



Subdirección Académica
Departamento de Sistemas y Computación
Ingeniería en Sistemas Computacionales
Semestre: Enero - Junio 2017
Materia: Sistemas Programables (3SC8A)

Nombre del tema:
Practica 12

Nombre del alumno:
Salcedo Morales José Manuel (13211419)

Nombre del catedrático:
Ingeniero Luís Alberto Mitre Padilla

Índice

1	Introducción	3
2	Componentes utilizados	3
3	Marco Teórico	4
4	Desarrollo	5
4.1	Imagenes	6
4.2	Diseño	10
4.3	Codigo	14
5	Conclusión	16

1 Introducción

En esta practica se pretende mostrar con LED's la potencia que da una fotoresistencia cuando no hay luz. Todo esto siendo controlado por un arduino.

2 Componentes utilizados

- Arduino
- Cables Jumper
- Fuente de alimentacion para arduino
- Resistencias
 - 2KΩ
 - 5KΩ
- Optointerruptor Itr8102
- Fotoresistencia

3 Marco Teórico

- Arduino: Arduino se refiere a una plataforma o placa de electrónica de código abierto y al software utilizado para programarlo. Arduino está diseñado para hacer la electrónica más accesible a los artistas, diseñadores, aficionados y a cualquiera interesado en la creación de objetos interactivos o entornos. Un tablero de Arduino se puede comprar pre-ensamblado o, porque el diseño de hardware es de código abierto, construido a mano. De cualquier manera, los usuarios pueden adaptar las tablas a sus necesidades, así como actualizar y distribuir sus propias versiones.
- Optointerruptor: Este sensor se compone de un emisor infrarrojo en un vertical y de un detector blindado del infrarrojo en el otro. Al emitir un haz de luz infrarroja desde una posición vertical a la otra, el sensor puede detectar cuando un objeto pasa entre los montantes, rompiendo la viga. Se utiliza para muchas aplicaciones, incluyendo interruptores ópticos de límite, dispensación de pellets, detección general de objetos, etc...
- Fotoresistencia: Las resistencias fotoeléctricas, también conocidas como resistencias dependientes de la luz (LDR), son dispositivos sensibles a la luz que se utilizan con mayor frecuencia para indicar la presencia o ausencia de luz, o para medir la intensidad de la luz. En la oscuridad, su resistencia es muy alta, a veces hasta $1M\Omega$, pero cuando el sensor LDR está expuesto a la luz, la resistencia disminuye drásticamente, incluso a unos pocos ohmios , dependiendo de la intensidad de la luz. LDRs tienen una sensibilidad que varía con la longitud de onda de la luz aplicada y son dispositivos no lineales.

