

Dos casos de generación de código de sistemas embebidos con Inteligencia Artificial

Armando-Martínez Reyes¹, Roxana-García Andrade¹,
Ernesto Rincón¹, Carlos-Hernández Santos¹, Arnoldo-
Fernández Ramírez¹, Dagoberto-Pamanes Hernández¹,
Nain de la Cruz¹

¹Tecnológico Nacional de México, Campus Nuevo León; Departamento de Ingeniería Eléctrica y Electrónica, Av. Eloy Cavazos 2001, Toluca, Guadalupe, N.L. 67170, México. Email principal: armando.mr@nuevoleon.tecnm.mx

Resumen

En este trabajo de investigación, se presentan dos casos de generación de código para dos dispositivos de sistemas embebidos, con ayuda de la Inteligencia Artificial, ChatGPT. El objetivo de esto es, presentar como las herramientas basadas en IA pueden ayudar, a generar el código funcional en los dispositivos embebidos. Así mismo resaltar el uso de la IA como herramienta para reducir los tiempos de desarrollo de la programación de sistemas embebidos, analizando dos casos en esta ocasión. El primero usando el ESP8266 para el internet de las cosas IoT, y un sensor de temperatura y humedad, el DHT11 para monitorear estas variables en tiempo real, y visualizarlas en una página web. Y como un segundo caso El microcontrolador Arduino Uno, el mismo sensor DHT11 para monitorear, temperatura y humedad, con la diferencia, que los resultados se observen en una pantalla Oled. Los resultados obtenidos fueron óptimos, sin embargo, se encontró que, en cuanto a limitaciones, se requieren revisar las actualizaciones de algunos drives y al hacer la solicitud a la IA se debe de especificar lo más posible, modelos, tipos y configuraciones que se requieran. Es decir, dar instrucciones más elaboradas que incluyan detalles específicos para obtener una respuesta más precisa. Esto es, las instrucciones deben ser claras, directas, específicas y con contexto.

Palabras clave—Sistemas embebidos, Internet de las cosas, ESP8266, Microcontrolador, Arduino Uno.

Abstract

In this research work, two cases of code generation for two embedded systems devices are presented, with the help of Artificial Intelligence, ChatGPT. The objective of this is to present how AI-based tools can help generate functional code on embedded devices. Likewise, highlight the use of AI as a tool to reduce development times for programming embedded systems, analyzing two cases on this occasion. The first using the ESP8266 for the Internet of Things IoT, and a temperature and humidity sensor, the DHT11 to monitor these variables in real time, and display them on a web page. And as a second case, the Arduino Uno controller, the same DHT11 sensor to monitor temperature and humidity, with the difference that the results are observed on an Oled screen. The results obtained were excellent, it was

only found that, in terms of limitations, it is necessary to review the updates of some drives and when making the request to the AI, the models, types and configurations that are required must be specified as much as possible. That is, give more elaborate instructions that include specific details to obtain a more precise response. That is, the instructions must be clear, direct, specific and with context

Keywords— *Embedded systems, Internet of things, ESP8266, Microcontroller, Arduino Uno.*

