FECHA de PUBLICACIÓN 30/01/2024
FECHA de ENTREGA 28/02/2024

PRÁCTICA 1

PROGRAMACIÓN EN C

FECHA de PUBLICACIÓN 30/01/2024
FECHA de ENTREGA 28/02/2024

Contenidos	https://www.arduino.cc/en/Reference/HomePage https://mischianti.org/vcc-gnd-studio-yd-esp32-s3-devkitc-1-clone-high-resolution-pinout-and-specs/ https://mischianti.org/wp-content/uploads/2023/08/vcc-gnd-studio-yd-esp32-s3-devkitc-1-clone-pinout-mischianti-high-resolution-1.png
Formato de entrega	La solución se presentará en formato pdf incluyendo el texto de los ficheros.ino creados (Un fichero pdf es un contenedor digital en el que se pueden incluir otros archivos)
Lugar de entrega	Buzón de moodle.
Límite de entrega	28/2/2025 23:59.
Nombre de entrega	El nombre del fichero será ProgramacionApellido1Apellido2Apellido3.pdf, por ejemplo ProgramacionEstebanGracia.pdf
Observaciones	• Para dudas y aclaraciones sobre el enunciado, podéis utilizar el foro de la asignatura. • Debéis responder ordenadamente a todas las preguntas del enunciado, manteniendo la pregunta a resolver. • Sed precisos en las respuestas, justificando cada decisión. • Comentad el código solución de los problemas • La ejecución de la práctica es individual cualquier sospecha de copia será castigada con el suspenso de la asignatura. **Nunca he suspendido a nadie por no saber resolver alguna cuestión, pero muchas veces por copiar solo una de ellas.**

FECHA de PUBLICACIÓN 30/01/2024
FECHA de ENTREGA 28/02/2024

Enunciado de la práctica

Escribe tu nombre y NIA, **si alguno de los dígitos es 0, escribe un 10**
Nombre: Apellidos: NIA:

A	B	C	D	E	F

Todos los programas tienen que ser **validados** por el profesor, durante la defensa de un programa se podrá solicitar realizar cualquier modificación o una explicación del código.

Sección1

La estructura básica de un programa en Arduino es:

```
setup(){
}

loop(){
}
```

La parte de código de setup solo se ejecuta una vez después del reset. La parte del loop se ejecuta continuamente (bucle infinito) y solo después de haber ejecutado el setup

Ejemplo 1:

```
setup(){
    Serial.begin(115200); // Initialice serial port to 115200 bauds.
    Delay(20);
    Serial.println("Hola mundo!");
}

loop(){ //Nothing to do.
}
```

La placa YD-esp32-S3 V1.3 o V1.4 tiene dos leds integrados en los puertos 43 y 44. Además tiene un led de tres colores en el puerto 48 modelo WS2812B

Ejemplo 2: Haga parpadear uno de los leds

```
#define LED 43
void setup() {
  pinMode(LED, OUTPUT); //Configure LED as output
}
```

```
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

Ejemplo: uso del led tricolor

```
   #include "Freenove_WS2812_Lib_for_ESP32.h"

   #define LEDS_COUNT  1
   #define LEDS_PIN      48
   #define CHANNEL       0

   Freenove_ESP32_WS2812 strip = Freenove_ESP32_WS2812(LEDS_COUNT,
LEDS_PIN, CHANNEL, TYPE_GRB);

   void setup() {
     strip.begin();
     strip.setBrightness(20);
   }

   void loop() {
     for (int j = 0; j < 255; j += 2) {
       for (int i = 0; i < LEDS_COUNT; i++) {
         strip.setLedColorData(i, strip.Wheel((i * 256 / LEDS_COUNT + j) & 255));
       }
       strip.show();
       delay(50);
     }
}
```

Para programar una señal como entrada se usa

pinMode(PIN, INPUT)

Para leer el valor de esa entrada

digitalRead(PIN)

1. Si A es el primer dígito de su NIA y F el último, diseñe un programa que actúe como un indicador digital de su número de NIA haciendo parpadear el LED43 con cada uno de los valores del NIA, para contar cada número realice parpadeos cortos de 100mS y entre dígitos retrase medio segundo.

Se valorará el uso de funciones para optimizar el código.

Mejora1: no use la función delay.

Mejora2: haga que el LED44 parpadee con una frecuencia fija de 400mS.

2. Alguna de las salidas de arduino se pueden convertir en analógicas usando la técnica PWM. La función utilizada es analogWrite(pin, value) para Arduino Uno o la clase ledc para esp32. Diseñe un programa que haga que el LED43 pase de apagado a encendido de forma progresiva en un tiempo total de B segundos, se mantenga C segundos encendido, se apague de forma progresiva en F segundos y vuelva a empezar. En la salida tendrá que escribir de 0 a 255 usando la función analogWrite o ledc para modular la intensidad del led. Haga que el LED44 luzca de forma inversa al LED43

Ejemplo con ESP32-S3

