

**UNIVERSIDAD NACIONAL DE JAÉN  
COMISIÓN ORGANIZADORA**



**UNIVERSIDAD NACIONAL  
DE JAÉN**

**FACULTAD DE INGENIERÍA**

**MONOGRAFÍA**

**Programación de Ingeniería**

**Lenguaje de Programación Embebido:  
Arduino**

**AUTORES:**

**Mg. Frans Fuentes Maza**

**Jaén – Perú, octubre, 2024**



## RESUMEN

Arduino es una plataforma de prototipado electrónico de código abierto que combina hardware y software, permitiendo a principiantes y expertos desarrollar sistemas embebidos de manera sencilla. Utiliza un entorno de desarrollo integrado (IDE) donde se programan placas, como Arduino Uno y Mega, con un lenguaje basado en C/C++. La estructura del código incluye funciones básicas como `setup()` y `loop()`, y cuenta con numerosas bibliotecas que facilitan la interacción con diversos dispositivos. Sus aplicaciones abarcan robótica, domótica, Internet de las Cosas (IoT) y educación, transformando la enseñanza de la electrónica y acelerando la innovación en la industria. Aunque presenta algunas limitaciones, su comunidad activa y continua evolución aseguran su relevancia en el futuro del desarrollo embebido.



## Contenido

<b>RESUMEN .....</b>	<b>2</b>
<b>Contenido.....</b>	<b>3</b>
<b>INTRODUCCIÓN .....</b>	<b>4</b>
<b>CAPÍTULO 1 GENERALIDADES .....</b>	<b>5</b>
<b>1.1. Historia y Evolución de Arduino.....</b>	<b>5</b>
<b>1.2. Que es el Arduino .....</b>	<b>5</b>
<b>1.3. Características generales .....</b>	<b>5</b>
<b>1.4. Entorno de Desarrollo Integrado (IDE) .....</b>	<b>5</b>
<b>CAPÍTULO 2 .....</b>	<b>7</b>
<b>Estructura y Componentes del Arduino.....</b>	<b>7</b>
<b>2.1. Estructura de un Programa Arduino .....</b>	<b>7</b>
<b>2.2. Componentes de Arduino .....</b>	<b>7</b>
<b>2.3. Arquitectura de Hardware .....</b>	<b>8</b>
<b>2.4. Aplicaciones de Arduino en Sistemas Embebidos.....</b>	<b>9</b>
<b>CAPÍTULO 3 .....</b>	<b>10</b>
<b>Impacto en la Educación y la Industria.....</b>	<b>10</b>
<b>3.1. Educación .....</b>	<b>10</b>
<b>3.2. Industria .....</b>	<b>12</b>
<b>CAPÍTULO 4 .....</b>	<b>13</b>
<b>Desafíos y Limitaciones de Arduino.....</b>	<b>13</b>
<b>4.1. Limitaciones de Hardware .....</b>	<b>13</b>
<b>4.2. Progreso del Aprendizaje con Arduino .....</b>	<b>13</b>
<b>4.3. Futuro de Arduino y el Desarrollo de Sistemas Embebidos.....</b>	<b>13</b>
<b>CONCLUSIÓN .....</b>	<b>15</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>16</b>



## INTRODUCCIÓN

El desarrollo de sistemas embebidos ha evolucionado muy rápidamente en la forma en que interactuamos con la tecnología en el día a día. Un sistema embebido es una aplicación de diseño para realizar tareas específicas, y generalmente está integrado en dispositivos más grandes. Estos sistemas están presentes en máquinas de procesos agroindustriales, electrodomésticos, vehículos, dispositivos médicos, entre otros.

Uno de los entornos más accesibles y populares para trabajar con sistemas embebidos es **Arduino**. Arduino no solo es una plataforma de hardware de código abierto, sino también un lenguaje de programación y un entorno de desarrollo integrado (IDE) que facilita el desarrollo de proyectos interactivos.

