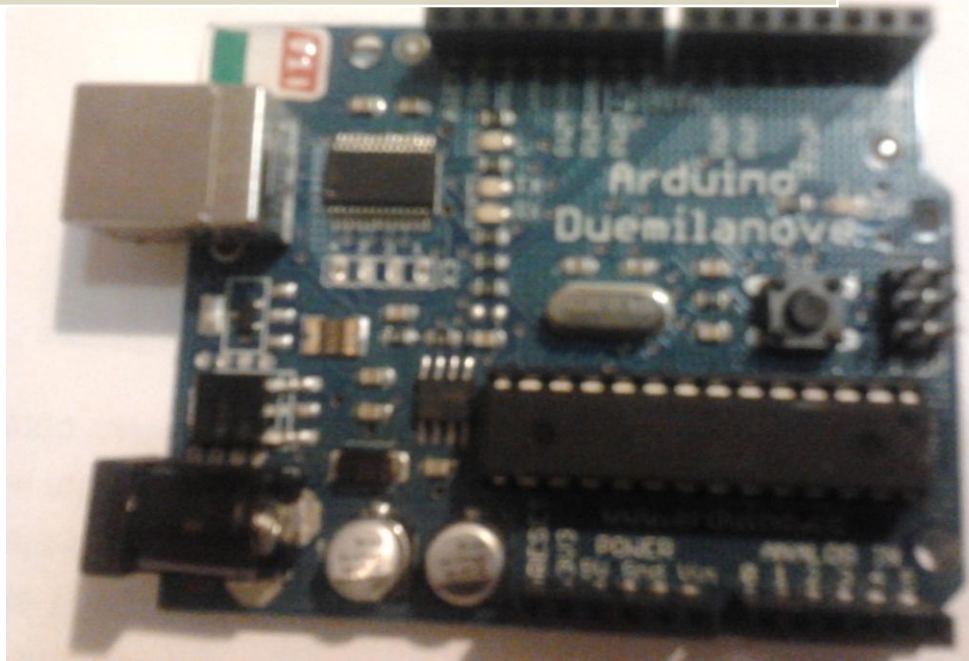


2011/
2012

Desarrollo de sistema inmótico basado en plataforma Arduino



Subproyecto:

Control de Iluminación con DMX512.

IES HERMINO ALMENDROS.

1. Historia del proyecto	2
1. Justificación	2
2. Introducción.....	2
2. Participantes y su colaboración	3
3. Recursos y equipamientos	3
4. Bases técnicas y recursos metodológicos utilizados	5
5. Actividades Realizadas, Resultados y Productos.	6
5.1 Actividades:	6
5.1.1. Desarrollo de un programa en el cual se obtiene el color rojo en el receptor RGB a través de DMX512.	6
5.1.2. <i>Desarrollo de un programa en el cual se obtiene el color verde en el receptor RGB a través de DMX512.</i>	8
5.1.3. Desarrollo de un programa en el cual se obtiene el color azul en el receptor RGB a través de DMX512.....	10
5.1.4. Desarrollo de un programa en el cual se obtiene el color blanco en el receptor RGB a través de DMX512.....	12
5.1.5. Desarrollo de un programa en el cual se obtiene la siguiente secuencia de colores en el receptor RGB a través de DMX512.	14
6. Bibliografía	15
7. Desviaciones de lo previsto y soluciones aplicadas	15
8. Conclusiones y aplicaciones futuras	15
9. Valoración final del proyecto	15

1. Historia del proyecto

1. Justificación

Conforme con la resolución del 5 de abril de 2011 por la que se convocan ayudas destinadas a realizar proyectos de innovación aplicada a la formación profesional del sistema educativo (BOE del 27 de abril de 2011), se presenta la solicitud para **proyecto inmótico con plataforma Arduino**. Este proyecto se aprueba y nuestro centro IES Herminio Almendros accede a él para desarrollar este subproyecto. Este tiene por misión crear un sistema de de iluminación de led a través del bus de comunicación DMX512.

Inicialmente, este proyecto iba a ser desarrollado utilizando el bus de comunicación DALI, pero por diferentes causas se ha decidido trabajar con DMX512.

2. Introducción

DMX512 (acrónimo de Digital Multiplex), es extremadamente fácil, de bajo costo y relativamente robusto. Debido a estas ventajas DMX512 se ha ganado gran popularidad.

Como su propio nombre indica, puede soportar hasta 512 canales para el control de dispositivos, en nuestro caso iluminación tipo LED. Se trata de un protocolo asíncrono unidireccional de transmisión serie que no prevé ninguna forma de retorno entre el receptor y el transmisor, ni ofrece algún tipo de comprobación de errores, o mecanismo de corrección. Por lo tanto, no es adecuado para cualquier aplicación crítica. Los datos se transmiten a 250k baudios utilizando un interface física compatible con el estándar RS-485, mediante dos cables y una tierra.