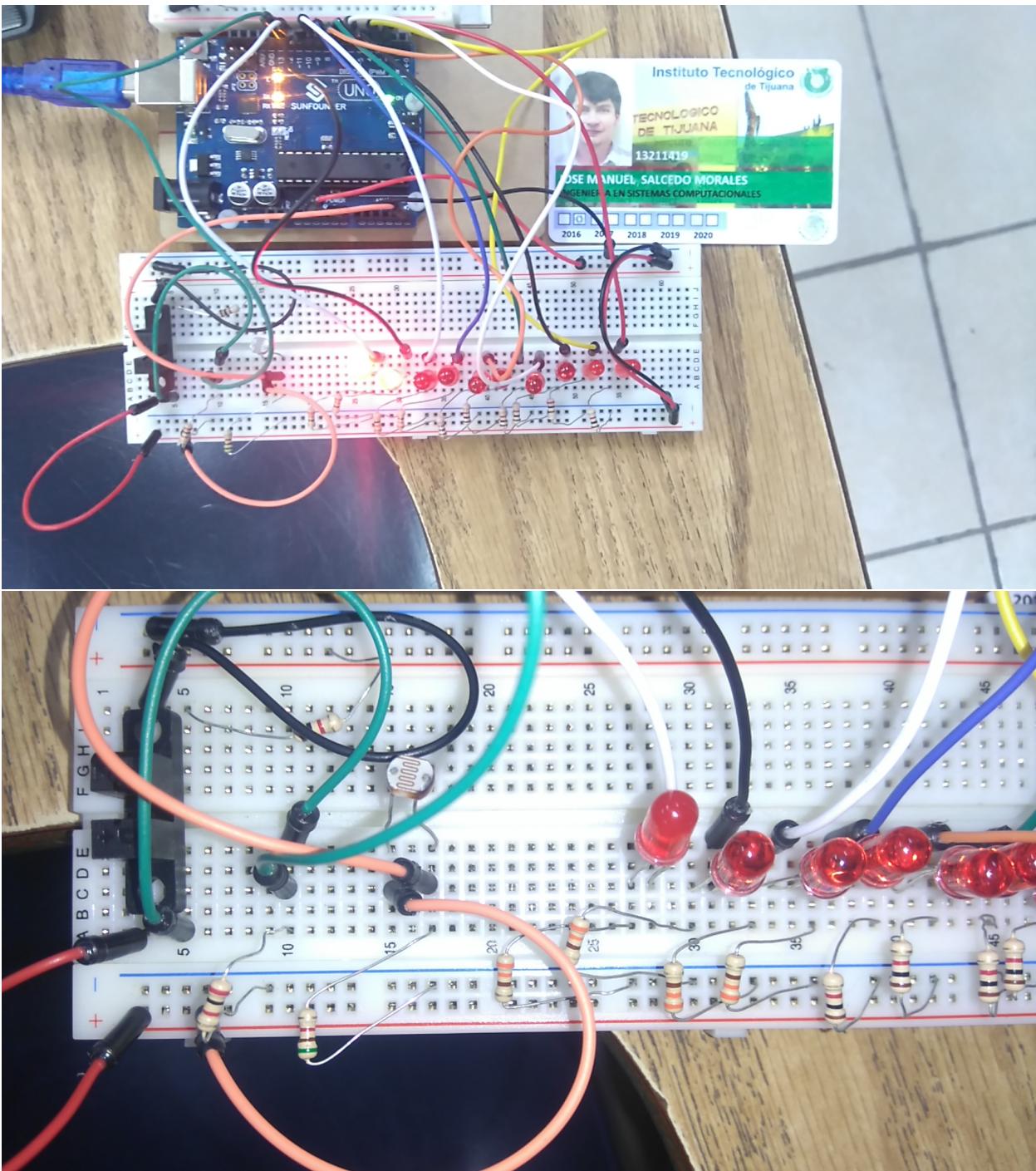
4 Desarrollo

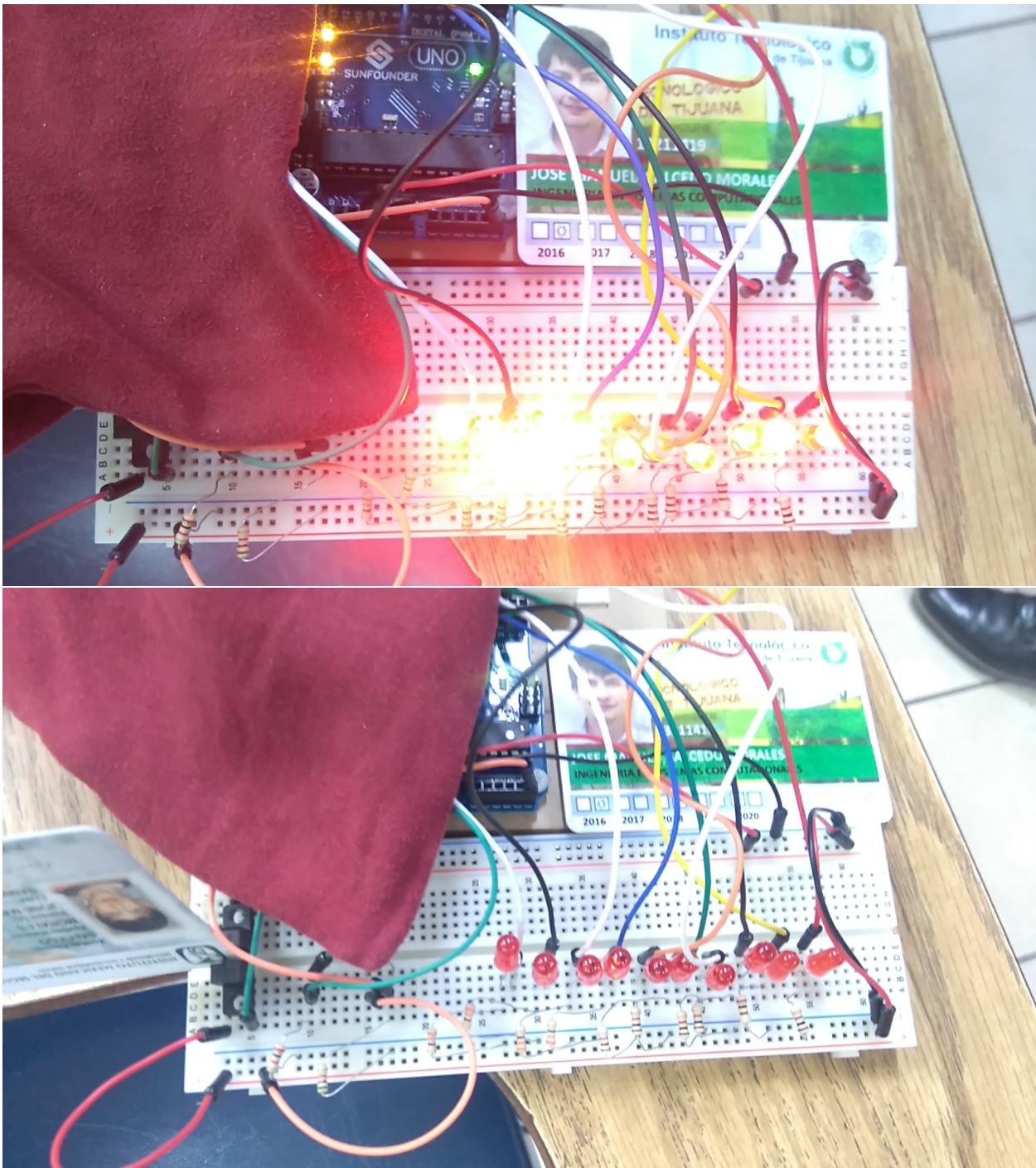
Dependiendo de la intensidad de luz que exista, el arduino lee la resistencia proporcionada por la fotoresistencia.