1. INTRODUCCIÓN

La inteligencia artificial (IA) es la implementación de programas computacionales que emulan el pensamiento, comportamiento y razonamiento humano, es decir, la inteligencia humana, así la describe [1]. La inteligencia Artificial ya se encuentra inmersa en nuestro ámbito diario, ejemplo de ello, en las sugerencias que nos dan en la selección de una película, en las plataformas, Alexa, Siri, por mencionar algunas. ¿Cómo lo menciona [2], será necesario limitar a la IA?. Surge entonces esta preocupación, lo que sugieren, “asegurar principios éticos y valores muy claros”, en el desarrollo y uso de las tecnologías, para enfocarlos en beneficios de la humanidad, como en la medicina, el medio ambiente, la educación. Por otro lado, está, la optimización de compiladores con el uso de machine learning, como lo expresa en [3]. Se concluye que esto ha tomado mucho auge, el uso de algoritmos de inteligencia artificial, que se automatizan, para optimizar las funciones del compilador. Es necesario tener un enfoque equilibrado y responsable de la integración de la IA en la industria así lo concluye [4]. Otra aplicación más de la IA se encuentra en los sistemas de visión, de acuerdo a [5], nos enfrentamos a un panorama cambiante lo que sugiere una mayor investigación en este enfoque. Así mismo la IA empleada en reconocimiento facial, en conjunto con AWS (Amazon Web Service) permiten identificar a las personas y devuelve esta información al sistema embebido, esto de acuerdo a [6]. Esto ayuda al sistema en el proceso de reconocimiento facial. La IA también ha logrado evaluar situaciones de riesgo en el sistema bancario, así se muestra en [7] en su trabajo. Actualmente consideramos, el uso de la IA como una herramienta en la enseñanza de la ingeniería. Muestra de ello son los dos casos que presentamos en este artículo. Donde estos circuitos son empleados en las materias de sistemas embebidos, microcontroladores de los programas de las carreras de Ingeniería Electrónica y Mecatrónica. Los sistemas embebidos son sistemas incrustados, diseñados para realizar trabajos específicos dentro de otro dispositivo mayor. Están integrados por un hardware especializado y programado, para ejecutar funciones de manera eficiente. Combinan componentes de hardware y software, de tal manera que se integran en un sistema con un entorno específico, así lo menciona [8]. Los sistemas embebidos requieren algunos elementos básicos, entre ellos un micro controlador, que es el corazón del sistema. Este a su vez puede contener, unidad central de proceso, memoria RAM y

ROM o flash, interfaces de entradas y salidas, en la figura 1, se muestra que los dispositivos embebidos interactúan con la memoria, interfaces de entrada y salida así como el software utilizado para realizar las funciones de control y comunicación. Este dispositivo se conecta externamente con periféricos como sensores, actuadores, pantallas por mencionar algunas para interactuar con el entorno. Otro elemento básico del sistema es el Software embebido que consiste en, instrucciones específicas para la aplicación. Los sistemas embebidos están presentes en nuestra vida cotidiana, en el área automotriz, se puede mencionar el sistema de frenos ABS, bolsas de aire. Mientras que el área de electrodomésticos inteligentes como son: televisores, lavadoras, microondas. De igual manera en teléfonos, relojes inteligentes. En el área de automatización industrial en controladores lógicos programables (PLC) sensores inteligentes. Y por supuesto en el internet de las cosas IoT en cámaras de seguridad, aires acondicionados por mencionar algunos. La idea de emplear diferentes embebidos es para ver las diferencias entre cada uno y al momento de elegir, de acuerdo al proceso, elegir el apropiado y eficiente para ello.



Figura 1: Sistemas embebidos

Esta investigación se enfoca en clasificar 3 áreas de grandes familias.

1.-Microcontroladores; para ello se realizaron pruebas con uno de los más populares que es el Arduino Uno Revisión 3, teniendo en cuenta que existe una gran cantidad de dispositivos similares de otras marcas, como se puede ver en [9]. 2.- Para los dispositivos del internet de las cosas (IoT), donde se observa que evolucionan rápidamente, teniendo dispositivos que se conectan a la red, así lo menciona [10]. Se utilizaron dispositivos de la familia ESP tales como el ESP8266 y SP32, estos dispositivos fueron diseñados para Internet de las cosas es un microcontrolador que tiene capacidades de WiFi y bluetooth, así lo explica[11], los cuales también son una muestra representativa de una gran cantidad de componentes similares de otras marcas. 3.- En el área donde se requiere la aplicación de diseños digitales combinacionales y secuenciales los dispositivos denominados FPGA (Field Programmable Gate Array) el cual utiliza lenguajes de implementación de hardware como el denominado VHDL (Very High Speed Integrated Circuits) este es un lenguaje descriptivo de hardware de diseños digitales. Esta clasificación es particular de los autores considerando tres grandes áreas de aplicación de la

electrónica digital, las cuales tienen sus diferencias muy particulares. En la tabla 1 se muestran algunas características.

