

Bitacora proyecto individual

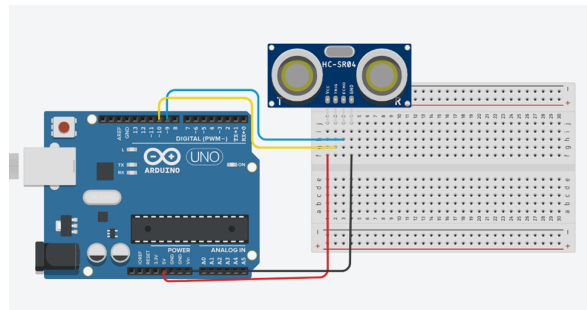
Angelo Ceciliano Ortega (2021035484)

4/4/2024

1. 28/3/24

Primeramente, se realizó un análisis de los solicitado en el proyecto en donde se fue analizando modulo por modulo.

Primeramente, se analizó el módulo del sensor donde se optó por la opción de un sensor ultrasónico HC-SR04, se simulo en Tinkercad primeramente



Físicamente se realizó, pero al ser un módulo simple no se documentó. Básicamente el sensor tiene la tensión, tierra, el trigger y el echo, el trigger manda una señal y al rebotar este es recibida por el echo, en el código de Arduino lo que se hace es buscar la distancia a través de una formula en donde se utiliza tanto el tiempo como la velocidad de duración.

La idea para generar el código grey es delimitar ese rango de 10cm en 10cm.

Numero decimal	Código grey	Rango del analógico
0	000	[0-10]cm
1	001	[10-20]cm
2	011	[20-30]cm
3	010	[30-40]cm
4	110	[40-50]cm
5	111	[50-60]cm
6	101	[60-70]cm
7	100	[70-80]cm

De 80 cm en adelante lo que se busca nada mas es que no se encienda los leds del siguiente modulo.

Se inicio con el siguiente modulo que sería el circuito combinatorio, es decir decodificador de grey a exceso 3. Para es lograr hacer el decodificador se busca 3 expresiones booleanas para implementarlas en circuito, para esto se utilizó los mapas de Karnaugh

Primeramente, se utiliza esta tabla para guiarnos en la creación de los mapas K para cada digito (bit de salida).

Primeramente, se utiliza esta tabla para guiarnos en la creación de los mapas K para cada digito (bit de salida).

Numero decimal	Código grey	Exceso 3	D3	D2	D1
0	000	011	0	1	1
1	001	100	1	0	0
2	011	101	1	0	1
3	010	110	1	1	0
4	110	111	1	1	1
5	111	000	0	0	0
6	101	001	0	0	1
7	100	010	0	1	0

Para el digito1(D1) se hizo el siguiente mapa

AB/C	00	01	11	10
0	1	0	1	0
1	0	1	0	1

Para el digito 2 (D2):

AB/C	00	01	11	10
0	1	1	1	1
1	0	0	0	0

Para el dígito 3 (D3)

AB/C	00	01	11	10
0	0	1	1	0
1	1	1	0	0

Utilizando minitérminos resultaron las siguientes expresiones booleanas.

$D_1 = -(ABC) + -ABC + AB - C + A - BC$ Simplificando y dándole un sentido más físico se obtiene:

$= -A(-B - C + BC) + A(B - C + -BC)$ (*Distributividad*)

$= -A(-(B \oplus C)) + A(B \oplus C)$

$D_2 = -C$

$D_3 = -AC + -AB + B - C$

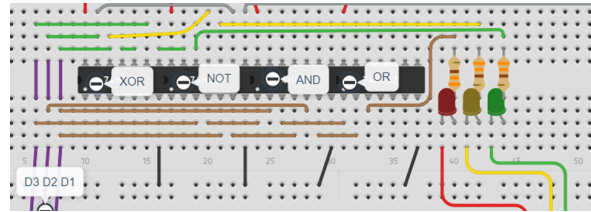
Simplificando un poco se obtiene:

$D_3 = -AC + B - C$ (*Consenso*)

2. 29/3/2024

ETeniendo las expresiones se podrá iniciar en el modelado del circuito combinatorio (decodificador).