Por lo expuesto anteriormente, se planteó como objetivo de esta monografía es analizar en detalle el lenguaje de programación embebido Arduino, su estructura, características, y aplicaciones.

## CAPÍTULO 1 GENERALIDADES

### 1.1. Historia y Evolución de Arduino

Arduino fue desarrollado en el año 2005 por un grupo de investigadores liderados por Massimo Banzi en el Interaction Design Institute Ivrea, Italia. Su objetivo inicial era crear una herramienta de bajo costo y fácil de usar para artistas y diseñadores sin experiencia previa en programación o electrónica. La plataforma ha mejorado considerablemente, con una amplia gama de placas, componentes y librerías que permiten diseñar y desarrollar aplicaciones desde proyectos básicos hasta sistemas embebidos más complejos.

### 1.2. Que es el Arduino

Es una plataforma para el diseño de prototipado electrónico de código abierto que combina un hardware fácil de usar con un entorno de desarrollo integrado (IDE) que permite programar placas de microcontroladores.

### 1.3. Características generales

Arduino utiliza una versión simplificada del lenguaje de programación C++, que permite a los usuarios desarrollar programas (también conocidos como sketches) para administrar el hardware embebido. Algunas de las características más destacadas son:

- 1. Fácil de Usar:** El lenguaje de programación está diseñado para ser sencillo, con sintaxis básica que facilita un aprendizaje significativo.
- 2. Bibliotecas predefinidas:** Arduino incluye muchas bibliotecas que facilitan la comunicación con sensores, motores, pantallas y otros componentes electrónicos.
- 3. Extensibilidad:** Al basarse en C++, permite a los usuarios más avanzados desarrollar sus propias bibliotecas y código personalizado.
- 4. Código abierto:** Tanto el hardware como el software de Arduino son de código abierto.

### 1.4. Entorno de Desarrollo Integrado (IDE)

El Arduino IDE es el entorno oficial para desarrollar y cargar código en las placas Arduino. Esta plataforma permite a los usuarios escribir código en un editor, compilarlo, ejecutarlo y cargarlo directamente en la placa. Las principales



características del IDE son:

- a) **Simulación y carga directa:** Permite simular, compilar y cargar el código de manera fácil y rápida en la placa.
- b) **Multiplataforma:** Funciona en Windows, Mac OS y Linux.
- c) **Verificación de código:** Ofrece una funcionalidad para verificar la sintaxis del código antes de cargarlo.
- d) **Monitor Serial:** Herramienta integrada que permite visualizar datos en tiempo real que se comunican entre el Arduino y el computador.

## CAPÍTULO 2

### Estructura y Componentes del Arduino

#### 2.1. Estructura de un Programa Arduino

Un sketch de Arduino tiene dos secciones básicas:

- a) **setup():** Función que se ejecuta una sola vez, cuando la placa es encendida o reiniciada. En esta función se configuran los pines de entrada y salida, se inicializan variables y se establecen las restricciones iniciales del sistema.

Figura 1 Sección Setup

```

34
35 void setup()
36
37     pinMode(ps1, OUTPUT);
38     pinMode(ps2, OUTPUT);
39     Serial.begin(9600);
40     Serial.setTimeout(10);
41
42     P.begin();
43     P.setZoneEffect(1, true, PA_FLIP_LR);
44     P.displayText("BIO", PA_CENTER, SCROLL_SPEED, PAUSE_TIME, PA_PRINT, PA_NO_EFFECT);
45     P.displayAnimate();
46
47     lcd.init();
48     lcd.backlight();//enciende luz de pantalla
49     lcd.begin(16, 2);
50
51     //lcd.setCursor(0, 0);
52     lcd.print("Frans");
53     lcd.setCursor(0, 1);
54     lcd.print("P");
55
    
```

- b) **loop():** Función que se ejecuta repetitivamente mientras la placa está encendida. Aquí se coloca el código que controla el proceso del sistema en tiempo real.