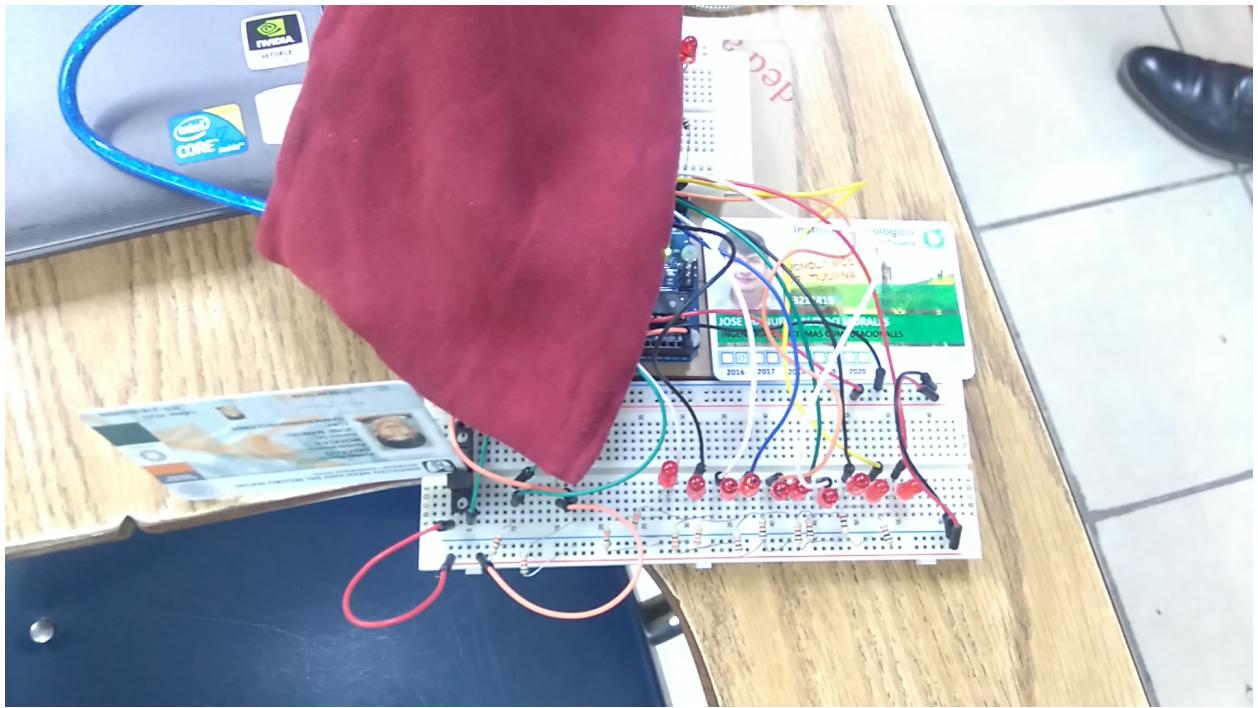
Con este valor, hay un maximo y minimo de voltaje necesario para que prendan o no una cierta cantidad de LED's. De acuerdo al rango alcanzado, se prenderan o apagaran cierta cantidad de LED'S.

Con el optointerruptor, si es interrumpido todos los LED's se mantienen apagados, independientemente de la cantidad de luz que exista.