Tabla 1. Clasificación de Dispositivos Embebidos

Dispositivo	Aplicaciones	Lenguaje de programación
Microcontrolador	En el desarrollo de prototipos donde no es requerido un diseño digital formal, su enfoque es didáctico.	Ambiente IDE (Ambiente de desarrollo) de programación Arduino
Dispositivos IoT	Por sus reducidas dimensiones y capacidad de comunicación inalámbrica su aplicación en el internet de las cosas permite control y monitoreo en aplicaciones de tipo; automotriz, medicina, industriales, electrodomésticos inteligentes etc.	Lenguaje propietario de las marcas, Phyton, Visual Basic y ambiente similar dentro del entorno IDE arduino, este último es el que se utilizó en esta investigación.
Dispositivos FPGA y VHDL	En el área donde se requiere diseños digitales complejos y de grandes dimensiones tales como los procesadores de las LAPTOP y computadoras y desarrollos industriales complejos son utilizados estos dispositivos	Lenguaje VHDL y lenguaje Verilog, para este proyecto se utilizó lenguaje VHDL

En estas tres grandes familias de dispositivos el elemento común es la necesidad de tener un programa (generación de un código) con lo cual se ve de gran utilidad el empleo de la inteligencia artificial para generar dichos códigos. Es importante mencionar que esta herramienta servirá de apoyo para las aplicaciones finales de estos dispositivos debiendo ser revisados, corregidos y optimizados para su uso final. Para los casos a presentar a continuación se observó que la IA (inteligencia artificial) no sustituye a la persona experta (alumno) será utilizada como una herramienta de apoyo. En esta investigación se implementaron varios circuitos electrónicos en el laboratorio con dos diferentes dispositivos embebidos mencionados y se recopiló la generación de código con una inteligencia artificial que en este caso fue, el CHATGPT 4 en su versión gratuita sin

entrenamiento previo en cada dispositivo, analizando sus resultados. La intención es observar como la inteligencia artificial impacta en estos dispositivos directamente en su programación. De tal manera que con la inteligencia artificial tenemos una gran herramienta para la programación. Que permite ahorrar una gran cantidad de tiempo, donde el papel como ingenieros es plantear adecuadamente los requerimientos del proyecto y posteriormente analizar los resultados para su depuración. Para lograr una adecuada depuración de un proyecto el ingeniero debe de tener bases sólidas de conocimientos básicos.

2. DESARROLLO

2.1 Sistema embebido con Microcontrolador ESP8266 y sensor DHT11

El primer caso que se presenta es un microcontrolador para IoT conocido comercialmente como ESP8266. Se pretende analizar y comprobar específicamente si la IA nos puede ayudar a generar código funcional en la aplicación de este dispositivo embebido. El objetivo es monitorear temperatura y humedad y observar los datos en una página Web, para realizar un histórico de estos datos. Como primer punto se realizó una descripción de los pasos y como se realiza la programación de este dispositivo embebido sin la ayuda de la IA. Para dar inicio se requiere que el alumno conozca las especificaciones del sensor DHT11 en la figura 7 se observa el diagrama de conexiones del sensor con el microcontrolador, que básicamente son la alimentación de CD de 5volts y un solo pin de las salida de las mediciones. Un solo pin porque lo hace en forma serial. Este sensor mide humedad y temperatura simultáneamente. Además, se deberá tener conocimiento de programación del microcontrolador ESP8266, el cual se desarrolla dentro del ambiente de programación del Arduino, para esta aplicación, es necesario, contar con la librería del sensor y la librería denominada WiFi, las cuales facilitan la toma de lecturas, y su envío de datos vía inalámbrica. Las mediciones se enviaran en forma inalámbrica mostrándose en una página web de un navegador de internet. Estos son los requerimientos para elaborar el código, sin ayuda de la IA. Ahora se solicitó a la IA [12] que genere código para microcontrolador ESP8266 y que utilice el sensor DHT11 para monitorear la temperatura y humedad y a su vez muestre los datos en una página web.

