



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Eléctrica y Electrónica

Carrera de Electrónica y Automatización

SISTEMAS BASADOS EN MCU

Práctica 3.11

PROGRAMACIÓN DE PLATAFORMAS ARDUINO

Autor:

Iza Tipanluisa Alex Paul

Docente:

Ing. Amparo Meythaler

NRC: 4891

1) OBJETIVOS

- Identificar la forma de programar en lenguaje Arduino.
- Realizar una aplicación para un GPIO básico que funcione en una plataforma Arduino.

2) MARCO TEORICO

PLATAFORMAS O TARJETAS ARDUINO

Arduino es una plataforma electrónica abierta para la creación de prototipos y que gira en torno a un microcontrolador.

Esta plataforma posee una arquitectura hardware guiada por un programa o software que le va a permitir ejecutar programas previamente diseñados. Este recurso abierto significa que puede ser usado, distribuido, modificado, copiado, etc. gratuitamente.

La placa hardware de Arduino incorpora un microcontrolador reprogramable y una serie de pines-hembra (los cuales están unidos internamente a las patillas de E/S del microcontrolador) que permiten conectar allí de forma muy sencilla y cómoda diferentes sensores y actuadores.



El software (más en concreto, un “entorno de desarrollo”) es gratis, libre y multiplataforma (ya que funciona en Linux, MacOS y Windows) que debemos instalar en el computador y que permite escribir, verificar y guardar (“cargar”) en la memoria del microcontrolador de la placa Arduino el conjunto de instrucciones que deseamos que este empiece a ejecutar. Es decir: nos permite programarlo. El lenguaje Arduino se basa internamente en código C/C++.

3) EQUIPOS Y MATERIALES

- PC con el IDE para Arduino
- PC con el paquete PROTEUS.

4) ACTIVIDADES

1) Trabajo Preparatorio:

- a) Realizar el diagrama de flujo, la codificación e implementación correspondiente, de un programa que presente un contador descendente, reciclable de número MOD escogido por Ud., el mismo que se visualizará en dos displays cátodo común.

b) Verifique el funcionamiento del MPLAB IDE y del PROTEUS.

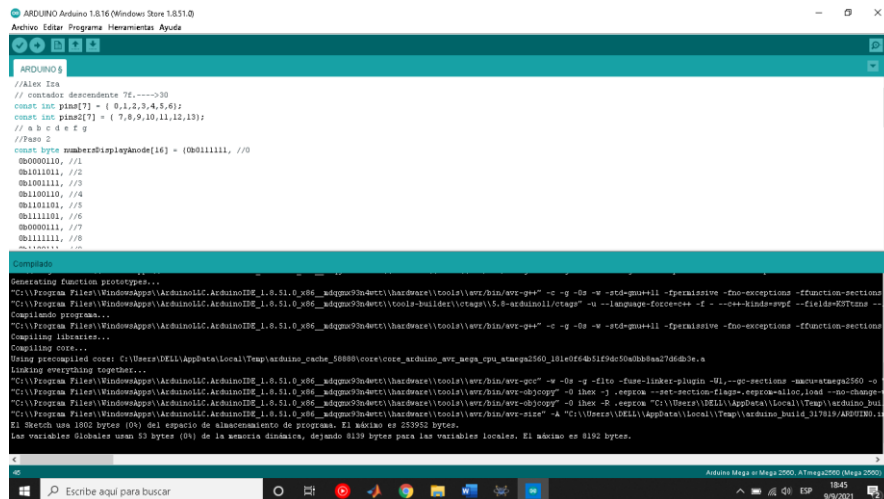


Ilustración 1, Compilación del programa

5) RESULTADOS

- **Explique los errores cometidos en el ejercicio realizado (si los tuvo) y la forma de corregirlos.**

No tuve complicaciones para realizarlo

- **¿Qué extensión tienen los archivos realizados en un IDE para Arduino?**

Los archivos tienen una extensión “.ino”