Cada transmisor DMX512 envía 512 valores de 8 bits, donde cada uno toma valores entre 0 y 255. Por ejemplo el valor 0 representa los Led apagados y 255 máxima intensidad en el color o colores elegidos.

En nuestra aplicación hemos seleccionado el controlador PX24506, cuyas características son:

- 3 canales de salida 3 A (máx.), uno para cada color RGB
- A 12 v DC 108W máximo
- A 24V 216 W máximo

Posibilidad de alimentación a 12V o 24V.

2. Participantes y su colaboración

El proyecto ha sido llevado por:

- Juan Ángel Martínez Alcarria
- José Ramón Navarro López
- José Miguel Martínez Macías
- Juan Gil Gualda
- Carmelo Peláez Muñoz

Juan Ángel ha sido el coordinador del grupo de trabajo, ha estado en contacto con la coordinadora del proyecto y ha acudido a todas las reuniones en representación del grupo.

Cada integrante del grupo se ha centrado más en el tema que le era más afín.

3. Recursos y equipamientos

Para la realización del subproyecto hemos necesitado los siguientes recursos y equipamiento:

- PC con tarjeta de puesto serie integrada
- F.A de 24v DC
- Winkhel 100
- Tira de 5m de LED RGB a 12V DC.
- Decodificador DMX512 PX24506

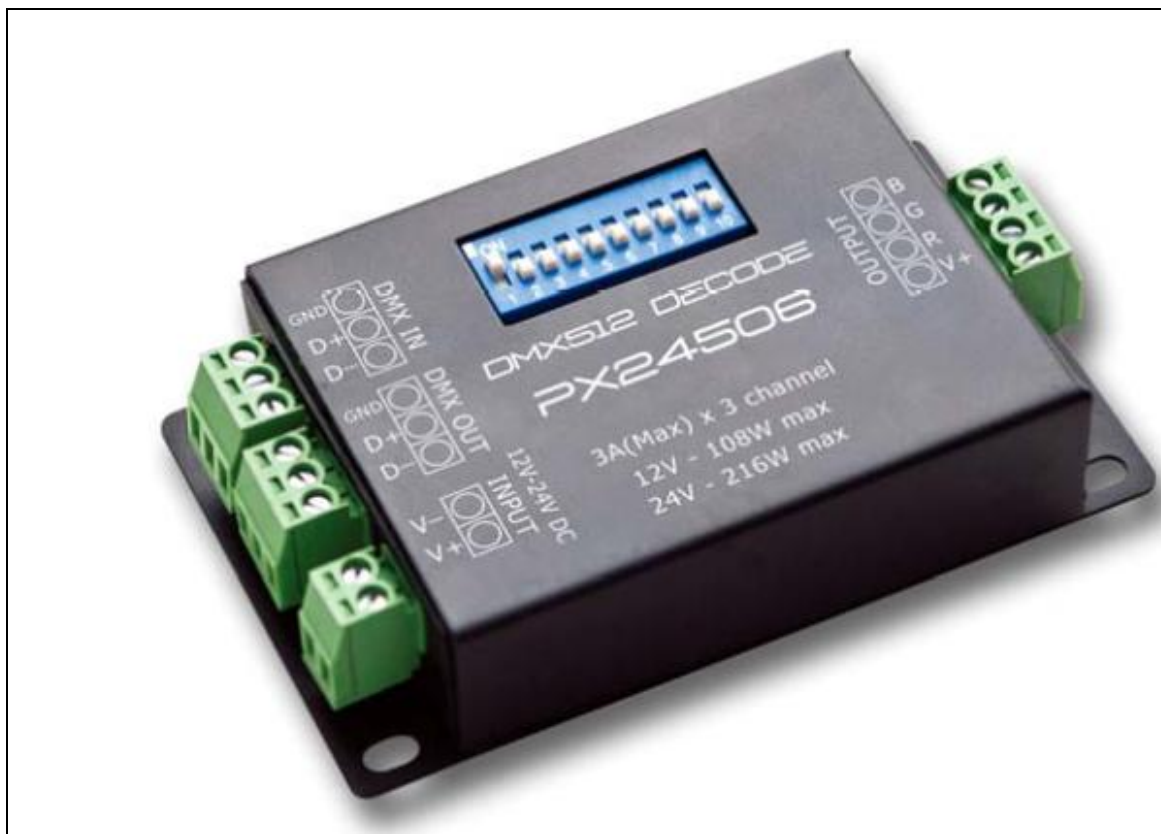
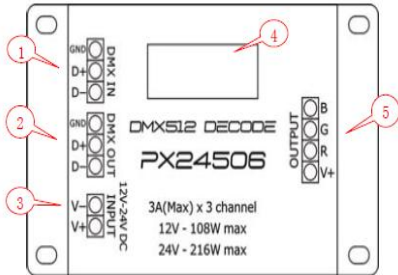