Figura 2 Sección loop

```

1  int x=2;
2  void setup() {
3      pinMode(x,OUTPUT);
4
5  }
6
7  void loop() {
8      tone(x,250,2000);
9      delay(500);
10     noTone(x);
11     delay(300);
12 }
13
    
```

#### 2.2. Componentes de Arduino

##### a) Placas:

Las placas Arduino, como Arduino Uno, Arduino Mega y Arduino Nano, son microcontroladores que permiten realizar diversas tareas. Cada placa tiene diferentes características en términos de memoria, número de pines y capacidades de comunicación.

**b) IDE Arduino:**

El entorno de desarrollo integrado permite escribir, compilar y cargar programas (conocidos como "sketches") en la placa de Arduino. Es un software libre disponible para Windows, macOS y Linux.

**c) Bibliotecas:**

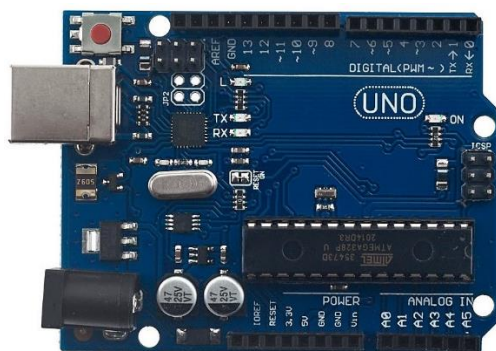
Arduino proporciona una serie de variedades de bibliotecas que facilitan la implementación de funciones específicas, como el control de motores, la comunicación con sensores, y la conectividad a Internet.

**2.3. Arquitectura de Hardware**

El hardware de Arduino está compuesto por diversas placas que contienen microcontroladores. Entre las más populares se encuentran:

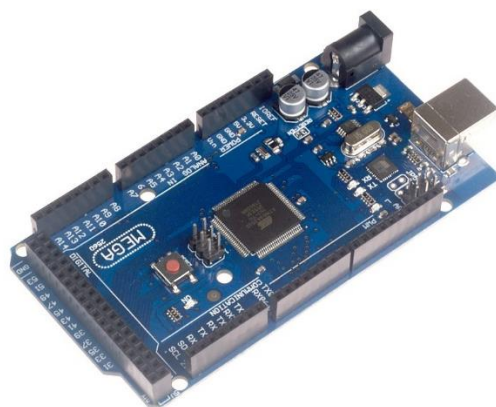
**Arduino UNO:** La placa más usada, con un microcontrolador ATmega328P, 14 pines digitales, 6 entradas analógicas y una memoria flash de 32 KB.

Figura 3 Arduino UNO



**Arduino MEGA:** Diseñada para proyectos más grandes, con más pines y un procesador más potente (ATmega2560), y existen varios modelos.

Figura 4 Arduino Mega



**Arduino Nano:** Versiones más compactas, ideales para proyectos donde el espacio es una limitación. Estas placas incluyen conexiones para alimentación, comunicación con otros dispositivos, y pines de entrada/salida para controlar periféricos, existen varios modelos.

Figura 5 Arduino Nano



#### 2.4. Aplicaciones de Arduino en Sistemas Embebidos

Arduino es ampliamente utilizado en el desarrollo de proyectos que requieren sistemas embebidos, algunos ejemplos son:

- a) **Domótica:** Control de luces, sistemas de seguridad, calefacción y ventilación mediante sensores y actuadores.
- b) **Robótica:** Desarrollo y diseño de robots controlados por sensores, motores y actuadores para diferentes tareas, como seguimiento de líneas o robótica educativa.
- c) **Dispositivos Médicos:** Monitorización de signos vitales y dispositivos portátiles (wearables) que recopilan datos del cuerpo humano
- d) **IoT (Internet de las Cosas):** La conectividad de Arduino permite desarrollar y diseñar dispositivos que pueden comunicarse a través de Internet, enviando y recibiendo datos.
- e) **Educación:** En entornos educativos, Arduino se utiliza para enseñar conceptos de programación, electrónica y diseño de prototipos de sistemas embebidos.

