



**TECNOLÓGICO
NACIONAL DE MÉXICO**



Instituto Tecnológico de Chihuahua

Arquitectura de programación para hardware

Laboratorio 3.

Nombre:

Jorge Arnoldo Hernández Hernández No. Control 18060726

Docente:

Alfredo Chacón Aldama

Chihuahua, Chih., 01/04/2022

Objetivo: Que el alumno realice un programa el cual simule un el uso de los ADC, aplicando el paradigma de programación orientada a objetos.

Marco teórico:

Herencia:

La herencia es cuando se crea una o mas clases nuevas a partir de una ya existente, heredando sus atributos y métodos. A la clase de la cual se derivan las subclases (clases hijas) se denominan superclase, clase padre, clase raíz o clase primaria.

Ejemplo: Al momento de crear una clase se piensa en un grupo de objetos con características similares, en este caso si se crea una clase con el nombre de “ConsolaDeVideosJuegos” esta es nuestra clase raíz o padre y de ahí de derivarían nuestras subclases, las cuales podrían ser PS5, XBOX, NintendoSwitch. Estas clases hijas heredarían los atributos y métodos que nuestra clase “ConsolaDeVideosJuegos”. Algunos de los atributos podrían ser “ColorDelaConsola”, “PuertosParaControles”, “Modelo” y algunas de la funciones podrían ser “Jugar()”, “LeerJuego()”, “GuardarDatos()”, etc. Entonces todos esos atributos y métodos son los que heredarían nuestras funciones hijas y esto es lo que se considera herencia en POO.

Polimorfismo:

Es la habilidad de un objeto de realizar una acción de diferentes maneras utilizando métodos iguales que se implementan de forma diferente en varias clases.

Ejemplo: Haciendo referencia al ejemplo de herencia supongamos que se crea la clase “ConsolaDeVideojuegos” y esta tiene el método o función “LeerJuego()”, y cada una de las subclases lo van a heredar, entonces las subclases “PS5”, “XBOX”, “NintendoSwitch” van a tener la función “LeerJuego” pero eso no quiere decir que todas las subclases la vayan a realizar de la misma manera, puede ser que para “PS5” se introduzca un CD, para “XBOX” el juego se lea desde descargas en la biblioteca de video juegos descargados y para “NintendoSwitch” se introduzca un caset. Entonces como se puede apreciar todas las subclases van a realizar la función “LeerJuego()”, pero cada una lo va a hacer de diferente forma, esto es polimorfismo, la capacidad de hacer algo de diferentes maneras.

Metodología: En estas practica se realizará un programa por medio de la programación orientada a objetos la cual nos permita realizar una simulación de un ADC con las diferentes resoluciones de 8, 10 y 12 bits, además de pedir la frecuencia y poder elegir en que puerto se trabajará.

También se creará un repositorio por medio de Git.

Materiales:

- Software Dev-c++
- Git

Desarrollo

Para la parte tres de esta practica lo primer que hice fue definir las librerías que se van a usar en el código, además de declarar un define para el máximo numero de canales a usar

```
1  #include <iostream>
2  #include <stdlib.h>
3  #define MAX 33
4
```

Después de esto lo que hice fue crear una clase con el nombre de “ADC” la cual tiene los atributos y métodos que voy a utilizar.

```
7  class ADC{
8
9      private:
10         int resolucion;
11         int resl;
12         int canales[MAX];
13         int num;
14         int frecuencia;
15         float conversion;
16         float voltaje [MAX];
17
18     public:
19         void capturar();
20         void mostrar();
21         float calculo(float x);
22
23 };
24
```

Seguido de esto cree lo que va a hacer cada función, en el caso de “capturar”, lo que hace como su nombre lo indica es guardar los datos que se están introduciendo, como, el numero de puertos a usar, el voltaje y la frecuencia.

```

25
26 void ADC::capturar(){
27
28     int i, anl;
29
30     cout<<"Ingresa la resolucion (8, 12, 14 bits)"<<endl;
31     cin >> resolucion;
32     if(resolucion==8) resl=255;
33     if(resolucion==10) resl=1023;
34     if(resolucion==12) resl=4095;
35
36     for (i=0;i<MAX;i++){ //Para inicializar los canales en 0 para evitar basura
37         canales[i]=0;
38     }
39
40     cout<<"Cuantos canales desea usar?";
41     cin>> num;
42
43     cout<<"Ingresa la frecuencia:";
44     cin>> frecuencia;
45
46     for(i=0; i<num; i++){
47         cout<<"Seleccione el canal a usar (max 32): AN";
48         cin>>anl;
49         canales[anl]=1; // Es para habilitar el canal a usar
50     }
51
52     for (i=0;i<MAX;i++){
53         if (canales[i]==1){
54             cout<<"Canal AN"<<i<<endl;
55         }
56     }
57
58     for(i=0; i<MAX; i++){

```

La siguiente función tiene como objetivo hacer la conversión de voltaje a la resolución que se le haya tecleado.

```

83 float ADC::calculo(float x){
84     x = conversion/3.3*resl;
85     return x;
86 }