```
const int ledChannel = 0;
const int ledPin = 5;
const int frequency = 5000;
const int resolution = 8;

void setup() {
  ledcSetup(ledChannel, frequency, resolution);
  ledcAttachPin(ledPin, ledChannel);
}

void loop() {
  // Incrementar el ciclo de trabajo gradualmente para aumentar la intensidad del LED
  for (int dutyCycle = 0; dutyCycle <= 255; dutyCycle++) {
    ledcWrite(ledChannel, dutyCycle);
    delay(10);
  }
```

```
  // Disminuir el ciclo de trabajo gradualmente para reducir la intensidad del LED
  for (int dutyCycle = 255; dutyCycle >= 0; dutyCycle--) {
    ledcWrite(ledChannel, dutyCycle);
    delay(10);
  }
}
```

3. La función random devuelve un valor aleatorio https://programarfacil.com/blog/random-en-arduino-como-usar-numeros-aleatorios/ , haga una llamada de random para que le devuelva un valor de 0 a 10 y si el valor devuelto es mayor que 5 incremente la intensidad del led y si es menor que 5 decremente esa intensidad. Tenga en cuenta que la intensidad aplicada al led con la función analogWrite o ledc no puede ser ni superior a 255 ni inferior a 0. Ajuste el valor en que se incrementa / decrementa de la intensidad del led y el tiempo transcurrido entre dos llamadas a random para conseguir cambios de iluminación continuos y no demasiado rápidos. Si tiene un led tricolor, aplique el algoritmo a cada una de los tres leds para obtener una lámpara relajante que vaya modulando su color e intensidad. Los leds externos deben colocarse con una resistencia en serie de 220 ohm.

4. Use el Arduino como un generador de una señal sinusoidal F*sen(w*t) donde F es el dígito F de su NIA en voltios y w = 1/C rad/s frecuencia de la señal sinusoidal. Para representar la salida utilice SerialPlotter de Arduino. https://www.luisllamas.es/arduino-serial-plotter/

Sección 2

La parte más complicada de todo programa es llegar a entender a fondo el sistema de I/O:

```
setup(){ //Only once
        Serial.begin(115200); //Initialice serial port to 115200 bauds
        Delay(20); //Give time to startup serial hardware
}

loop(){ //Every milisecond
        int i;
        char s[100];

        i=0; //Index to the very last caracter read
        while(Serial.available()){ //Are new bytes available in serial port?
                s[i++]=Serial.read(); //Read the first one
                delay(1); //Wait to the next byte. One byte is (10/115200)”
        }
        s[i]=0; //Put the end of string mark
        if(i>0){ //ToDo your wonderful code
        }
}
```

Funciones que leen cosas de un vector

strtok, atoi, atof

Recorrido de un vector

```
For(int j=0; j<limite; j++){
        Vector[j] = …
```

Salida por pantalla

Serial.print, Serial.println, Serial.printf (Solo funciona en algunos modelos de arduino)

5. Diseñe un programa que lea por teclado una secuencia de letras. Cuando encuentre una a, encienda el led44, cuando encuentre una s apague el led44, una b, encienda el led43, cuando encuentre una c apague el led43, cuando encuentre una d haga parpadear el led43 E veces, cuando encuentre una f haga parpadear el led 44 C veces.

Tenga en cuenta que el sistema operativo no envía nada al Arduino por el puerto serie hasta que no pulse la tecla Enter y que los caracteres ASCII del Enter también serán enviados al Arduino.

6 Diseñe un programa que solicita al usuario su nombre y responda saludando al usuario por su nombre

Hola ¿Cómo te llamas?

Julia

Salida: Hola Julia que tengas un gran día

7. Diseñe un programa al que introduzca una lista de números y devuelva la suma de todos ellos, la lista es de longitud variable.

Entrada: 1 34 56 90 47

Salida: 228

8. Solicite un número entero como entrada y responda indicando si el número es primo o no

Dame un número: 4

El número 4 no es primo

9. El programa solicita un número como entrada y proporciona los valores de la serie de Fibonacci inferiores al valor introducido como entrada https://es.wikipedia.org/wiki/Sucesi%C3%B3n_de_Fibonacci

10. El programa solicita un número entero como entrada y responde indicando todos los factores primos del número. Por ejemplo, si el número es 8 -> 2^3, si el número es 24 -> 2^3; 3^1

Sección 3

Un puntero es un tipo de variable especial que sirve para acceder a cualquier variable de su mismo tipo. Una variable ocupa una determinada zona de memoria, un puntero es capaz de acceder directamente a esa zona de memoria.

