UNIVERSIDAD DEL SURESTE
CAMPUS S.C.L.C.C. CHIAPAS

Actividad

Codificación

Compuerta Lógica AND Y NOT

Reporte

Diseño Lógico

Ing. Sistemas computacionales.

Ian Jair Gómez Méndez.

Ing. Emmanuel Fabio Santiago Aguilar

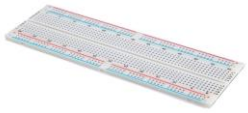
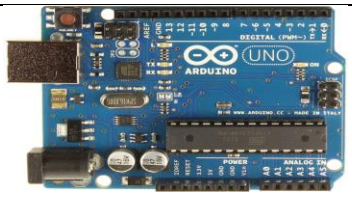
6º cuatrimestre.

San Cristóbal de las casas, Chiapas; a 04 de julio de 2020.

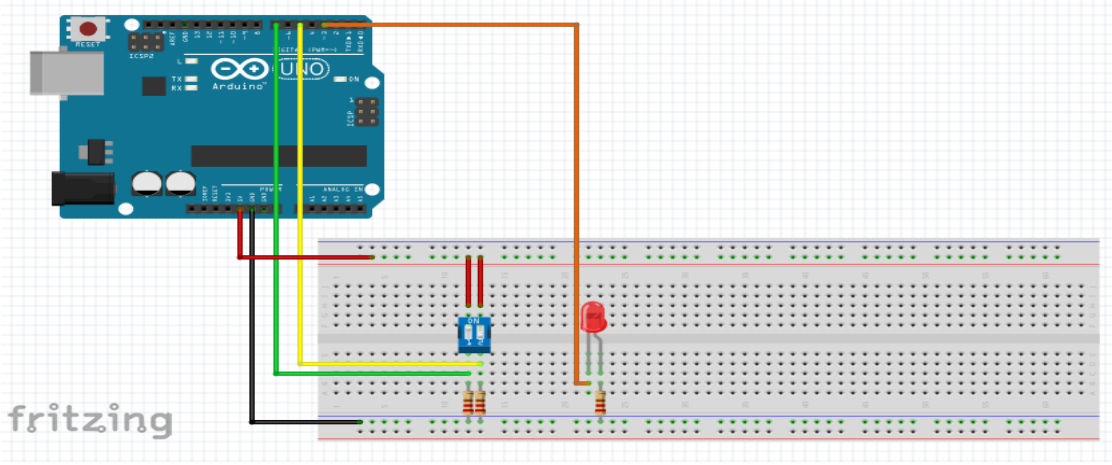
INDICE

Contenido	Pag.
Descripción de materiales	3
Esquema de fritzing	3
Esquema inicial de conexión	4
Código de solución	4
Código de compuerta lógica AND	4
Compuerta lógica OR	5
Resultados	6
Reflexiones	6