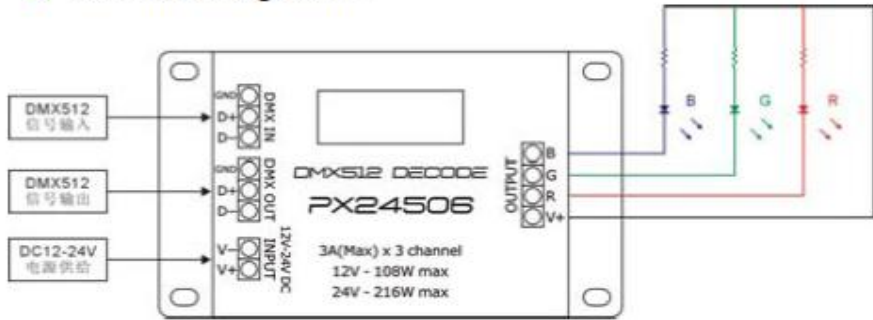


Imagen 1: decodificador DMX512 PX24506

Características técnicas	<p>Decode CH.: 3CH</p> <p>Input Signal: DMX-512/1990 digital signal</p> <p>Output Signal: can drive 3A(Each CH.)</p> <p>Power Supply: DC 12~24V</p> <p>Power Dis.: <1W</p> <p>Power Output: <216W(24V);<108W(12V)</p> <p>Operating Temp.: 0~50°C</p> <p>Size: 86(mm)*51(mm)*17.8(mm)</p> <p>Weight: 110g</p>
--------------------------	---

Conexión		<p>(1) DMX signal input interface</p> <p>(2) DMX signal output interface</p> <p>(3) Power input interface</p> <p>(4) Address setting interface</p> <p>(5) Driver output interface</p>
----------	---	---

Interface DMX	<p>◆ DMX signal interface</p> 
Circuito eléctrico típico	<p>◆ Circuit Diagram 1</p> 

4. Bases técnicas y recursos metodológicos utilizados

Este decodificador para arduino y más concretamente para Winkhel sólo puede funcionar con los micros Atmel 168 y 328, cómo se puede leer en este enlace <http://www.mathertel.de/Arduino/DMXSerial.aspx>, con lo cual sólo podríamos hacerle funcionar en el Winkhel 400 y en el Winkhel 100. Para facilitar la programación hemos usado la librería DMXSerial.h para Arduino.

Para configurar el número de nodo se hace seleccionando los microswitches del frontal del decodificador, teniendo en cuenta que si le asignamos el nodo 1, no podremos usar los canales 1, 2, 3 que quedan reservados para el rojo (1), verde (2) y azul (3) (RGB), con lo cual el siguiente módulo decodificador lo tendríamos que direccionar en el nodo 4.

A modo de ejemplo si en el canal 1, asignamos el valor 255, y tanto al canal 2 y 3 el valor 0, se obtendría el color rojo de máxima intensidad.

En principio si usásemos un arduino como por ejemplo el Arduino Duemilanove, y quisiéramos crear un bus DMX, deberíamos de usar para ello 3 salidas de control de ancho de pulso, PWM, para poder realizar el bus y controlar la intensidad de cada color., para leer más al respecto se puede visitar el siguiente enlace:

http://tomekness.files.wordpress.com/2007/01/dmx_and_arduino_tutorial.pdf o

<http://www.mathertel.de/Arduino/DMXSerial.aspx>.

5. Actividades Realizadas, Resultados y Productos.

Hemos seleccionado el Winkhel 100, que tiene el micro 328 y que también tiene F.A de 12 V, necesarios para alimentar el bus y los diodos LED.