## CAPÍTULO 3

### Impacto en la Educación y la Industria

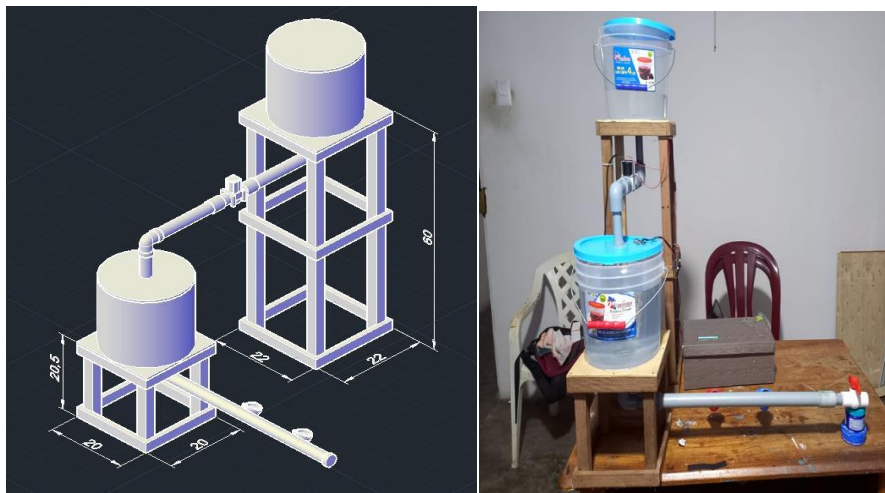
#### 3.1. Educación

Arduino ha transformado la enseñanza de la electrónica y la programación. Su enfoque práctico y accesible permite a los estudiantes experimentar y aprender de manera efectiva. Existen numerosos recursos en línea, tutoriales y comunidades que apoyan el aprendizaje de Arduino.

Se presenta algunos ejemplos de su aplicación desarrollados en el curso de programación de ingeniería:

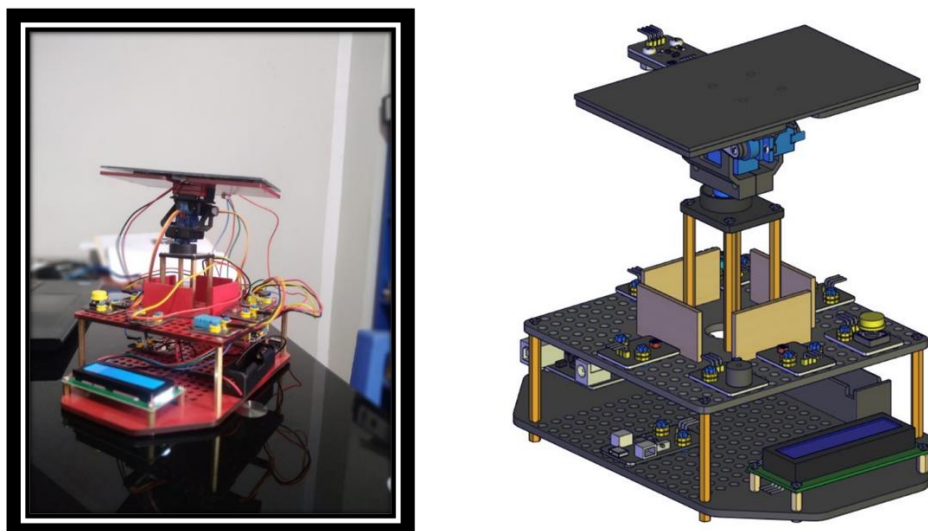
- a) Diseño de un prototipo automatizado para un bebedero de gallinas