6) DISEÑO

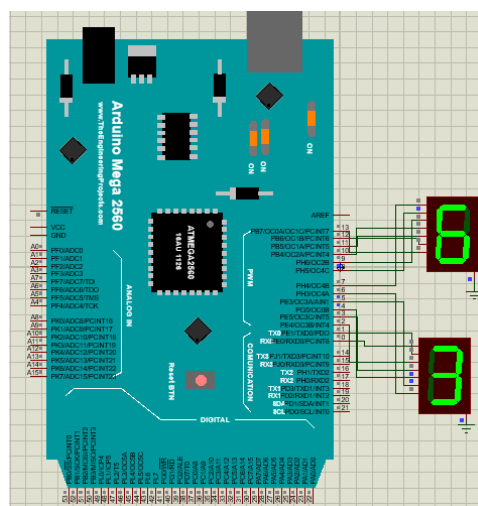


Ilustración 2, Visualización contador dato 63

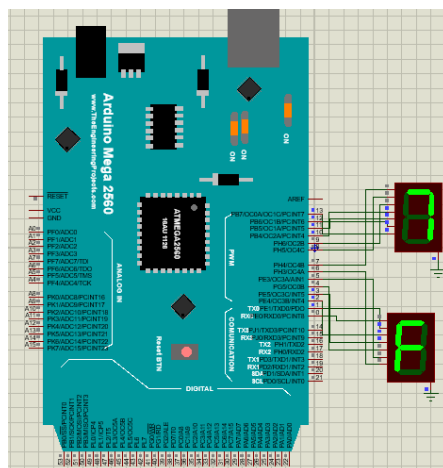


Ilustración 3, Visualización del contador dato 7F

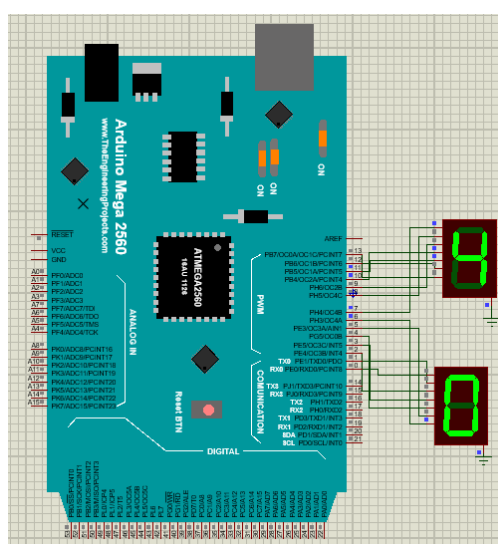


Ilustración 4, Visualización del contador dato 40

7) DIAGRAMA DE FLUJO

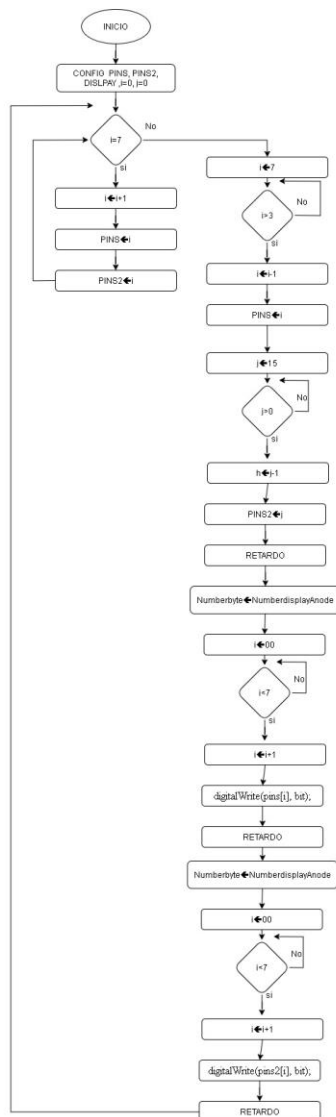
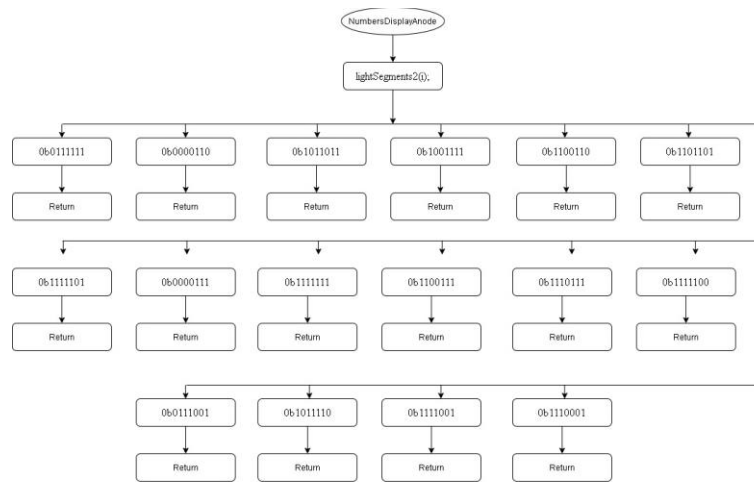