5.1 Actividades:

5.1.1. Desarrollo de un programa en el cual se obtiene el color rojo en el receptor RGB a través de DMX512.

```
#include <DMXSerial.h>

int RedList[]    = {255,255, 255,255,255,255};
int GreenList[] = { 0,0,0, 0, 0, 0};
int BlueList[]  = { 0, 0, 0, 0, 0, 0};

int RedLevel, GreenLevel, BlueLevel;

int RedNow = 0;
int GreenNow = 0;
int BlueNow = 0;

int state = 0;

void setup() {
  DMXSerial.init(DMXController);
  /* DMXSerial.write(1, 255);
  DMXSerial.write(2, 0);
  DMXSerial.write(3,0);*/
}

void loop() {
  RedLevel = RedList[state];
  GreenLevel = GreenList[state];
  BlueLevel = BlueList[state];

  if ((RedLevel == RedNow) && (GreenLevel == GreenNow) && (BlueLevel == BlueNow)) {
    state += 1;
    if (state == 6)
      state = 0;

  }else {
    if (RedNow < RedLevel) RedNow++;
    if (RedNow > RedLevel) RedNow--;
    DMXSerial.write(1, RedNow);
  }
}
```



```
if (GreenNow < GreenLevel)  GreenNow++;  
if (GreenNow > GreenLevel)  GreenNow--;  
DMXSerial.write(2, GreenNow);  
  
if (BlueNow < BlueLevel)   BlueNow++;  
if (BlueNow > BlueLevel)   BlueNow--;  
DMXSerial.write(3, BlueNow);  
}  
  
delayMicroseconds(2000);
```

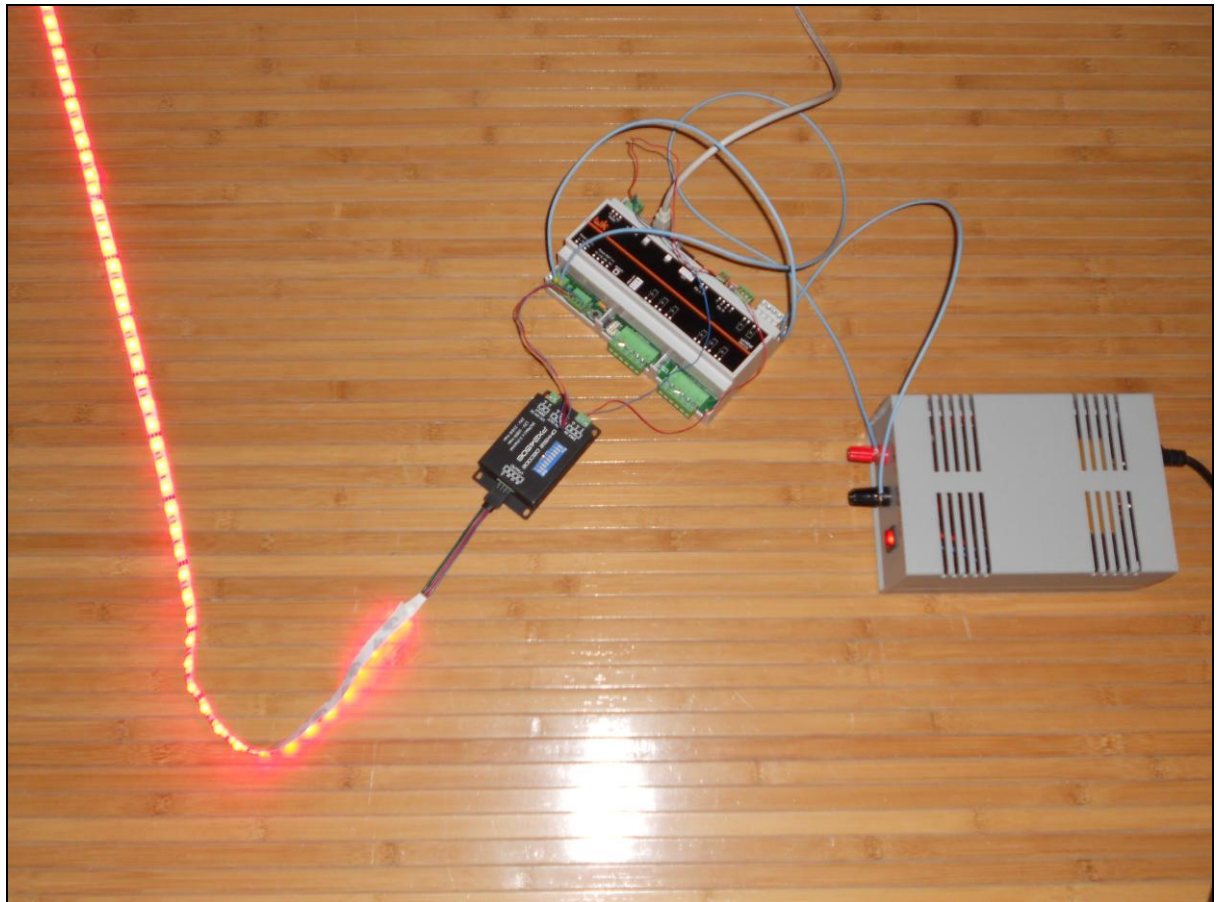


Imagen 2: color rojo a máxima intensidad (255)

