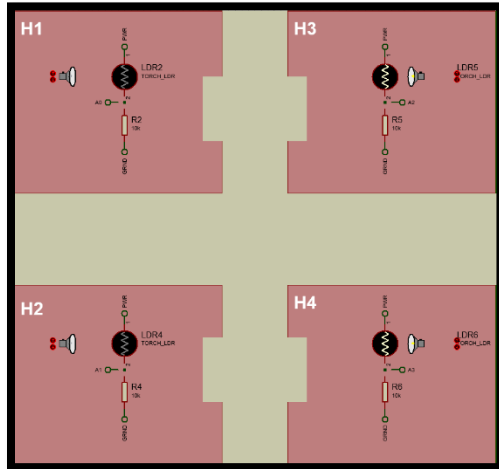


## Control de Luces

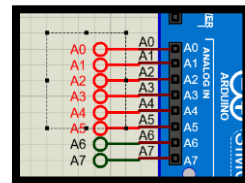
### Componentes utilizados:

Para el control de luces se utilizaron los siguientes materiales en proteus:



- 4 Fotorresistencias LDR
- 4 Resistencias 10 k

Se utilizaron las entradas Analógicas A0, A1,A2,A3.



### Lógica utilizada:

Se utilizaron las siguientes variables.

CCLight: Variable booleana para activar motor si la contraseña es correcta.

Hi<sub>n</sub> : Estado inicial de Luces

Ha<sub>n</sub> : Estado actual de Luces

```
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);  
//Variables Control de Luces  
bool CCLight=false;  
  
//Habitaciones: Estado Inicial - Estados Actual  
int Hi1=0; int Ha1=0;  
int Hi2=0; int Ha2=0;  
int Hi3=0; int Ha3=0;  
int Hi4=0; int Ha4=0;
```

El propósito de estas variables es para comparar. En caso que haya algún cambio en el estado en el que se encontraban las luces al momento de hacer el chéquel inicial del

control de luces y el momento actual. En caso que exista un cambio, se repetirá el chequeo. El control de luces se manejó con la siguiente función "CheckControlLight".

```
void CheckControlLight(){
    Hi1=Ha1; Ha1=analogRead(A0);
    Hi2=Ha2; Ha2=analogRead(A1);
    Hi3=Ha3; Ha3=analogRead(A2);
    Hi4=Ha4; Ha4=analogRead(A3);
    if((Hi1+Hi2+Hi3+Hi4)==0 && (Ha1+Ha2+Ha3+Ha4)==0){ //Estado Inicial
        RecorrerHabitaciones();
    }else if(((Hi1-Ha1)+(Hi2-Ha2)+(Hi3-Ha3)+(Hi4-Ha4))==0){ //Ya hizo un recorrido y nada ha cambiado
        lcd.setCursor(1,0);
        lcd.print("Control Luces");
        lcd.setCursor(1,1);
        lcd.print("Finalizado   ");
        CCLight=true;
        delay(800);
    }else{ //Ya hizo un recorrido y hay cambios
        RecorrerHabitaciones();
        CCLight=false;
    }
}
```

Se utilizaron 3 condiciones:

1. Estado inicial: Todas las variables de comparación están en cero. Eso significa que si estas son sumadas su resultado será cero.
2. No hay Cambios: No hubo ningún cambio en el manejo de luces, el sistema para esperando nuevos cambios u otra instrucción.
3. Cambio de estado: Hubo un cambio de estado en alguna luz de las habitaciones. Eso significa que si se comparan las variables del estado anterior con las actuales, por medio de operaciones aritméticas, su estado será diferente de cero y se necesitará actualizar el control de luces.

La función "CheckControlLight", hizo uso de la función "RecorrerHabitaciones", para actualizar los estados de las luces en la pantalla lcd.

```
}
int ldr=0;
void RecorrerHabitaciones(){
    for (int i=1; i<=4;i++){
        if (i==1){ //Habitacion 1
            ldr=analogRead(A0);
            Ha1=ldr;
            lcd.setCursor(1,0);
            lcd.print("Habitacion 1:");
            if(ldr<=105){
                lcd.setCursor(1,1);
                lcd.print("Luz Apagada");
                delay(3000);
            }else{
                lcd.setCursor(1,1);
                lcd.print("Luz Encendida");
                delay(3000);
            }
        }else if (i==2){
            ldr=analogRead(A1);
            Ha2=ldr;
            lcd.setCursor(1,0);
            lcd.print("Habitacion 2:");
            if(ldr<=105){
                lcd.setCursor(1,1);
                lcd.print("Luz Apagada");
                delay(3000);
            }else{
                lcd.setCursor(1,1);
                lcd.print("Luz Encendida");
                delay(3000);
            }
        }
    }
}
```

Se utilizó la condición for para recorrer las habitaciones y se utilizó la función AnalogRead, para obtener el valor de las luces según el pin de entrada análogo que se le especificó.