Diagrama 1

8) PROGRAMA

```
//Alex Iza

// contador descendente 7f.---->40

const int pins[7] = { 0,1,2,3,4,5,6};

const int pins2[7] = { 7,8,9,10,11,12,13};

// a b c d e f g

//Paso 2

const byte numbersDisplayCatode[16] = {0b01111111, //0
0b00001110, //1
0b1011011, //2
0b1001111, //3
0b1100110, //4
0b1101101, //5
0b1111101, //6
0b0000111, //7
0b1111111, //8
0b1100111, //9
0b1110111, //A
0b1111100, //B
0b0111001, //C
0b1011110, //D
0b1111001, //E
0b1110001}; //F

void setup() {

    for(int i = 0; i < 7; i++) {

        pinMode(pins[i], OUTPUT);
```

```
pinMode(pins2[i], OUTPUT);  
}
```

```
lightSegments(0);  
}
```

```
void loop() {
```

```
for(int i = 7; i >3;-- i) {
```

```
lightSegments2(i);
```

```
for(int j=15;j>=0; --j){
```

```
lightSegments(j);
```

```
delay(200);
```

```
}
```

```
}
```

```
}
```

```
void lightSegments(int number) {
```

```
byte numberBit = numbersDisplayAnode[number];
```

```
for (int i = 0; i < 7; i++) {
```

```
int bit = bitRead(numberBit, i);
```

```
digitalWrite(pins[i], bit);
```

```
}
```

```
}
```

```
void lightSegments2(int number) {
```

```
byte numberBit = numbersDisplayAnode[number];
```

```
for (int i = 0; i < 7; i++) {
```

```
int bit = bitRead(numberBit, i);
```

```
digitalWrite(pins2[i], bit);  
  
}  
  
}
```

9) CONCLUSIONES

- Se ha concluido que el código para la visualización de los displays son los mismos en cada uno de las plataformas es decir que se puede reutilizar y aplicarlas sin importar si la programación sea en ensamblador, C o Arduino.
- Se ha concluido que cada una de las plataformas usadas para la programación tienen un diferente código dependiendo de su lenguaje de compilación.
- Se ha concluido que los comandos cambian en cada uno de los lenguajes de programación y cada uno de ellos tienen una diferente aplicación.

10) RECOMENDACIONES

- Se recomienda revisar los comandos de programación dependiendo del lenguaje con el que trabaje la plataforma.
- Se recomienda reutilizar los códigos de programación para que la programación sea más fácil.
- Se recomienda tomar en cuenta el tipo de display con el que se va a trabajar, ya que de este depende que el código sea el mismo o cambie.

11) BIBLIOGRAFIA

Meythaler, A. (2021). Sistemas basados en MCU. Ecuador: UFA ESPE