5.1.2. Desarrollo de un programa en el cual se obtiene el color verde en el receptor RGB a través de DMX512.

```
#include <DMXSerial.h>

int RedList[] = { 0,0,0, 0, 0, 0};
int GreenList[] = {255,255, 255,255,255,255};
int BlueList[] = { 0, 0, 0, 0, 0, 0};

int RedLevel, GreenLevel, BlueLevel;

int RedNow = 0;
int GreenNow = 0;
int BlueNow = 0;

int state = 0;

void setup() {
  DMXSerial.init(DMXController);
  /* DMXSerial.write(1, 255);
  DMXSerial.write(2, 0);
  DMXSerial.write(3,0);*/
}

void loop() {
  RedLevel = RedList[state];
  GreenLevel = GreenList[state];
  BlueLevel = BlueList[state];

  if ((RedLevel == RedNow) && (GreenLevel == GreenNow) && (BlueLevel == BlueNow)) {
    state += 1;
    if (state == 6)
      state = 0;

  }else {
    if (RedNow < RedLevel) RedNow++;
    if (RedNow > RedLevel) RedNow--;
    DMXSerial.write(1, RedNow);

    if (GreenNow < GreenLevel) GreenNow++;
    if (GreenNow > GreenLevel) GreenNow--;
    DMXSerial.write(2, GreenNow);
```

```
if (BlueNow < BlueLevel)  BlueNow++;  
if (BlueNow > BlueLevel)  BlueNow--;  
DMXSerial.write(3, BlueNow);  
}  
  
delayMicroseconds(2000);  
}
```

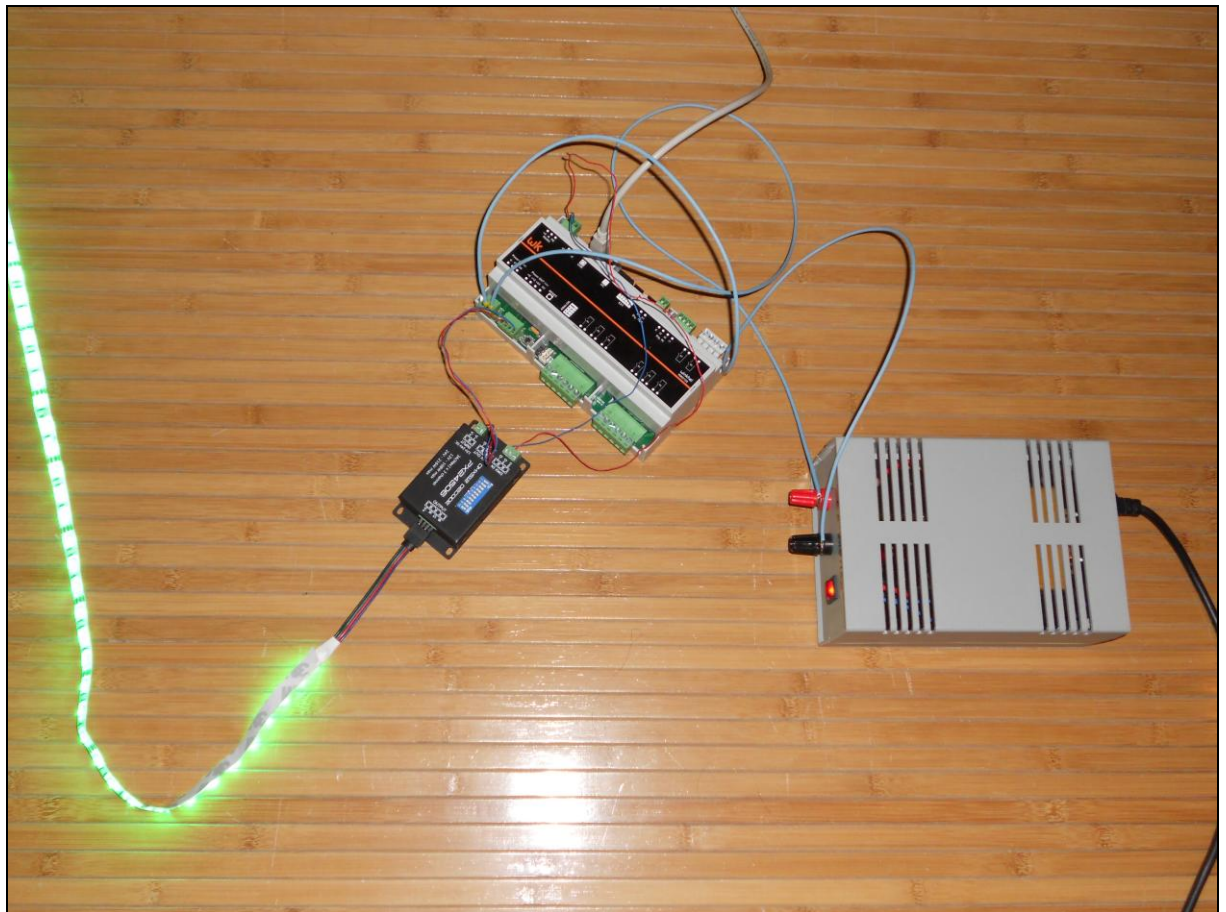


Imagen 3: color verde a máxima intensidad (255)