Figura 6 Proyecto 1



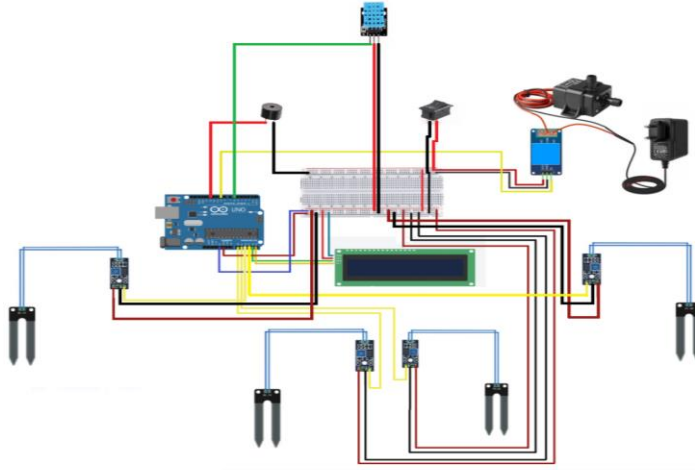
- b) Diseño de un prototipo automatizado para cargar dispositivos inteligentes

Figura 7 Proyecto 2



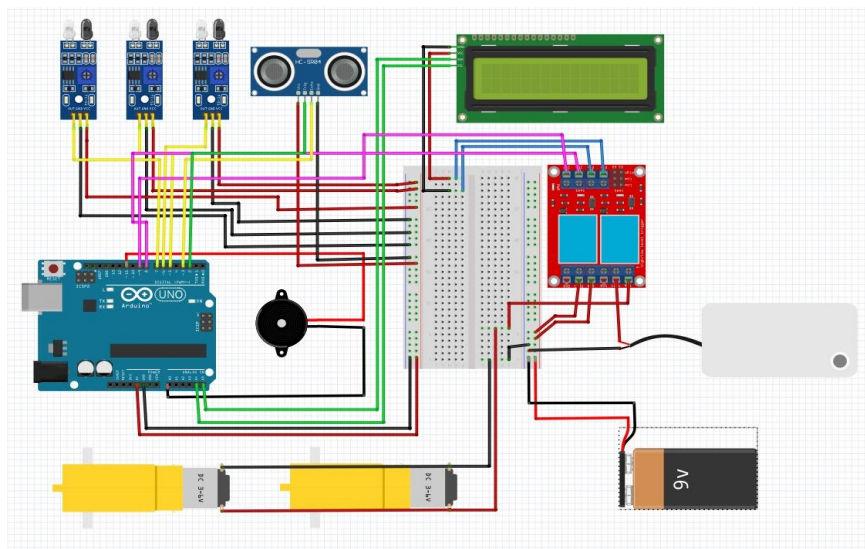
- c) Diseño de un prototipo de riego Automatizado para los jardines de la UNJ

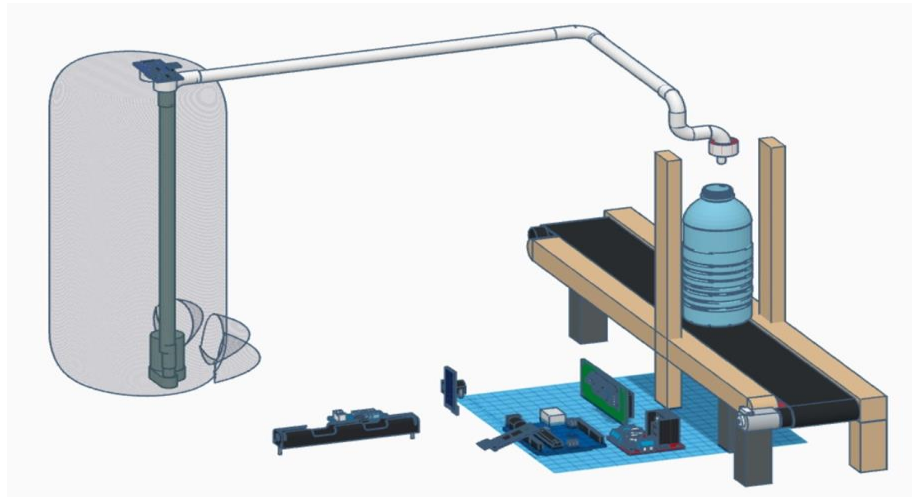
Figura 8 Proyecto 3



- d) Diseño de un prototipo de una faja transportadora automatizada para el embazado de líquidos

Figura 9 Proyecto 4





### 3.2. Industria

En el ámbito industrial, Arduino ha influido en el desarrollo de prototipos y en la creación de productos innovadores. Muchas startups y empresas utilizan Arduino para desarrollar soluciones rápidas y económicas, lo que acelera el proceso de innovación.