Para modelar el circuito se utilizó Tinkercad.



Se establecieron los cables morados como las salidas del Arduino en los pines 13, 12 y 11 con los dígitos D3, D2 y D1 respectivamente.

Siguiendo las expresiones para cada bit se logró decodificar grey a exceso 3, esto se pudo observar al usar los leds.

Esta salida de exceso 3 será devuelta al Arduino para convertirla en BCD.

3. 30/3/2024

Se realiza el módulo del desacople.

Para este módulo se debía elegir ciertos rangos para enciendan el actuador. Para este caso se eligió un motor TT como desacople.

Los rangos del fotosensor que se eligieron para que se encienda el motor son los siguiente:

Numero decimal	Código grey	Rango encendido
0	000	1
1	001	1
2	011	0
3	010	0
4	110	0
5	111	0
6	101	1
7	100	1

Para que se encienda por los rangos deseados se debe pasar las salidas por un circuito combinatorio en donde se debe conformar por compuertas lógicas, es decir integrados. Para saber que circuito se ocupa debe encontrar una expresión lógica, eso se hizo mediante mapas K.

AB/C	00	01	11	10
0	1	0	0	1
1	1	0	0	1

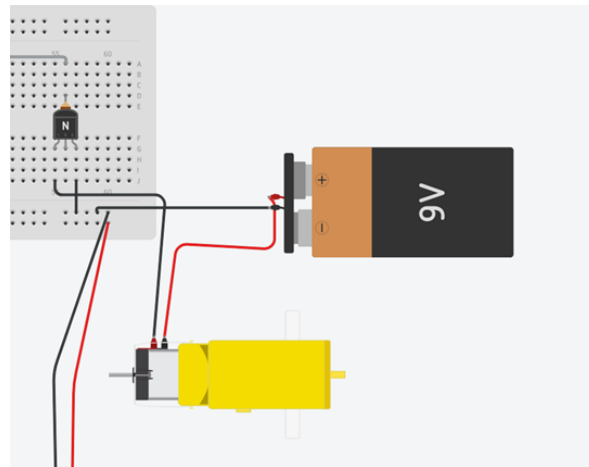
$$-A - B + A - B$$

Simplificando

$$-B(-A + A)(Distribuidad)$$

$$= -B(Cobertura)$$

Simulado se vería así



En donde mediante un transistor BJT se activa el circuito del motor y la batería, donde esta salida del Arduino pasara por una compuerta negativa e ira a la base del transistor, para encender o apagar dependiendo del rango que agarre el fotosensor.

4. 31/3/2024

Se realizó el visualizador LCD, este se hizo mediante un 7-segmentos, para esto se usaron 7 pines del Arduino.

La idea es mostrar el código BCD en Hexadecimal del circuito combinatorio, para eso se agarró tres líneas del resultado del combinatorio, es decir en exceso 3 y se decodificaron en el Arduino a BCD. Para esta decodificación también se utilizó los mapas K para la obtención de las operaciones booleanas que nos ayudaran a llegar a la combinación deseada.

Para la creación de los mapas se usó como ayuda esta tabla:

Numero decimal	Exceso 3	D3	D2	D1
0	011	0	0	0
1	100	0	0	1
2	101	0	1	0
3	110	0	1	1
4	111	1	0	0
5	000	1	0	1
6	001	1	1	0
7	010	1	1	1

Respectivamente del bit se hicieron los mapas.