5.1.3. Desarrollo de un programa en el cual se obtiene el color azul en el receptor RGB a través de DMX512.

```
#include <DMXSerial.h>

int RedList[] = { 0,0,0, 0, 0, 0};
int GreenList[] = { 0, 0, 0, 0, 0, 0};
int BlueList[] = {255,255, 255,255,255,255};
int RedLevel, GreenLevel, BlueLevel;

int RedNow = 0;
int GreenNow = 0;
int BlueNow = 0;

int state = 0;

void setup() {
  DMXSerial.init(DMXController);
  /* DMXSerial.write(1, 255);
  DMXSerial.write(2, 0);
  DMXSerial.write(3,0);*/
}

void loop() {
  RedLevel = RedList[state];
  GreenLevel = GreenList[state];
  BlueLevel = BlueList[state];

  if ((RedLevel == RedNow) && (GreenLevel == GreenNow) && (BlueLevel == BlueNow)) {
    state += 1;
    if (state == 6)
      state = 0;

  }else {
    if (RedNow < RedLevel) RedNow++;
    if (RedNow > RedLevel) RedNow--;
    DMXSerial.write(1, RedNow);

    if (GreenNow < GreenLevel) GreenNow++;
    if (GreenNow > GreenLevel) GreenNow--;
    DMXSerial.write(2, GreenNow);

    if (BlueNow < BlueLevel) BlueNow++;
    if (BlueNow > BlueLevel) BlueNow--;
    DMXSerial.write(3, BlueNow);
  }
}
```