## CAPÍTULO 4

### Desafíos y Limitaciones de Arduino

#### 4.1. Limitaciones de Hardware

A pesar de sus ventajas, Arduino también presenta algunos desafíos y limitaciones que se deben de considerar:

##### a) Limitaciones de Hardware

Recursos Limitados: Las placas Arduino suelen tener recursos limitados en términos de almacenamiento, memoria y potencia de procesamiento, lo que puede restringir la complejidad de los proyectos. Para aplicaciones más exigentes, puede ser necesario recurrir a plataformas más avanzadas como los PLCs.

##### b) Falta de Soporte de Tiempo Real:

Arduino no está diseñado para aplicaciones de tiempo real críticas. Si un proyecto requiere un control preciso en tiempo real, se podría considerar utilizar microcontroladores más sofisticados o sistemas operativos de tiempo real.

#### 4.2. Progreso del Aprendizaje con Arduino

El progreso del aprendizaje puede ser empinada para aquellos que no tienen antecedentes en programación o electrónica. Es fundamental invertir tiempo en la comprensión de los conceptos básicos, para poder iniciar con algún proyecto.

La depuración de errores en proyectos complejos puede ser un desafío. La falta de herramientas avanzadas de depuración en el IDE de Arduino puede dificultar la identificación de problemas.

#### 4.3. Futuro de Arduino y el Desarrollo de Sistemas Embebidos

El futuro de Arduino y la programación embebida se presenta prometedor, impulsado por tendencias emergentes y avances tecnológicos.

##### a) Integración con IoT y Machine Learning

El creciente interés en el Internet de las Cosas (IoT) y el aprendizaje automático está abriendo nuevas oportunidades para Arduino. Se están desarrollando nuevas bibliotecas y plataformas que permiten a los dispositivos Arduino integrarse fácilmente en redes IoT y realizar tareas de procesamiento de datos utilizando modelos de machine learning.



### **b) Prototipado Rápido**

Arduino seguirá siendo una herramienta clave para el prototipado rápido. Con la evolución de la tecnología y la creciente demanda de dispositivos personalizados, Arduino permitirá a los innovadores y desarrolladores crear y probar prototipos de manera ágil y eficiente.

### **c) Comunidad y Soporte**

La comunidad de Arduino, que ha crecido enormemente, seguirá siendo un pilar fundamental. La colaboración abierta y el intercambio de conocimientos entre usuarios y desarrolladores fomentarán la innovación y la mejora continua de la plataforma.

### **d) Impacto**

Arduino ha transformado la educación en electrónica, permitiendo un aprendizaje práctico y accesible. En la industria, facilita el prototipado rápido y la innovación tecnológica. Aunque presenta algunas limitaciones en recursos y manejo de errores, su comunidad activa y su continua evolución aseguran que seguirá siendo relevante en el futuro del desarrollo de sistemas embebidos.



## CONCLUSIÓN

Arduino ha revolucionado el campo de del desarrollo de la programación de sistemas embebidos, facilitando el acceso a la tecnología tanto para principiantes como para expertos. Su simplicidad, combinada con su potencia y flexibilidad, lo convierte en una herramienta esencial en la educación y en la industria. A medida que la tecnología avanza, es probable que Arduino continúe teniendo mejoras y desempeñando un papel clave en el desarrollo de nuevas aplicaciones y soluciones innovadoras.



## REFERENCIAS BIBLIOGRÁFICAS

Arduino. (07 de 10 de 2024). *Arduino*. Obtenido de <https://www.arduino.cc/reference/es/>