elabora código para microcontrolador esp8266 que utiliza un sensor de humedad y temperatura dht11 mostrando las lecturas en una pagina web

¡Claro! A continuación, te proporcionaré un ejemplo de código para un ESP8266 que utiliza un sensor de humedad y temperatura DHT11 y muestra las lecturas en una página web sencilla.

Figura 2. Se le solicita a la IA el código

En la figura 2 se observa el texto donde se le solicita, a la IA que genere el código con la descripción, el modelo del microcontrolador y los sensores como se utilizaran.

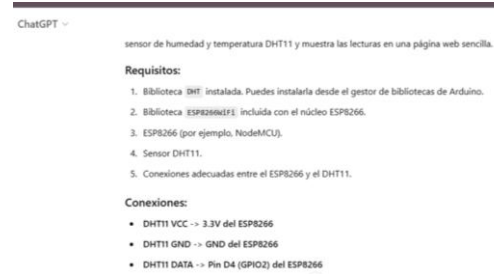


Figura 3. Descripción de requisitos de la IA

Como respuesta de la IA, figura 3. La IA describe los requisitos básicos para que la práctica funcione, esto es, las librerías tanto del microcontrolador, como de los sensores, a su vez nos da la descripción de las conexiones (hardware) de los sensores y el microcontrolador. Para posteriormente generar el código ver. En la figura 4 se observa el inicio del código donde aparecen las librerías del sensor y la comunicación Wi Fi para realizar las mediciones.

Código:

```
cpp
#include <ESP8266WiFi.h>
#include <DHT.h>

// Definición de pines y tipo de sensor
#define DHTPIN D4 // Pin donde está conectado el DHT11
#define DHTTYPE DHT11 // Definiendo el tipo de sensor

DHT dht(DHTPIN, DHTTYPE);
```

Figura 4. La IA genera el código

La IA define, de acuerdo al microcontrolador ESP8266 y al sensor DHT los pines o puntos de conexión, y a su vez el tipo de sensor, que en este caso es de temperatura y humedad DHT11. En la figura 4, la IA describe la configuración del servidor web, que consiste en habilitar el puerto 80 que corresponde al protocolo HTTP referido a la página web mencionada.

```
// Configuración del servidor web
WiFiServer server(80);

void setup() {
  // Inicializar el puerto serie
  Serial.begin(115200);
  delay(10);
  Serial.println("DHT11 sensor example using ESP8266");

  // Inicializar el sensor DHT11
  dht.begin();

  // Conectarse a la red WiFi
  Serial.println();
  Serial.print("Conectando a ");
  Serial.println(ssid);
```

Figura 5. Configuración puerto serial

En esta misma figura, se muestra, la configuración de la página web donde se mostraran los datos obtenidos de monitoreo de temperatura y humedad. De igual forma la inicialización del sensor. En la figura 6, se declara abrir una terminal serial donde te dará la información de la conexión incluyendo la denominada dirección IP, que corresponde al protocolo TCP/IP.

```
Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

Serial.println("");
Serial.println("WiFi conectado");
Serial.println("Dirección IP: ");
Serial.println(WiFi.localIP());

// Iniciar el servidor
server.begin();
Serial.println("Servidor iniciado");
}
```

Figura 6. Configuración de TCP/IP

Al concluir con el código se realizaron las conexiones indicadas por la IA. Como se muestra en la figura 7. Donde se observa, el sensor conectado a los pines del microcontrolador.

