Materia:

DISEÑO ELECTRÓNICO BASADO EN SISTEMAS EMBEBIDOS

Alumno:

Posadas Pérez Isaac Sayeg

Paniagua Rico Juan Julian

García Azzúa Jorge Roberto

Grado y grupo:

8°G

Profesor:

Garcia Ruiz Alejandro Humberto

Tarea 3:

Arduino Fundamentos y Características (Variables, tipos de datos, estructuras de control, etc.)

Arduino: Fundamentos y Características

Arduino es una plataforma de hardware libre que se utiliza principalmente para la creación de proyectos electrónicos. Su simplicidad y accesibilidad lo han convertido en una herramienta popular tanto para principiantes como para expertos en electrónica y programación. En esta investigación, abordaremos los fundamentos y características de Arduino, centrándonos en las variables, tipos de datos, estructuras de control y otros conceptos clave.

1. Introducción a Arduino

Arduino es una plataforma basada en microcontroladores que permite a los usuarios interactuar con el mundo físico mediante la creación de dispositivos electrónicos. Consiste en una placa de circuito (hardware) y un entorno de desarrollo integrado (IDE) que permite escribir y cargar código (software) en la placa. La programación en Arduino se basa en un dialecto del lenguaje de programación C/C++.

1.1. Componentes Principales

- **Placas Arduino**: Existen varias versiones de placas Arduino, siendo las más populares Arduino Uno, Arduino Mega y Arduino Nano. Cada una tiene características específicas, como la cantidad de pines de entrada/salida, capacidad de memoria y funcionalidades adicionales.
- **Entorno de Desarrollo (IDE)**: Es un software que permite escribir, compilar y cargar programas en la placa Arduino. El IDE proporciona una interfaz amigable y fácil de usar, con ejemplos y bibliotecas preinstaladas.

2. Variables en Arduino

Las variables son contenedores que almacenan datos que pueden cambiar durante la ejecución del programa. En Arduino, se pueden declarar variables de diferentes tipos, según la naturaleza de los datos que se desean almacenar.

2.1. Tipos de Datos

Arduino admite varios tipos de datos, que se pueden clasificar en dos categorías: tipos de datos primitivos y tipos de datos compuestos.

2.1.1. Tipos de Datos Primitivos

- **int**: Almacena números enteros. Rango de -32,768 a 32,767.
- **float**: Almacena números de punto flotante (decimales). Utiliza 4 bytes de memoria.
- **char**: Almacena un solo carácter. Utiliza 1 byte de memoria.
- **boolean**: Almacena valores de verdadero (true) o falso (false).

2.1.2. Tipos de Datos Compuestos

- **Arrays**: Permiten almacenar múltiples valores del mismo tipo en una sola variable. Por ejemplo, `int numeros[5]` puede almacenar cinco enteros.

Estructuras (struct): Permiten agrupar diferentes tipos de datos en una sola variable, creando un nuevo tipo de dato. Por ejemplo:

```
struct Persona {
   String nombre;
   int edad;
};
```

3. Estructuras de Control

Las estructuras de control son fundamentales en la programación, ya que permiten tomar decisiones y ejecutar bloques de código basados en condiciones específicas. En Arduino, se utilizan varias estructuras de control comunes.

3.1. Estructuras de Control Condicionales

3.1.1. if y else

La estructura `if` evalúa una condición y ejecuta un bloque de código si la condición es verdadera. La estructura `else` se ejecuta si la condición es falsa.

```
if (temperatura > 30) {
```

```
    // Encender ventilador
} else {
    // Apagar ventilador
}
```

3.1.2. switch

La estructura `switch` permite evaluar una variable contra múltiples valores. Es útil cuando hay varias condiciones posibles.

```
switch (opcion) {
    case 1:
        // Opción 1
        break;
    case 2:
        // Opción 2
        break;
    default:
        // Opción por defecto
}
```

3.2. Estructuras de Control de Repetición

3.2.1. for

El bucle `for` se utiliza para repetir un bloque de código un número específico de veces.

```
for (int i = 0; i < 10; i++) {
    // Ejecutar código 10 veces
}
```

3.2.2. while

El bucle `while` repite un bloque de código mientras una condición sea verdadera.

```
while (estado == true) {
    // Ejecutar código mientras estado sea verdadero
}
```

4. Funciones

Las funciones son bloques de código que se pueden reutilizar. Se pueden definir funciones para organizar el código y hacer que sea más legible y modular. Una función se define especificando su tipo de retorno, nombre y parámetros.

```
int sumar(int a, int b) {
    return a + b;
}
```

5. Conclusiones

Arduino es una plataforma que es bastante accesible que permite a los usuarios interactuar con el mundo físico a través de la programación y la electrónica. Es importante comprender los fundamentos de las variables, tipos de datos y estructuras de control es esencial para desarrollar proyectos exitosos en Arduino. Estas características facilitan la creación de programas eficientes y bien estructurados, lo que permite a los usuarios realizar tareas complejas de manera efectiva.

Bibliografía

1. Banzi, M., & Shiloh, M. (2014). *Getting Started with Arduino*. Maker Media, Inc.
    - Este libro proporciona una introducción a la plataforma Arduino, que incluye fundamentos y ejemplos de programación.
2. Monk, S. (2015). *Programming Arduino: Getting Started with Sketches*. McGraw-Hill Education.
    - Un recurso que presenta ejemplos prácticos sobre la programación en Arduino, incluidas variables y estructuras de control.
3. Margolis, M. (2011). *Arduino Cookbook*. O'Reilly Media.
    - Un libro que ofrece una colección de recetas y ejemplos prácticos para trabajar con Arduino en diferentes proyectos.
4. Arduino. (n.d.). *Arduino Reference*. Retrieved from https://www.arduino.cc/reference/en/
    - Documentación oficial de Arduino que detalla las funciones, variables y estructuras de control en el lenguaje de programación de Arduino.
5. Adafruit. (n.d.). *Learning System*. Retrieved from https://learn.adafruit.com/
    - Un recurso en línea con tutoriales sobre programación de Arduino y electrónica.