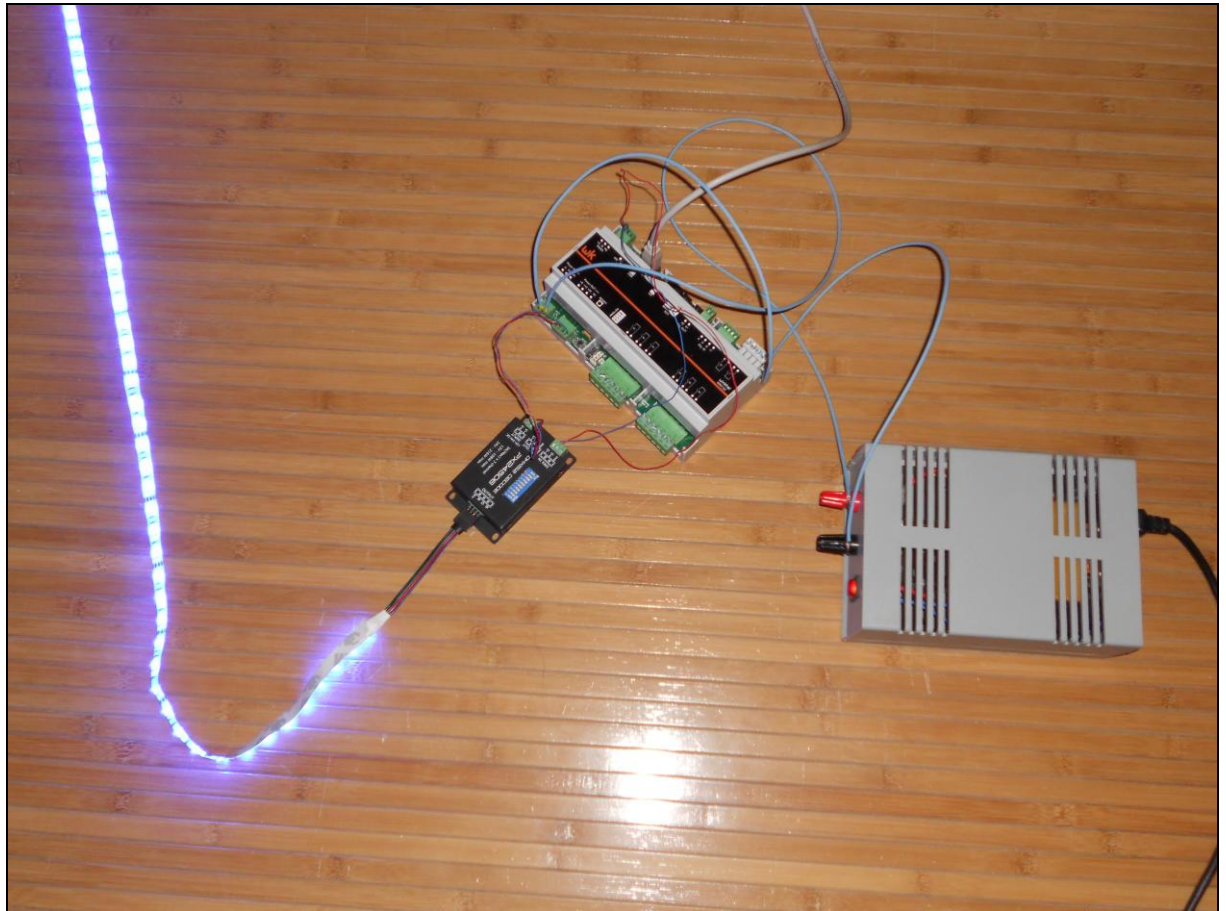


Imagen 4: color azul a máxima intensidad (255)

5.1.4. Desarrollo de un programa en el cual se obtiene el color blanco en el receptor RGB a través de DMX512.

```
#include <DMXSerial.h>

int RedList[] = {255,255, 255,255,255,255};
int GreenList[] = {255,255, 255,255,255,255};
int BlueList[] = {255,255, 255,255,255,255};
int RedLevel, GreenLevel, BlueLevel;

int RedNow = 0;
int GreenNow = 0;
int BlueNow = 0;

int state = 0;

void setup() {
  DMXSerial.init(DMXController);
  /* DMXSerial.write(1, 255);
  DMXSerial.write(2, 0);
  DMXSerial.write(3,0);*/
}

void loop() {
  RedLevel = RedList[state];
  GreenLevel = GreenList[state];
  BlueLevel = BlueList[state];

  if ((RedLevel == RedNow) && (GreenLevel == GreenNow) && (BlueLevel == BlueNow)) {
    state += 1;
    if (state == 6)
      state = 0;

  }else {
    if (RedNow < RedLevel) RedNow++;
    if (RedNow > RedLevel) RedNow--;
    DMXSerial.write(1, RedNow);

    if (GreenNow < GreenLevel) GreenNow++;
    if (GreenNow > GreenLevel) GreenNow--;
    DMXSerial.write(2, GreenNow);

    if (BlueNow < BlueLevel) BlueNow++;
    if (BlueNow > BlueLevel) BlueNow--;
    DMXSerial.write(3, BlueNow);
  }

  delayMicroseconds(2000);
}
```