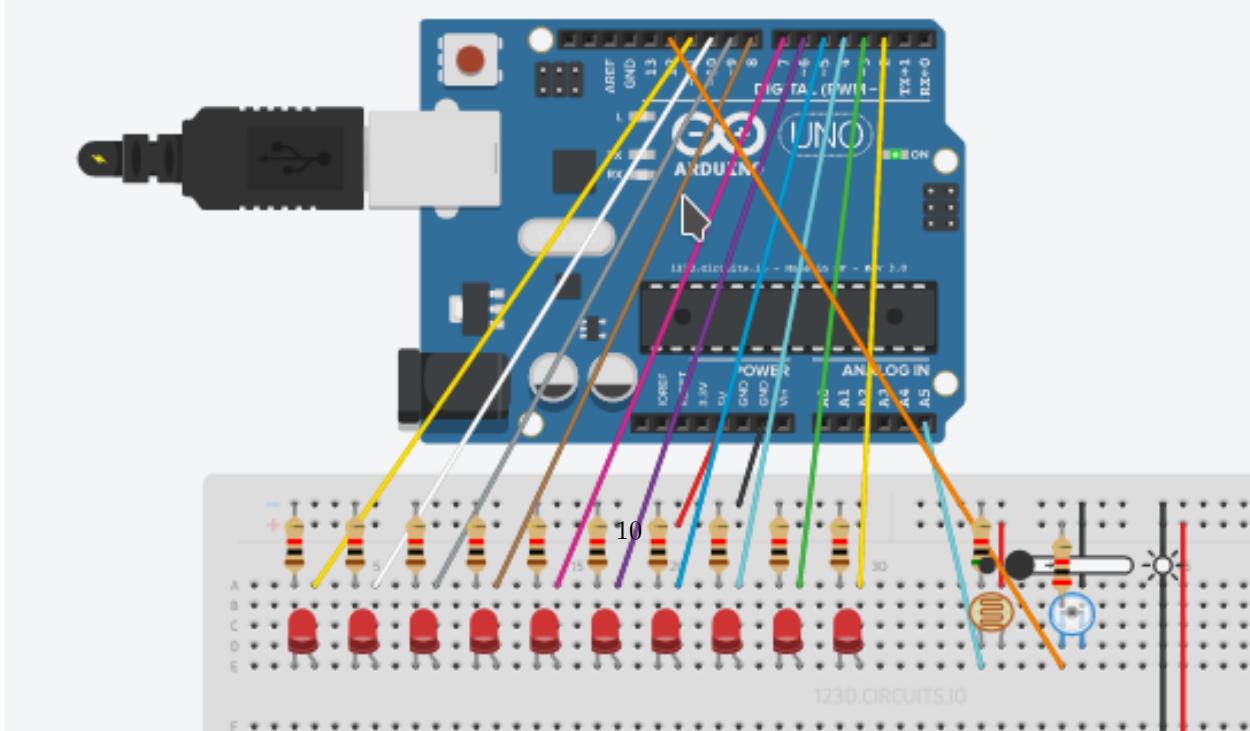
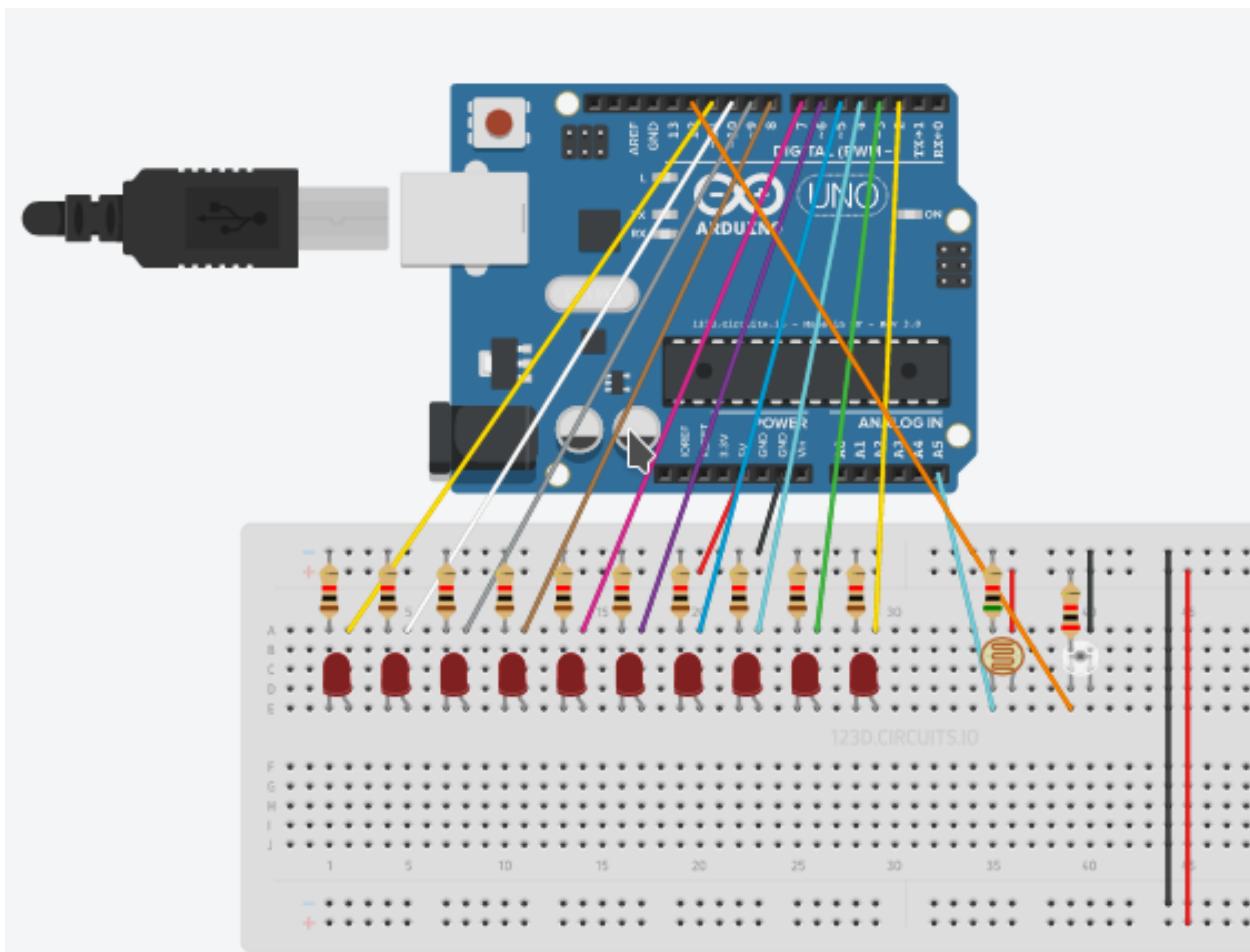
4.1 Imagenes

