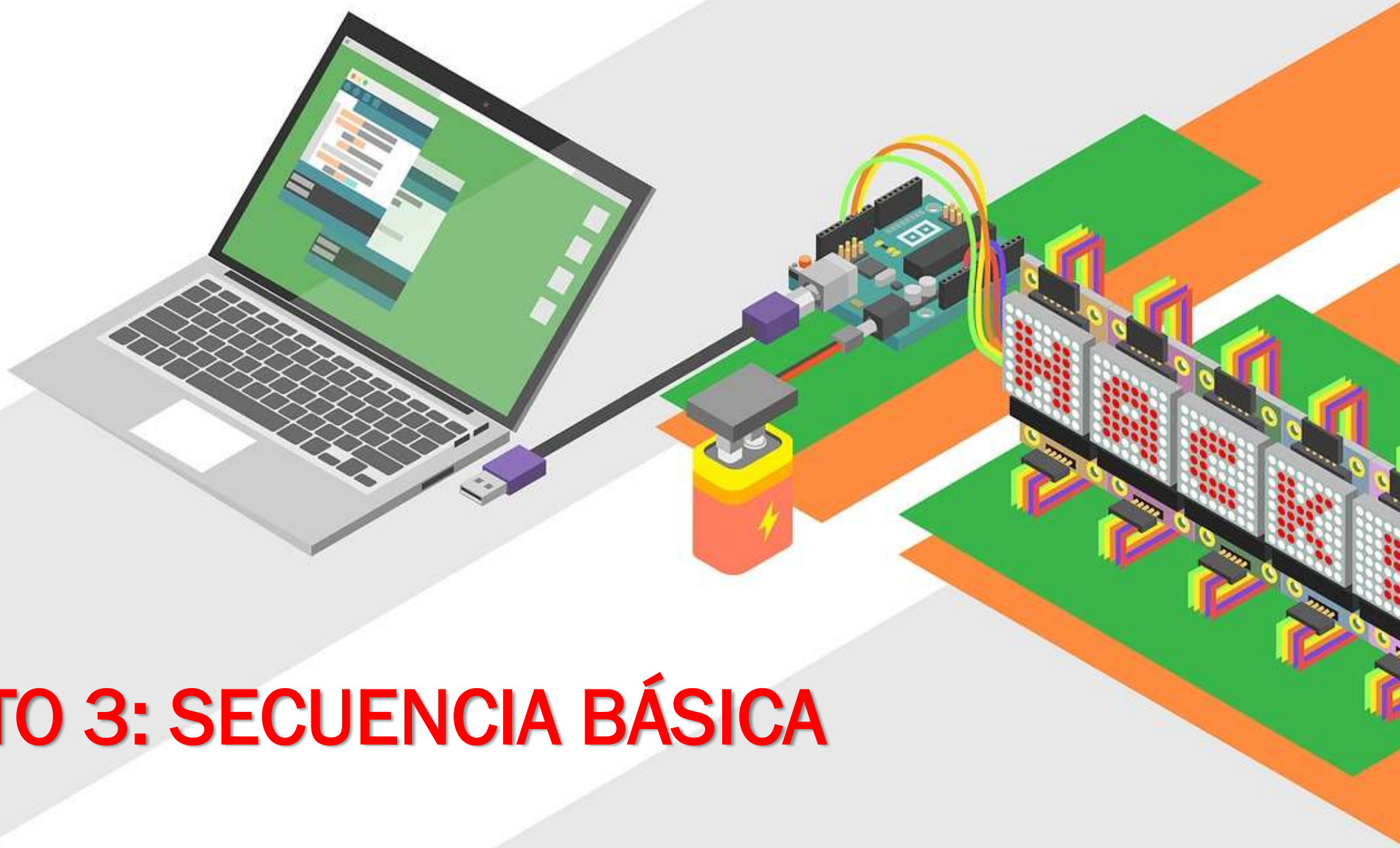
DIAGRAMA DE CONEXIÓN



PROGRAMACIÓN

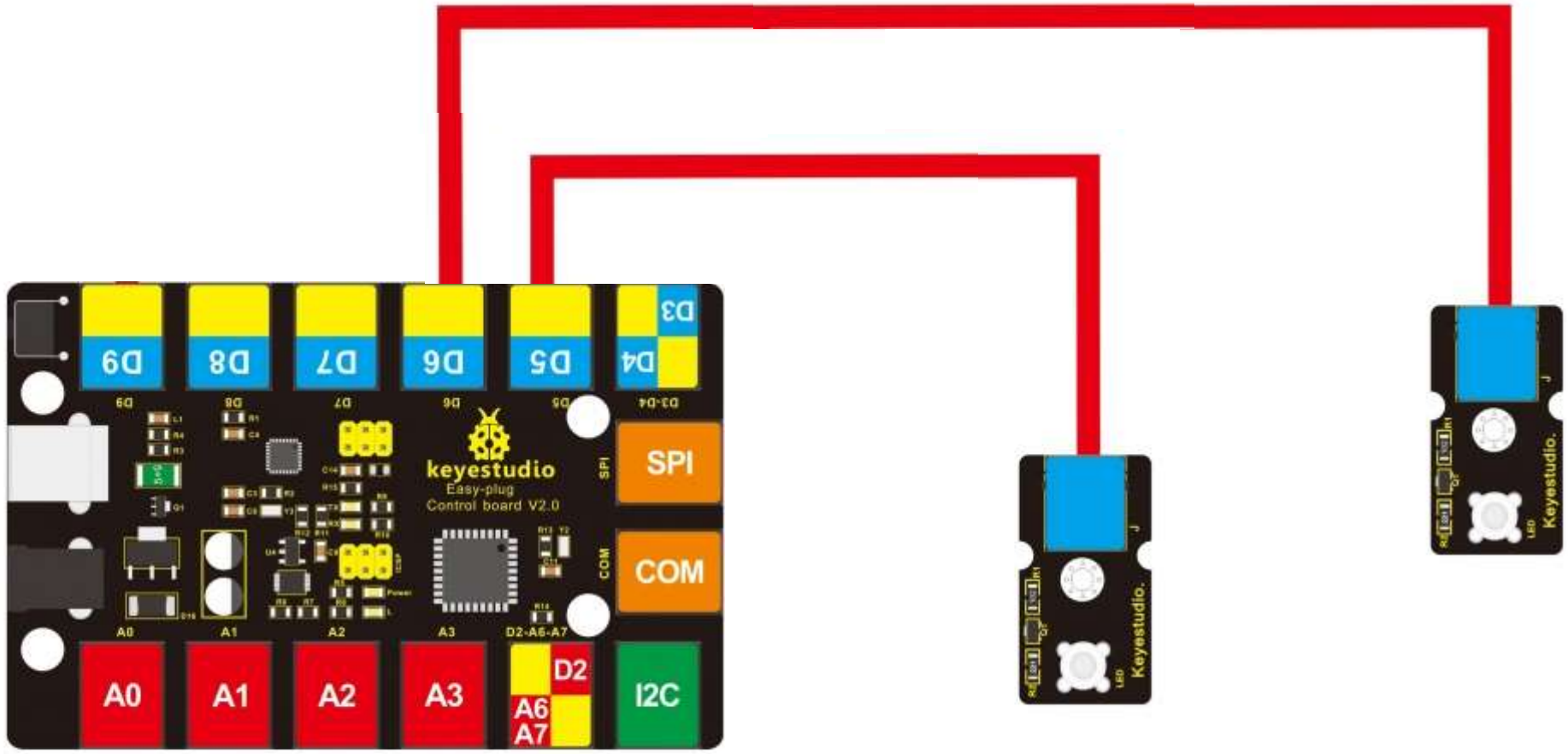
```
void setup() {
pinMode(5, OUTPUT);
}

void loop() {
  digitalWrite(5, HIGH);
  delay(1000);
  digitalWrite(5, LOW);
  delay(1000);
}
```