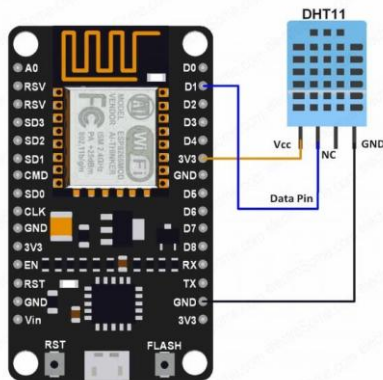


Figura 7 Conexiones del sensor al ESP8266

Se sube el código a la plataforma de Arduino, para posteriormente subirlo al ESP8266, como se observa en la figura 8. Una vez que se ejecuta el dispositivo genera una dirección IP, la cual puede ser consultada remotamente en forma inalámbrica colocando esta dirección en un navegador web.

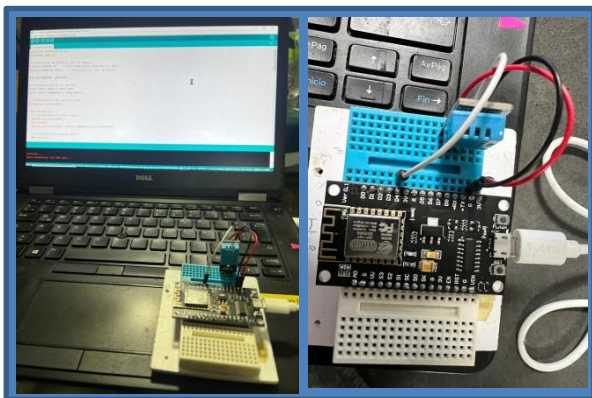


Figura 8. Código en plataforma Arduino y Circuito ESP8266

En la figura 9 se muestra la dirección IP que se obtiene. Al obtener la dirección la ponemos en un navegador. Y despliega las lecturas de las variables.

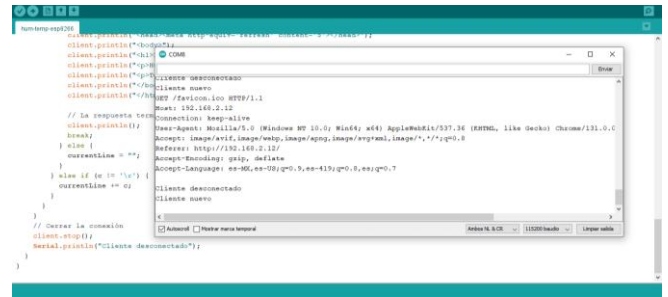


Figura 9. Dirección IP

A continuación en la figura 10 se muestra la lectura de humedad y temperatura, obtenida con código generado por la IA. Cabe mencionar que estas lecturas son en tiempo real.



Figura 10. Lectura de medición

2.2 Sistema embebido con Microcontrolador Arduino UNO

El segundo caso, tiene como objetivo monitorear dos variables que son, temperatura y humedad y mostrar resultados en una pantalla Oled (led organico) I2C. En este caso se utilizó un microcontrolador Arduino Uno rev. 3. Se comprobó si la IA es capaz de generar sin errores los códigos para programar este dispositivo embebido. Lo primero se define el modelo del sensor de temperatura y humedad como en el caso anterior se utilizó el DHT11 y el modelo de la pantalla Oled Adafruit SSD1306. Se describe el procedimiento que genera la programación de este microcontrolador sin la ayuda de la IA. Para dar inicio es necesario que se conozcan las especificaciones del sensor como se mostró en el caso anterior. Así como el entorno y lenguaje de programación del microcontrolador Arduino uno rev. 3. Con las especificaciones, del sensor y microcontrolador se procede a realizar las conexiones de los mismos. Para la ejecución de este código es necesario la instalación de las siguientes librerías para el sensor de humedad y temperatura la librería DHT11 y para la pantalla Oled las librerías, Adafruit SSD1306, Adafruit GFX. Para la pantalla que se está usando se define su resolución. A continuación, se realiza la solicitud a la IA ChatGPT 4 para la generación del código, observando que la solicitud se realizó con faltas de ortografía. Sin embargo, genera el código y hace mención que este ejemplo utilizó las librerías Adafruit. En la figura 11 se observa el texto, como se solicitó, la generación del código a la IA.