Para el D1:

AB/C	00	01	11	10
0	1	1	1	1
1	0	0	0	0

Para el D2:

AB/C	00	01	11	10
0	0	1	1	0
1	1	0	0	1

Para el D3:

AB/C	00	01	11	10
0	1	1	0	0
1	1	0	1	0

Teniendo como resultado la expresiones:

$$D_1 = \neg C$$

$$D_2 = \neg A \neg B C + B \neg C + A \neg B C$$

$$= \neg B C (\neg A + A) + B \neg C \text{ (Distribucion)}$$

$$= \neg B C + B \neg C \text{ (Complemento)}$$

$$D_2 = B \oplus C$$

$$D_3 = \neg A \neg C + \neg A \neg B + A B C$$

En código de C++ de Arduino se implementaron así:

Para el primer bit:

```
int digit1_tobcd(byte C){
    if(!C){
        return 1;
    }else{
        return 0;
    }
}
```

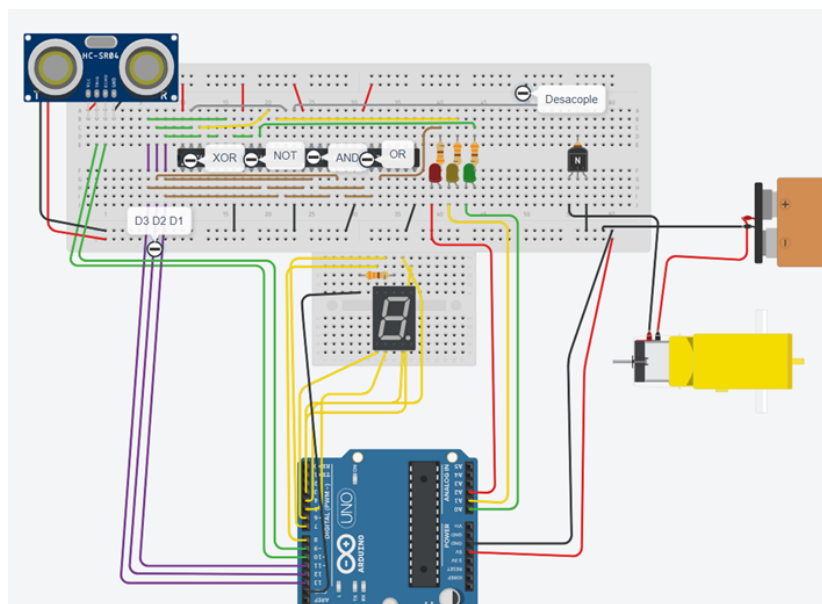
Segundo Bit:

```
int digit2_tobcd(byte B, byte C){
    if(B ^ C){
        return 1;
    }else{
        return 0;
    }
}
```

Para el ultimo bit:

```
int digit3_tobcd(byte A, byte B, byte C) {  
  if (!A && !C) {  
    return 1;  
  } else if (!A && !B) {  
    return 1;  
  } else if (A && B && C) {  
    return 1;  
  } else {  
    return 0;  
  }  
}
```

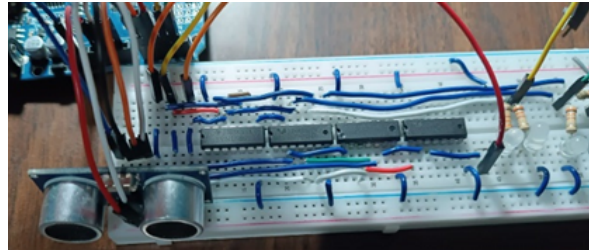
En general el circuito implementado completamente en simulación quedaría así:



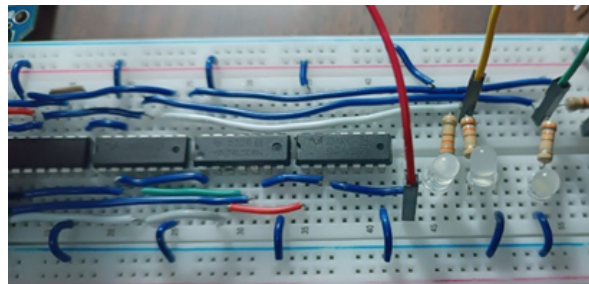
5. 1/4/2024

ESe implementa en físico.