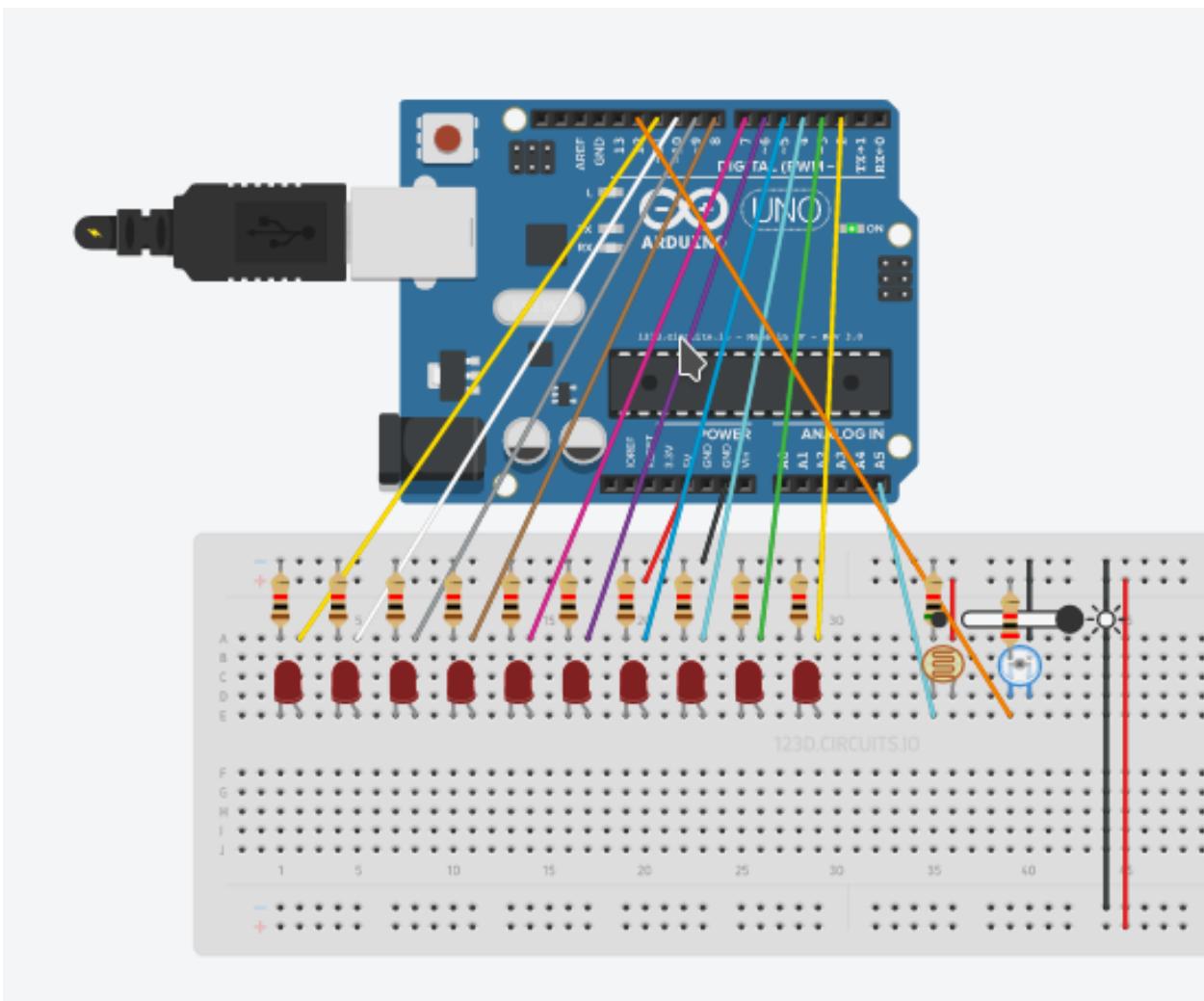


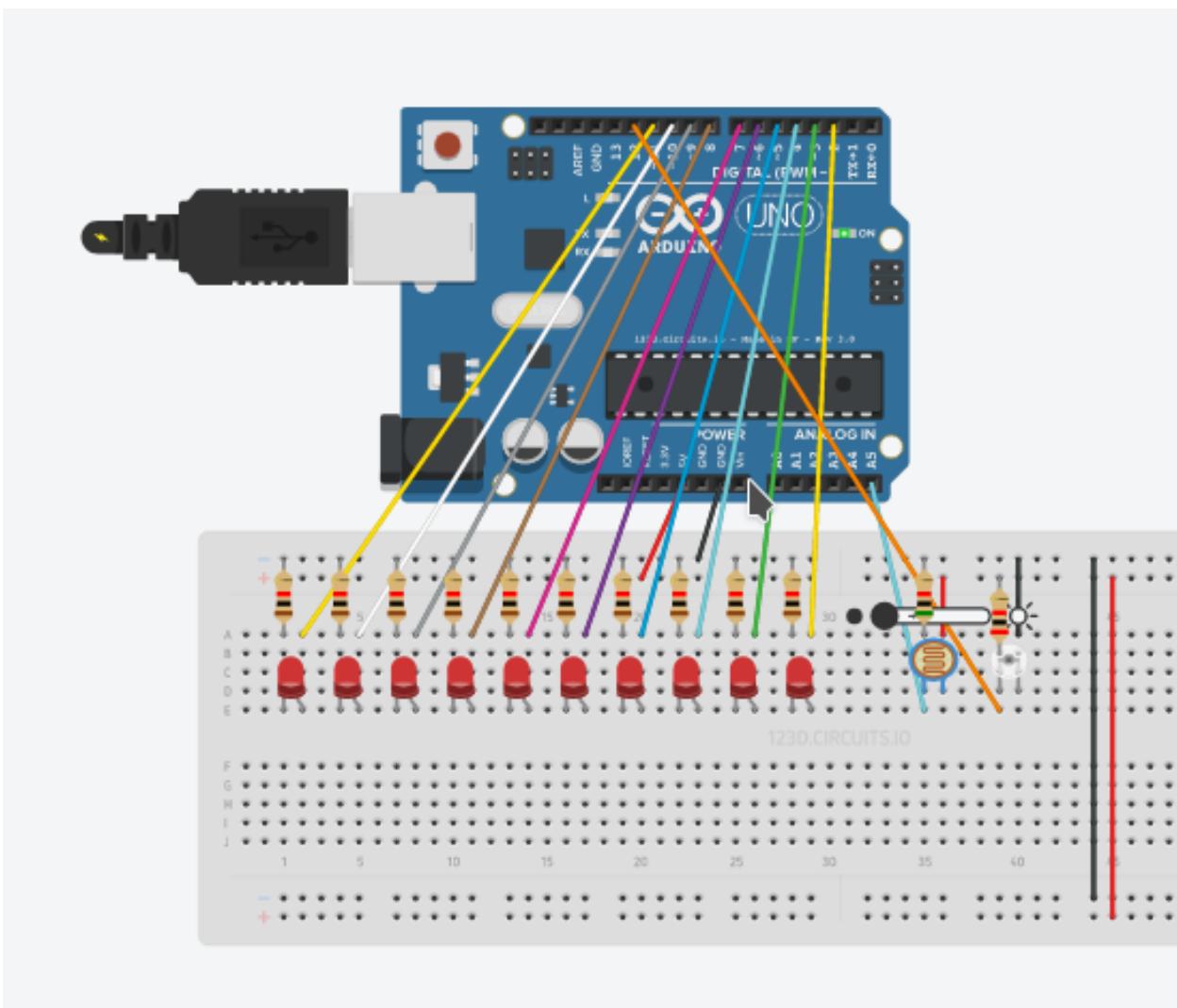


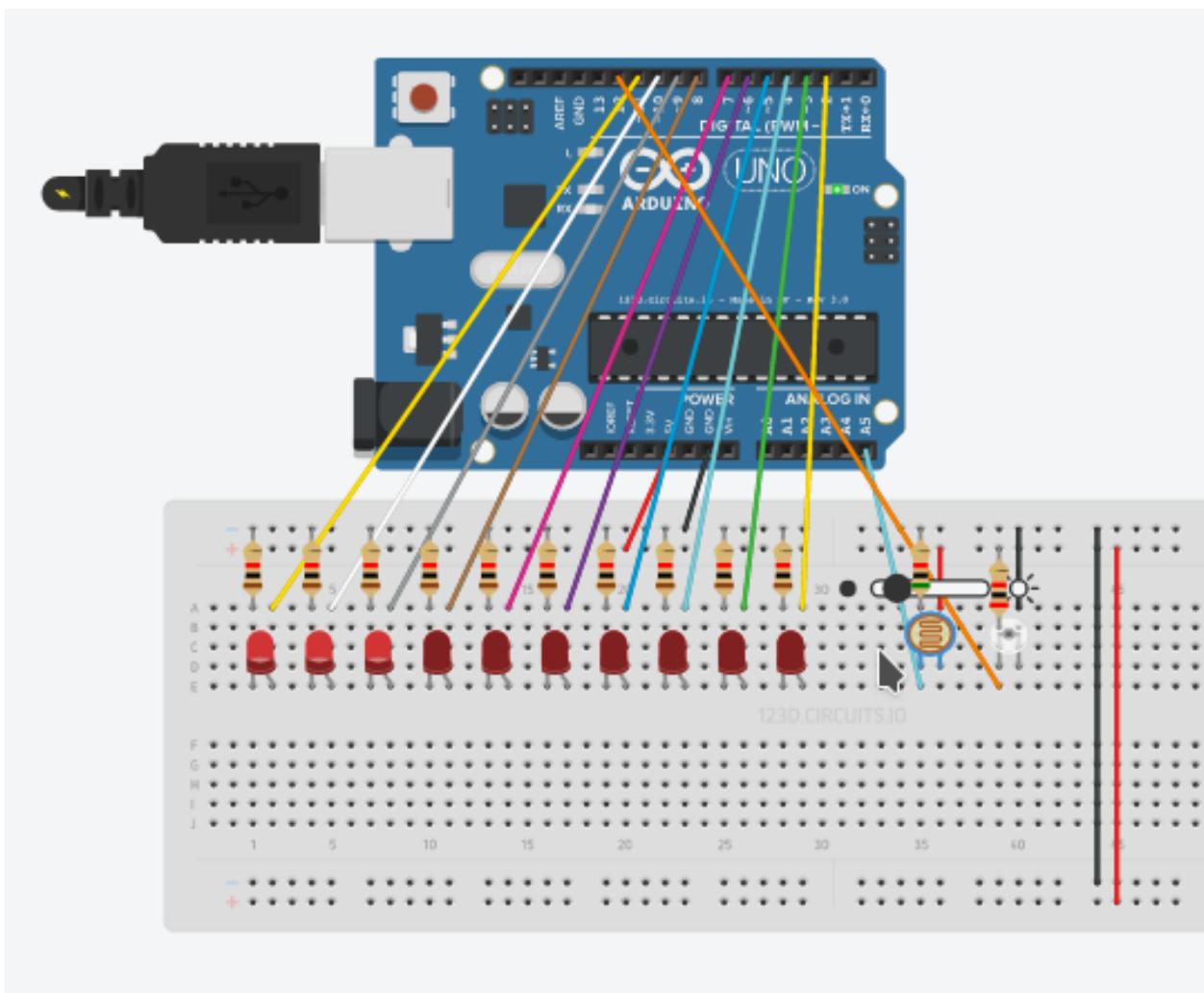


4.2 Diseño









4.3 Código

```
// / \ \ \ \ \ / \ \ \ \ \ / \ \ \ \ \ / \ \ \ \ \ / \ \ \ \ \ / \ \ \ \ \ 
// | / \ \ \ \ D | \ \ \ \ \ \ / \ \ \ \ \ / \ \ \ \ \ / \ \ \ \ \ / \ \ \ \ 
// | \ \ \ \ \ \ / \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ 
// 
// ITT 8Sem SP Practica 12 Arduino y Fotoresistencia
//
// Made by Salcedo Jose
// License: CC-BY-SA 3.0
// Downloaded from: https://circuits.io/circuits/4522757-itt-8sem-sp-practica-12-arduino-y-fotoresistencia

// DATOS ABSTRACTOS.
enum VALOR_ANALOGICO {
    VALOR_ANALOGICO_MINIMO = 0,
    VALOR_ANALOGICO_MAXIMO = 1023
};

// CONSTANTES.
const byte PIN_FOTORESISTENCIA = 5;
const byte PIN_FOTOTRANSISTOR = 12;
const byte LED[] = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11};
const byte CANTIDAD_LEDS = 10;

// Variables globales.
float gValorAnalogoMinimo = 0;
float gValorAnalogoMaximo = 0;

// FUNCIONES.
// Prender una cantidad de led's de acuerdo al valor
// dado de una lectura analoga.
void PrenderLedsPorValor(const byte colecciónLeds[], const byte cantidadLeds, int valorAnalogo)
{
    // Obtener el rango el cual es obtenible por una lectura analoga.
    const int rangoValorAnalogo = gValorAnalogoMaximo - gValorAnalogoMinimo;

    // Obtener el rango de division, el cual define cuantos led's se prenden.