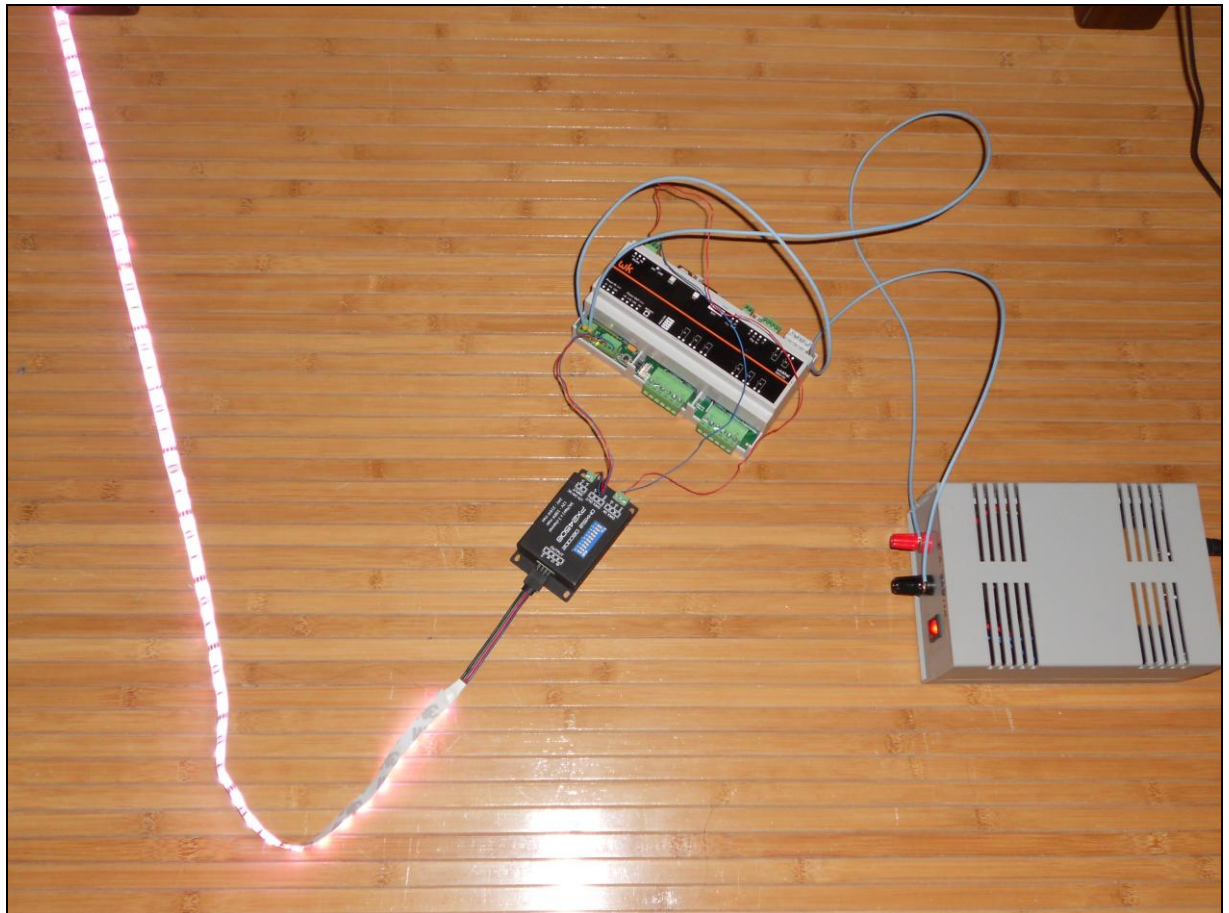



Imagen 5: color blanco a máxima intensidad (255)

5.1.5. Desarrollo de un programa en el cual se obtiene la siguiente secuencia de colores en el receptor RGB a través de DMX512.

```
#include <DMXSerial.h>

int RedList[] = {255, 128, 0, 0, 0, 128};
int GreenList[] = { 0, 128, 255, 128, 0, 0};
int BlueList[] = { 0, 0, 0, 128, 255, 128};

int RedLevel, GreenLevel, BlueLevel;

int RedNow = 0;
int GreenNow = 0;
int BlueNow = 0;

int state = 0;

void setup() {
    DMXSerial.init(DMXController);
    /* DMXSerial.write(1, 255);
    DMXSerial.write(2, 0);
    DMXSerial.write(3, 0);*/
}

void loop() {
    RedLevel = RedList[state];
    GreenLevel = GreenList[state];
    BlueLevel = BlueList[state];

    if ((RedLevel == RedNow) && (GreenLevel == GreenNow) && (BlueLevel == BlueNow)) {
        state += 1;
        if (state == 6)
            state = 0;

    }else {
        if (RedNow < RedLevel) RedNow++;
        if (RedNow > RedLevel) RedNow--;
        DMXSerial.write(1, RedNow);

        if (GreenNow < GreenLevel) GreenNow++;
        if (GreenNow > GreenLevel) GreenNow--;
        DMXSerial.write(2, GreenNow);

        if (BlueNow < BlueLevel) BlueNow++;
        if (BlueNow > BlueLevel) BlueNow--;
        DMXSerial.write(3, BlueNow);
    }

    delayMicroseconds(2000);
}
```

6. Bibliografía

Entre otras páginas web de consulta, podemos destacar:

<http://www.arduino.cc/>

www.formacionconarduino.com/moodle/login/index.php

<https://github.com/>

<http://www.mathertel.de/Arduino/DMXSerial.aspx>

<http://en.wikipedia.org/wiki/DMX512>

7. Desviaciones de lo previsto y soluciones aplicadas

En una primera aproximación al proyecto "Desarrollo de sistema inmótico basado en plataforma Arduino", este Subproyecto se iba a desarrollar con protocolo DALI.

Por problemas técnicos se decidió cambiar a iluminación LED RGB mediante protocolo DMX512.

8. Conclusiones y aplicaciones futuras

Modbus es un protocolo de comunicación muy interesante desde el punto de vista didáctico.

Arduino es una plataforma abierta basada en software y hardware flexibles y fáciles de usar. Es idóneo para la creación de prototipos, así, en nuestro Centro tenemos previsto su uso para módulos relacionados fundamentalmente con la Domótica, Regulación y Control de iluminación.

9. Valoración final del proyecto

El proyecto ha resultado muy interesante porque la iluminación con LED supone un importante ahorro energético y está de actualidad, además su regulación es fácil si se usa el protocolo DMX512.

Con este proyecto se ha actualizado y renovado conocimientos.