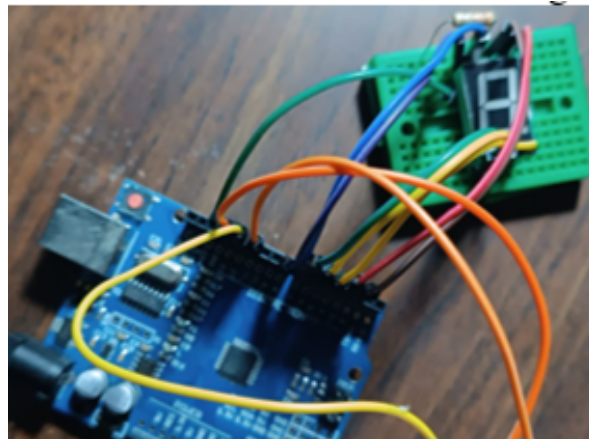
Acá se ve el módulo del sensor, y el circuito combinacional (Decodificador)



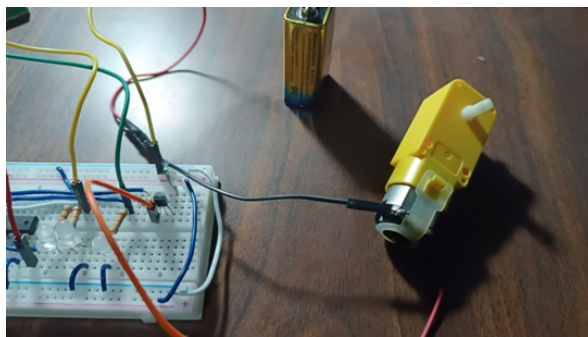
Acá se observa lo que es el resultado del decodificador que se representa con 3 leds, y se agarran 3 líneas que irán al input del Arduino.



Acá se muestra el visualizador 7-segmentos para el BCD del exceso 3.

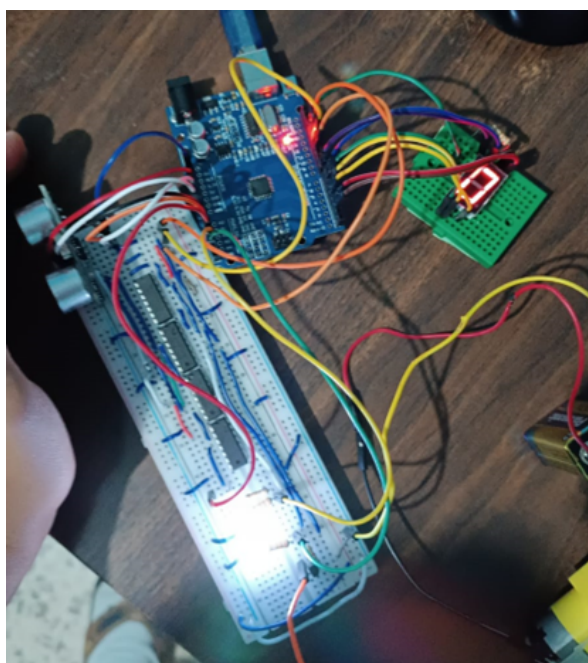


Este es el desacople del combinatorio, donde se puede observar el transistor 2N2222a como interruptor, un motor TT y su alimentación batería de 9 volts.



Se muestra el sistema en general funcionando, colocando la mano a menos de 10 centímetros del sensor.

En el visualizador de 3 leds no se distingue, pero se muestra un “0 1 1”, (1er led apagado, 2do encendido, 3ro encendido), también se ve claramente que el visualizador 7-segmentos es 0. En la foto no se nota, pero el motor está girando en ese rango.



En esta otra imagen el sensor esta agarrando otro rango, que estaría entre 40-30cm por ende en el visualizador de 3 leds habría “1 0 1”, y también en el 7-segmentos un 3.