    const int rangoDivision = rangoValorAnalogo / (cantidadLeds + 1);

    // Definir si prender o no cada led individual.
    for (int led = cantidadLeds - 1; led >= 0; led--) {
        // Definir una cantidad, la cual define si se prende
        // o no un led.
        int proporcion = rangoDivision * (led + 1);

        // Si la proporcion supera el rango permitido,
        // prender el led.
        if (proporcion >= valorAnalogo) {
            digitalWrite(colecciónLeds[led], HIGH);
        } else {
            digitalWrite(colecciónLeds[led], LOW);
        }
    }

    // Guardar un distinto valor maximo si el actual es mayor.
    if (valorAnalogo > gValorAnalogoMaximo) {
        gValorAnalogoMaximo = valorAnalogo;
    }

    // Guardar un distinto valor minimo si el actual es menor.
    if (valorAnalogo < gValorAnalogoMinimo) {
        gValorAnalogoMinimo = valorAnalogo;
    }
}

// INICIO.
void setup() {
    Serial.begin(9600);

    // Preparar para salida cada led existente.
    for (byte led = 0; led < CANTIDAD_LEDS; led++) {
        pinMode(LED[led], OUTPUT);
    }

    // Definir valores iniciales para los valores de lectura analoga de maximo y minimo.
    gValorAnalogoMinimo = VALOR_ANALOGICO_MAXIMO / (CANTIDAD_LEDS + 2);
    gValorAnalogoMaximo = gValorAnalogoMinimo * (CANTIDAD_LEDS - 1);
}

// PRINCIPAL.
void loop() {
    // Leer el valor que provee la fotoresistencia en conjunto con una resistencia.
    bool fototransistorActivo = digitalRead(PIN_FOTOTRANSISTOR);