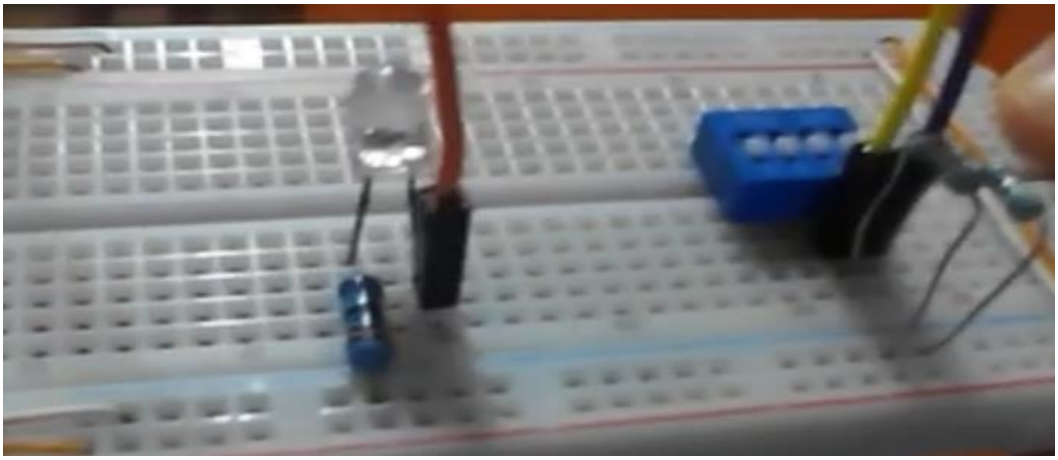
Descripción de materiales

Cantidad	Nombre del material	Imagen
2	Resistencias de 1 kilo	
1	Resistencia de 220 ohms	
1	LED	
1	Dip switch	
1	Protoboard	
1	Placa de Arduino uno	
varios	Cables de conexión	

Esquema de fritzing



Esquema inicial de conexión



Código de solución

Código de compuerta lógica AND

1. Se declararon las entradas y salidas en este caso los pines como entradas y la salida el LED.

```
//declara las entradas y ssalidas a manejar en el circuito
//entradas
const int pe1=7;
const int pe2=5;
//Salida
const int led=0;
```

2. En este paso se declaró los valores de las entradas para recibir de las variables

```
//Declarar las variables para recibir los valores de las entradas y salidas
int ve1=0;
int ve2=0;
```

3. Aquí se definieron los componentes como entradas y salidas, los primeros como entradas y el ultimo como salida, esto en la parte de void setup.

```
void setup() {
  // Definir lod pines de los componentes como entradas o ssalidas
  pinMode(pe1, INPUT);
  pinMode(pe2, INPUT);
  pinMode(led, OUTPUT);

}
```

4. Aquí se declaró que el código pueda leer y almacenar, los valores de las entradas del dip switch. Esto con el valor entrada 1 y 2 y el pin entrada 1y 2 igualitariamente.

```
void loop() {
  // Leer y almacenar los valores como entradas del dip switch
  ve1 = digitalRead(pe1);
  ve2 = digitalRead(pe2);
```

5. Por último, se definió la función AND por medio de las condiciones para la codificación que es if/else, en este caso tiene que sumar los dos valores, para que el led encienda; y si un valor esta apagado el led no encenderá" solo los dos valores encendidos encenderán el led."

```
  //definir la funcion AND
  if(ve1 and ve2){
    digitalWrite (led, HIGH);

  }else{
    digitalWrite (led, LOW);
  }

}
```

Compuerta lógica OR

En el caso de esta compuerta solo cambio en una sola palabra de todo el código donde se encuentra la funcionalidad de OR.

```
  //definir la funcion OR
  if(ve1 or ve2){
    digitalWrite (led, HIGH);

  }else{
    digitalWrite (led, LOW);
  }
```

Resultados

Esta práctica se ve difícil, pro no lo es más fácil de lo que uno cree en los resultados de la compuerta lógica AND es que las entradas tienen que estar activadas, para la salida que es el LED encienda, porque si una de las entradas está apagada esto no funcionara, puesto que en el código está diciendo que debe de leer las dos entradas, o más; ya que AND en español es y.

En el caso de OR hay tres posibles encendidos ya que OR significa o y este podrá encender con una entrada encendida y la otra apagada o bien las dos encendidas ya que está diciendo que ya sea una de las este encendida el funcionara, ya que está diciendo puedo trabajar, pero lo otro no lo necesito tanto, como por ejemplo un celular no puede funcionar sin un sistema operativo, ya sea Android, IOS, puesto que esto vital para su funcionamiento, esto sería para AND, OR seria de que el mismo celular puede funcionar con o sin juegos, cámara, ya que esto no es fundamental para su funcionamiento.

Reflexiones

Pues las compuertas lógicas son muy importantes en cuestión de los sistemas computacionales y de más ramas ya que están nos ayudan a facilitar protocolo, como de encendido y apagado de cualquier dispositivo, o cualquier situación, puesto que se programó antes en la cual el proceso va ir detectando si tiene y puede seguir con la función o bien si puede funcionar sin una parte de algún elemento, y si no abortar la función y seguir con otro proceso que el usuario le indique.