En la siguiente parte de código , en la figura 14 se muestran las instrucciones para que se despliegue en la pantalla Oled las mediciones de humedad y temperatura en el formato solicitado siendo este un ciclo repetitivo.

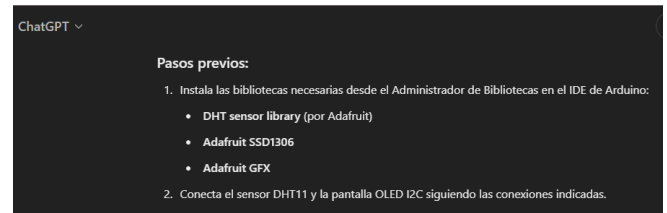


Figura 11. Se hace la solicitud a la IA

Al generar el código, la IA figura 12 se declaran las librerías Adafruit y DHT y realiza la configuración del sensor en el pin 2 y el tipo de sensor DHT11. De igual manera la resolución de la pantalla Oled, Ancho y Alto de la misma, siendo, 128 x 64.

Código Arduino

```
cpp Copiar Editar

#include <Adafruit_SSD1306.h> // Librería para la pantalla OLED
#include <Adafruit_GFX.h> // Librería base para gráficos
#include <DHT.h> // Librería para el sensor DHT

// Configuración del DHT11
#define DHTPIN 2 // Pin al que está conectado el DHT11
#define DHTTYPE DHT11 // Tipo de sensor DHT
DHT dht(DHTPIN, DHTTYPE);

// Configuración de la pantalla OLED
#define SCREEN_WIDTH 128 // Ancho de la pantalla OLED
#define SCREEN_HEIGHT 64 // Alto de la pantalla OLED
#define OLED_RESET -1 // No se necesita reset hardware para algunas pantallas
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

Figura 12. Muestra el código que generó la IA

En la siguiente figura 13, del código generado se muestran los comandos de despliegue de la información. Así como, la inicialización de la comunicación a una velocidad de 9600 bauds y a su vez se inicializa el display

```
Copiar Editar

void setup() {
  // Inicializar el puerto serie
  Serial.begin(9600);

  // Inicializar el sensor DHT11
  dht.begin();

  // Inicializar la pantalla OLED
  if (!display.begin(SSD1306_I2C_ADDRESS, 0x3C)) { // Dirección I2C 0x3C por defecto
    Serial.println(F("No se encontró la pantalla OLED"));
    for (;;) // Bucle infinito si falla
  }
  display.clearDisplay();
  display.setTextColor(SSD1306_WHITE); // Texto blanco
  display.setTextSize(1); // Tamaño del texto
  display.setCursor(0, 0); // Cursor al inicio
  display.println("Inicializando...");
  display.display(); // Mostrar el mensaje
  delay(2000); // Pausa de 2 segundos
}
```

Figura 13. Se inicia comunicación a 9600 bauds

```
Copiar Editar

void loop() {
  // Leer temperatura y humedad del sensor
  float temperatura = dht.readTemperature();
  float humedad = dht.readHumidity();

  // Comprobar si la lectura es válida
  if (!isnan(temperatura) || !isnan(humedad)) {
    Serial.println(F("Error al leer el sensor DHT11"));
    display.clearDisplay();
    display.setCursor(0, 0);
    display.println("Error leyendo");
    display.println("Sensor DHT11");
    display.display();
    delay(2000); // Esperar antes de reiniciar
    return;
  }

  // Mostrar datos en la pantalla OLED
  display.clearDisplay();
  display.setCursor(0, 0);
  display.println("Temperatura");
  display.println(temperatura);
  display.println("Humedad");
  display.println(humedad);
  display.display();