    // Leer el valor que provee la fotoresistencia en conjunto con una resistencia.
    int valorFotoresistencia = analogRead(PIN_FOTORESISTENCIA);

    // Desplegar la informacion obtenida en puerto serial.
    Serial.println("Fototransistor:" + String(fototransistorActivo));
    Serial.println("Fotoresistencia:" + String(valorFotoresistencia));

    // Si el fototransistor no esta tapado, prender y/o apagar los led's
    // de acuerdo a la cantidad de fotoresistencia.
    // Si esta tapado, apagar todos los led's.
    if (fototransistorActivo) {
        // Prender led's de acuerdo al valor dado en la fotoresistencia.
        PrenderLedsPorValor(LED, CANTIDAD_LEDS, valorFotoresistencia);
    } else {

```

```
// Apagar todos los led's.  
PrenderLedsPorValor(LED, CANTIDAD_LEDS, VALOR_ANALOGICO_MAXIMO);  
}  
}
```

5 Conclusión

Considerando que leer y comparar valores provenientes de un componente electronico seria mas complicado sin un arduino, el arduino en si es muy valioso para el control de electronicos.

Referencias

- [1] What is Arduino? - Definition from Techopedia. (n.d.). Retrieved March 26, 2017, from <https://www.techopedia.com/definition/27874/arduino>
- [2] Sparkfun. (n.d.). Photo Interrupter - GP1A57HRJ00F. Retrieved February 13, 2017, from <https://www.sparkfun.com/products/9299>
- [3] Resistorguide.com. (n.d.). Photo resistor. Retrieved from <http://www.resistorguide.com/photoresistor/>