```
int i, *p;
p = &i;
```

Se declara p como un puntero capaz de apuntar a variables de tipo int y a continuación hacemos que p apunte a la posición de memoria donde está situado i.

El modificador * indica que la variable es un puntero

El modificador & indica que nos queremos referir a la dirección de la variable i

Cuando se quiere consultar el valor de la variable que contiene el puntero, se puede hacer de dos formas, usando el operador * o el operador []

```
int b;
b = *p + 2;
b = p[0] + 2;
```

Las ventajas de los punteros son:

- Permiten modificar el valor de las variables en el interior de funciones
- Pueden mejorar la eficiencia de ciertas operaciones
- Permiten la asignación dinámica de memoria

No debe olvidarse que usar punteros entraña ciertos riesgos y es que un puntero puede apuntar a cualquier lugar de la memoria y eso puede provocar fallos en el programa.

A un puntero le podemos sumar o restar un número y lo que hará será apuntar a las siguientes variables del mismo tipo. Por ejemplo, si *p es un puntero a entero, p = p + 1, apuntará al siguiente entero de la memoria. O sea que, si un entero ocupa 4 posiciones de memoria, la dirección de p se habrá incrementado en 4 unidades.

La definición de un vector es una definición de un puntero. Al declarar int v[10], *p; podemos hacer p = v (sin necesidad de &) porque v es un puntero al vector que apunta.

Las funciones son los elementos básicos de todo programa. La estrategia básica de escritura de un programa es dividir un problema complejo en otros más sencillos y abordar independientemente cada uno de ellos. En C cada uno de estos subproblemas se implementa como una función.

Los parámetros a una función se pasan “por valor”, esto es realmente solo se envía una copia de la variable, pero no la variable en sí misma, de forma que cualquier cambio que se produzca dentro de la función, no se refleja en el programa principal.

El paso "por referencia", o sea se puede modificar el valor de las variables en el programa llamante se consigue pasando un puntero a la variable en vez de la variable.

11. strncat es una función de librería que copia una cadena al final de otra https://linux.die.net/man/3/strcat char *strncat(char *dest, char *src, size_t n)

Escriba el código que debe contener la función strncat para que se realice la copia, tenga en cuenta que dest debe tener espacio suficiente para contener la combinación de ambas cadenas.

12. bool strinit(char *s, char *t) es una función que devuelve 1 si la cadena t representa el principio de la cadena s o 0 en caso contrario. Implemente dicha función.

13. implemente una función que se encargue de ordenar los elementos de un vector numérico. void ordena(int v[]), a la función se le pasa el vector desordenado y lo devuelve ordenado de mayor a menor. El algoritmo tradicional para la ordenación es el de la burbuja https://es.wikipedia.org/wiki/Ordenamiento_de_burbuja

14. el algoritmo de la burbuja no es muy óptimo, repita el ejercicio anterior usando el algoritmo QuickShort https://es.wikipedia.org/wiki/Quicksort

15. Torres de Hanoi. El juego, en su forma más tradicional, consiste en tres postes verticales. En uno de los postes se apila un número indeterminado de discos perforados por su centro (elaborados de madera), que determinará la complejidad de la solución. Por regla general se consideran siete discos. Los discos se apilan sobre uno de los postes en tamaño decreciente de abajo arriba. No hay dos discos iguales, y todos ellos están apilados de mayor a menor radio -desde la base del poste hacia arriba- en uno de los postes, quedando los otros dos postes vacíos. El juego consiste en pasar todos los discos desde el poste ocupado (es decir, el que posee la torre) al tercer poste pudiendo usarse el poste central como ayuda.

El programa comenzará solicitando el número de discos que se van a usar y comenzará a indicar los movimientos que se producen entre los postes. Los discos se numeran desde 1 el de radio menor a n el de radio mayor.