  // Esperar 2 segundos antes de la siguiente lectura
  delay(2000);
}
```

Figura 14. Finaliza el código

Figura 15. La IA nos muestra los pasos previos

Como siguiente paso en la figura 15 la IA nos muestra que debemos cargar las librerías en el ambiente de desarrollo de Arduino (IDE). Y posteriormente conectar el sensor y la pantalla Oled, de acuerdo a las conexiones proporcionadas por la misma, al final del código. En la siguiente figura 16, se observan los requisitos previos que se requieren para ejecutar el código. Siendo estas, las librerías necesarias y las conexiones y además la IA sugiere verificar que la pantalla Oled tenga cierta resolución y menciona que si no lo es, modificar el código en la línea especificada como se aprecia esto se aprecia en la figura 16.

Requisitos:

1. Librerías necesarias:
 - DHT para el sensor. Puedes instalarla desde el Administrador de Librerías en Arduino IDE.
 - Adafruit_SSD1306 y Adafruit_GFX para la pantalla OLED. También puedes instalarlas desde el Administrador de Librerías.
2. Conexiones:
 - Conecta el pin de datos del DHT11 al pin digital 2 del Arduino.
 - Conecta la pantalla OLED a los pines I2C:
 - SDA al pin A4 en un Arduino UNO (o al pin correspondiente en otros modelos).
 - SCL al pin A5 en un Arduino UNO.
3. Dirección I2C:
 - Asegúrate de que la dirección I2C de tu pantalla OLED sea 0x3C. Si no lo es, ajusta la línea `display.begin(SSD1306_I2C_ADDRESS, 0x3C);` con la dirección correcta.

Figura 16. Se muestra el código generado.

Como siguiente paso en la figura 17, se observa la declaración de la resolución de la pantalla Oled.