PROYECTO 3: SECUENCIA BÁSICA

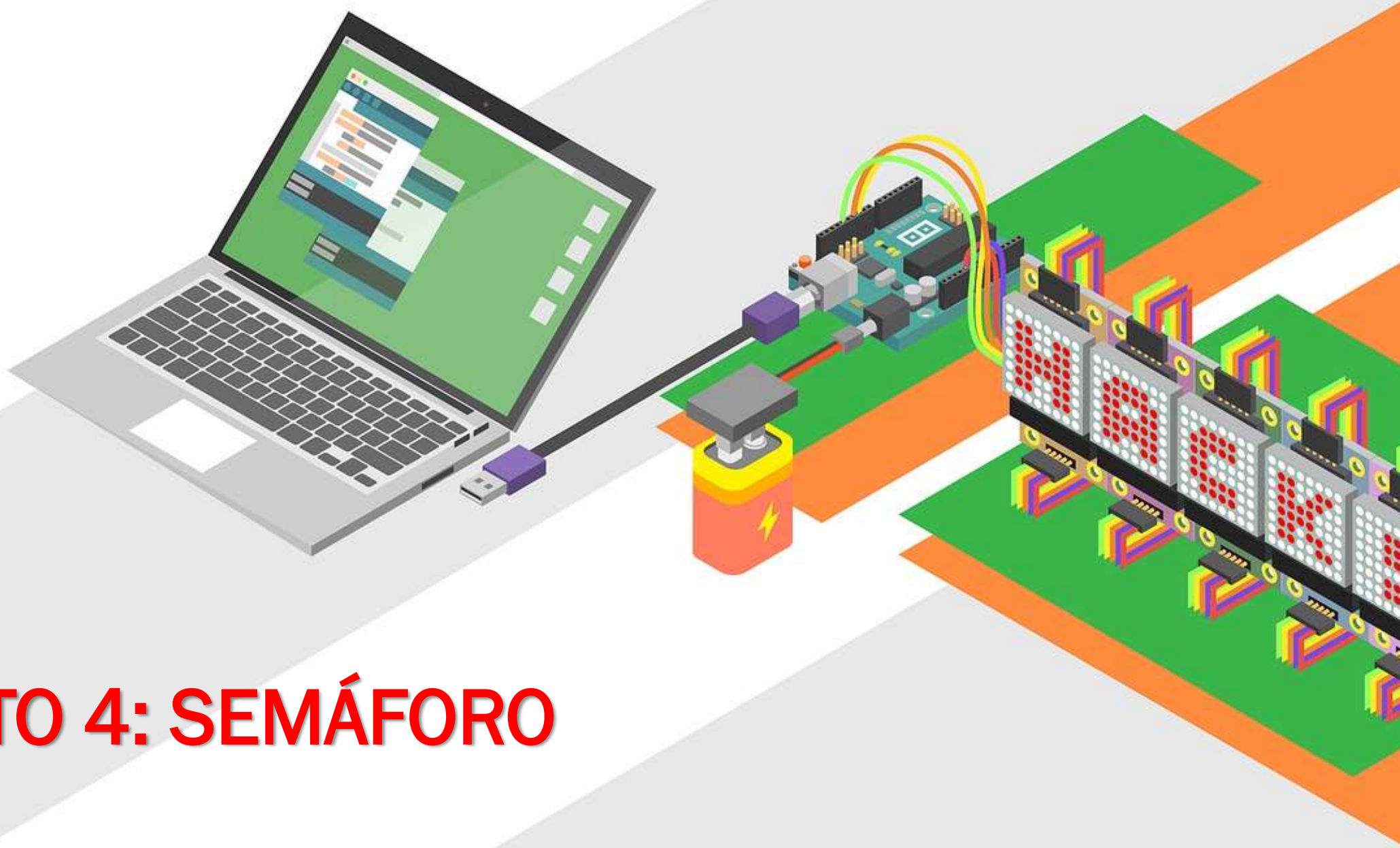
DIAGRAMA DE CONEXIÓN



PROGRAMACIÓN

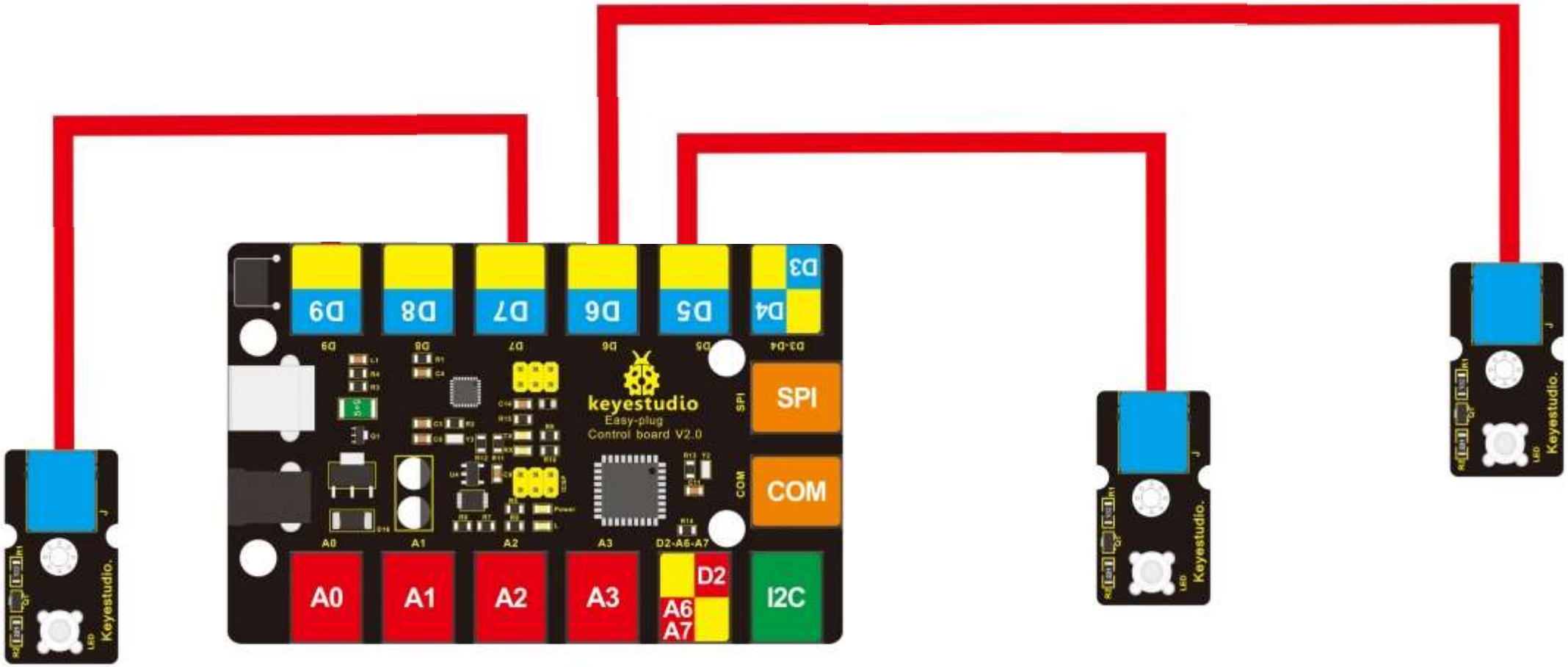
```
void setup() {
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
}

void loop() {
  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
  delay(1000);
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
  delay(1000);
}
```



PROYECTO 4: SEMÁFORO

DIAGRAMA DE CONEXIÓN



PROGRAMACIÓN

```
void setup() {
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
}

void loop() {
  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  delay(3000);
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
  digitalWrite(7, LOW);
  delay(1000);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, HIGH);
  delay(3000);

}
```