```

Y la última función lo que hace es imprimir los datos en pantalla

```

68
69 void ADC::mostrar(){
70     int i;
71
72     cout<<"\nMostrando datos";
73     cout<<"\nResolucion "<<resolucion<<" bits "<<endl;
74     cout<<"\nFrecuencia "<<frecuencia<<"Khz "<<endl;
75
76     for(i=0; i<MAX; i++){
77         if(canales[i]==1){
78             cout<<"AN"<<i<<" conversion "<< voltaje[i]<<endl;
79         }
80     }
81 }

```

Ya por ultimo tenemos el main la cual tiene una variable con el nombre de "x" con la cual se hace el conversión por medio de la función "calculo", seguido de esto se crea un objeto del tipo de la clase ADC llamado "Canal" el cual se usa para llamar a las funciones "capturar" y "mostrar datos"

```

83 float ADC::calculo(float x){
84     x = conversion/3.3*resl;
85     return x;
86 }
87
88
89 int main(){
90
91     float x;
92
93     ADC Canal;
94
95     Canal.capturar();
96     Canal.mostrar();
97
98     system("pause");
99
100    return 0;
101
102 }
103

```

Resultados:

```

C:\Users\jorge\Desktop\Arqui\Laboratorio3_ADC\Laboratorio3_ADC.exe
Ingresa la resolucion (8, 12, 14 bits)
8
Cuantos canales desea usar?4
Ingresa la frecuencia:1
Seleccione el canal a usar (max 32): AN1
Seleccione el canal a usar (max 32): AN3
Seleccione el canal a usar (max 32): AN6
Seleccione el canal a usar (max 32): AN9
Canal AN1
Canal AN3
Canal AN6
Canal AN9
Ingresa el voltaje (max 3.3v) del AN1:2.3
Ingresa el voltaje (max 3.3v) del AN3:2.4
Ingresa el voltaje (max 3.3v) del AN6:1.2
Ingresa el voltaje (max 3.3v) del AN9:1.9

Mostrando datos
Resolucion 8 bits

Frecuencia 1Khz
AN1 conversion 177.727
AN3 conversion 185.455
AN6 conversion 92.7273
AN9 conversion 146.818
Presione una tecla para continuar . . .

```

Conclusión: En esta practica se reforzo mas los visto en clase sobre como utilizar C++ además de poder implementar la programación orientada a objetos. Así como repasar los conceptos de herencia y polimorfismo.

Anexos:

```
/*//*****  
// FileName:    Laboratorio3.c  
// Program version: Dev-C++ 5.11  
// Company:     TECNM - ITCh  
// Description:  Simulador de ADC  
// Authors:     Jorge Arnoldo Hernández Hernández  
// Updated:     01/04/2022  
//*****/
```

```
#include <iostream>
```

```
#include <stdlib.h>
```

```
#define MAX 33
```

```
using namespace std;
```

```
class ADC{
```

```
    private:
```

```
        int resolucion;
```

```
        int res;
```

```
        int canales[MAX];
```

```
        int num;
```

```
        int frecuencia;
```

```
        float conversion;
```

```
        float voltaje [MAX];
```

```
    public:
```

```
        void capturar();
```

```

        void mostrar();

        float calculo(float x);

};

void ADC::capturar(){

    int i, anl;

    cout<<"Ingresa la resolucion (8, 12, 14 bits)"<<endl;
    cin >> resolucion;
        if(resolucion==8) resl=255;
        if(resolucion==10) resl=1023;
        if(resolucion==12) resl=4095;

    for (i=0;i<MAX;i++){ //Para inicializar los canales en 0 para evitar basura
        canales[i]=0;
    }

    cout<<"Cuantos canales desea usar?";
    cin>> num;

    cout<<"Ingrese la frecuencia:";
    cin>> frecuencia;

    for(i=0; i<num; i++){
        cout<<"Seleccione el canal a usar (max 32): AN";
        cin>>anl;
    }
}

```

```

        canales[anl]=1; // Es para habilitar el canal a usar
    }

    for (i=0;i<MAX;i++){
        if (canales[i]==1){
            cout<<"Canal AN"<<i<<endl;
        }
    }

    for(i=0; i<MAX; i++){
        if(canales[i] == 1){
            cout<<"Ingresa el voltaje (max 3.3v) del AN"<<i<<":";
            cin>> conversion;
            voltaje[i]= calculo(conversion);
        }
    }
}

```

```

void ADC::mostrar(){
    int i;

    cout<<"\nMostrando datos";
    cout<<"\nResolucion "<<resolucion<<" bits "<<endl;
    cout<<"\nFrecuencia "<<frecuencia<<" Khz "<<endl;

    for(i=0; i<MAX; i++){
        if(canales[i]==1){

```



```
        cout<<"AN"<<i<<" conversion "<< voltaje[i]<<endl;
    }
}
}
```

```
float ADC::calculo(float x){
    x = conversion/3.3*resl;
    return x;
}
```

```
int main(){

    float x;

    ADC Canal;

    Canal.capturar();
    Canal.mostrar();

    system("pause");

    return 0;

}
```