```
Copiar Editar

// Inicializar la pantalla OLED
if (!display.begin(SSD1306_I2C_ADDRESS, 0x3C)) { // Dirección I2C 0x3C por defecto
  Serial.println(F("No se encontró la pantalla OLED"));
  for (;;) // Bucle infinito si falla
}
```

Figura 17. Configuración pantalla Oled

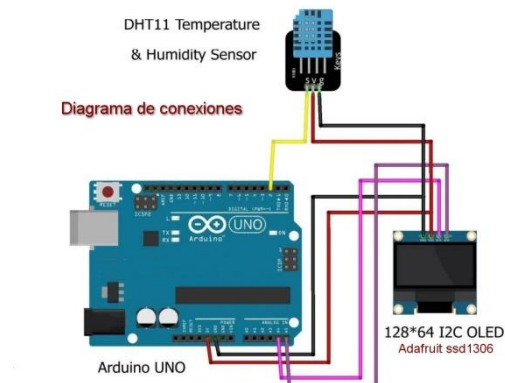


Figura 18. Conexiones del microcontrolador

A continuacion se realizaron las conexiones que indico la IA en la figura 16. De esta manera queda conectado el hardware. Se corre el programa y las lecturas de humedad y temperatura se aprecian en la pantalla Oled figura 19.

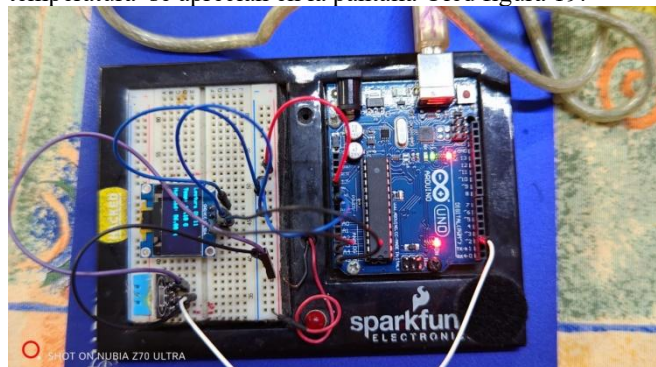


Figura 19. Muestra la lectura de humedad y temperatura en el display OLED.

3. CONCLUSIONES Y RECOMENDACIONES

El uso de la IA en sistemas embebidos representa un avance significativo para aplicaciones como IoT. La IA es una herramienta que agiliza los tiempos del desarrollo de código y se debe considerar como un apoyo y no una solución final. Los mejores resultados se obtienen cuando especificamos a la IA detalladamente los requerimientos del proyecto. Se puede apreciar como la IA ayuda en la simplificación del proceso, cuando son desarrolladores iniciales. Es necesario, en ocasiones, adaptar el código que genero la IA para requerimientos específicos. Para estos casos la depuración requerida fue revisar las versiones requeridas por las librerías del sensor DHT11. La IA puede no considerar todos los requerimientos de Hardware que se necesitan. Es por ello que el usuario requiere conocimientos básico o elementales de las especificaciones o requerimientos que se le solicitan a la IA.

3.1 Trabajos a futuro.

Como una segunda parte a esta investigación se realizaran más ejemplos de generación de códigos con otros dispositivos embebidos y sus aplicaciones en la rama de la enseñanza de la ingeniería. Alguno de estos son el FPGA, Raspberry pi y VHDL. Y a su vez otras IA como chat.gpt o qwenlm.ai.

4. REFERENCIAS

- [1] J. Granados Ferreira, «Análisis de la inteligencia artificial en las relaciones laborales», *rev.ces derecho*, vol. 13, n.º 1, pp. 111–132, may 2022.
- [2] P. E. Ruiz Avances en inteligencia artificial y su impacto en la sociedad. [online]. Disponible en: <http://hdl.handle.net/20.500.11912/4942.C>.
- [3] J. Zaldaña, Optimizacion de compiladores con el uso de machine learning. 10.13140/RG.2.2.14771.20004. (2023).

- [4] C. Chen Cheng, E. Chung, y N. Correa, «La inteligencia Artificial y su Impacto en la Industria de la Ingeniería», *REICT*, vol. 3, n.º 1, pp. 26–40, jul. 2023.
- [5] Fabre W, Haroun K, Lorrain V, Lepecq M, Sicard G. From Near-Sensor to In-Sensor: A State-of-the-Art Review of Embedded AI Vision Systems. *Sensors*. 2024; 24(16):5446. <https://doi.org/10.3390/s24165446>
- [6] E. Saavedra Quijada, L. A., Medina Muñoz, F., Morales Solís, & G., López Valencia, (2023). Reconocimiento facial usando herramientas de IA de Amazon Web Services y sistemas embebidos. *Research in Computing Science*, 152(8),69-77. https://rcs.cic.ipn.mx/2023_152_8/Reconocimiento%20facial%20usando%20herramientas%20de%20IA%20de%20Amazon%20Web%20Services%20y%20sistemas%20embebidos.pdf
- [7] A., E.,Marín Herrera. Implementación de un sistema de seguridad bancario en sistemas embebido utilizando inteligencia artificial [Trabajo de Grado Pregrado, Universidad de Pamplona]. Repositorio Hulgao Universidad de Pamplona. (2015). <http://repositoriodspace.unipamplona.edu.co/jspui/handle/20.500.12744/392>
- [8] Aceves-Fernandez, Marco. (2019). Hablemos Embebido: Guia para diseñar Sistemas Embebidos.
- [9] D. Schmidt, Arduino para todos. Argentina: Autores Ed., 2023.
- [10] Greengard, S. (2021). The internet of things. MIT press.
- [11] A. Maier, A. Sharp and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things," *2017 Internet Technologies and Applications (ITA)*, Wrexham, UK, 2017, pp. 143-148, doi: 10.1109/ITECHA.2017.8101926.
- [12] OpenAI, "ChatGPT," Chat.openai.com, 2025. [En línea]. Disponible en: <https://chat.openai.com/>