



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



**ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA**

Curso Académico:

ÍNDICE MEMORIA

| | |
|--------------------------------------------------------------------|-----------|
| ÍNDICE ILUSTRACIONES..... | 3 |
| RESUMEN | 5 |
| TRABAJOS PREVIOS..... | 7 |
| 1. INTRODUCCIÓN | 9 |
| 2. ANTECEDENTES, MOTIVACIÓN Y JUSTIFICACIÓN | 13 |
| 2.1 EQUIPAMIENTO DISPONIBLE EN EL MERCADO | 13 |
| 2.2 MOTIVACIÓN Y JUSTIFICACIÓN | 13 |
| 3. OBJETIVOS | 15 |
| 4. MATERIALES Y MÉTODOS | 17 |
| 4.1 SOFTWARE..... | 17 |
| 4.1.1 Lenguaje de programación: Arduino programming language | 17 |
| 4.1.2 Lenguaje de programación: LabVIEW..... | 18 |
| 4.2 HARDWARE | 19 |
| 4.2.1 Microcontrolador: Arduino | 19 |
| 4.2.2 Sensor TCS3200..... | 19 |
| 4.2.3 Motor paso a paso..... | 20 |
| 4.2.4 PCB..... | 21 |
| 4.2.5 Interruptor de final de carrera | 21 |
| 4.2.6 Estructura | 21 |
| 4.3 CALIBRACIÓN..... | 22 |
| 5. DESARROLLO Y RESULTADOS..... | 23 |
| 5.1 HARDWARE..... | 23 |
| 5.2 CALIBRACIÓN..... | 27 |
| 5.3 SOFTWARE..... | 32 |
| 5.3.1 Arduino | 32 |
| 5.3.2 LabVIEW..... | 35 |
| 5.3.3 Comunicación Arduino-LabVIEW | 36 |
| 5.4 RESULTADOS | 37 |
| 6. CONCLUSIONES..... | 41 |
| 6.1 OBJETIVOS SUPERADOS | 41 |
| 6.2 PROPUESTAS DE MEJORA | 41 |
| 7. PRESUPUESTO | 43 |
| 8. BIBLIOGRAFÍA | 45 |
| 9. ANEXOS | 49 |
| ANEXO I – MANUAL DE USUARIO | 49 |
| ANEXO II – DOCUMENTACIÓN TÉCNICA SENSOR DE COLOR TCS3200D | 50 |
| ANEXO III – DOCUMENTACIÓN TÉCNICA PLACA ARDUINO | 50 |

| | |
|------------------------------------------------------------------------------------|----|
| ANEXO IV – DOCUMENTACIÓN TÉCNICA MOTOR PASO A PASO | 51 |
| ANEXO V – DOCUMENTACIÓN TÉCNICA CONTROLADOR MOTOR PASO A PASO POLOLU DRV8825 | 51 |
| ANEXO VI – CÓDIGO ARDUINO..... | 52 |
| ANEXO VII – CÓDIGO LABVIEW..... | 69 |
| ANEXO VIII – PCB TRABAJO PREVIO | 70 |
| ANEXO IX – ESQUEMÁTICO PCB..... | 71 |
| ANEXO X – DISTRIBUCIÓN PCB | 72 |
| ANEXO XI – CUADRO DE PRECIOS DESCOMPUESTOS | 73 |

ÍNDICE ILUSTRACIONES

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Fig.1 Espectro solar (Carlos, 2007) | 9 |
| Fig.2 Espectro bombilla incandescente (Carlos, 2007) | 10 |
| Fig.3 Espectro bombilla led (jeffkrol, 2013) | 10 |
| Fig.4 Respuesta visión fotópica y escotópica (González, 2006) | 12 |
| Fig.5 Diferentes lenguajes de programación (campusMVP, 2014) | 17 |
| Fig.6 Tabla para la selección del formato de pasos en el controlador PLOLU DRV8825 (Pololu, 2014)..... | 20 |
| Fig.7 Interruptor final de carrera (cetronic, s.f.) | 21 |
| Fig.8 Montaje para medir la intensidad de la fuente lumínica a la izquierda (Fig.8.a) y espectrofotómetro utilizado para calibrar la fuente a la derecha (Fig.8.b) (X-Rite, s.f.) | 22 |
| Fig. Máscara utilizada para la creación de la PCB final. | 25 |
| Fig.9 Vistas de la PCB acabada con todos los componentes ensamblados incluida la placa Arduino Duemilanove. Perfil derecho en la parte superior derecha (Fig.9.a), alzado en la parte superior izquierda (Fig.9.b) y planta en la parte inferior izquierda (Fig.9.c) | 26 |
| Fig.10 Estructura final. Vista en alzado en la parte superior izquierda (Fig.10.a), vista en planta en la parte inferior izquierda (Fig.10.b) y vista de perfil izquierdo en la parte superior derecha (Fig.10.c). | 27 |
| Fig.11 Calibrado de la fuente a 470 nm de longitud de onda. | 28 |
| Fig.12 Calibrado de la fuente a 520 nm de longitud de onda. | 28 |
| Fig.13 Calibrado de la fuente a 640 nm de longitud de onda. | 29 |
| Fig.14 Tabla de eficacia luminosa de radiación (González, 2006)..... | 29 |
| Fig.15 Calibrado del prototipo a 470 nm de longitud de onda | 30 |
| Fig.16 Calibrado del prototipo a 520 nm de longitud de onda | 30 |
| Fig.17 Calibrado del prototipo a 640 nm de longitud de onda | 31 |
| Fig.18 Función para controlar los motores paso a paso | 32 |
| Fig. 19 Función para medir la intensidad lumínica | 33 |
| Fig.20 Parámetros para la selección de escala de salida y filtro de los fotodiodos en el sensor TCS3200D (TAOS, s.f.)..... | 33 |
| Fig.21 Recorrido del prototipo a la hora de hacer mediciones. Recorrido en alzado en la parte superior (Fig.21.a) y en planta en la parte inferior (Fig.21.b) | 34 |
| Fig.22 Interfaz gráfica vista por el usuario creada con LabVIEW | 35 |
| Fig.23 Resultados obtenidos en el primer ensayo con bombilla incandescente. | 37 |

| | |
|--------------------------------------------------------------------------------------------------------|----|
| Fig.24 Valores representativos del primer ensayo..... | 38 |
| Fig.25 Resultados obtenidos en el segundo ensayo con bombilla led..... | 39 |
| Fig.26 Resultados obtenidos en el tercer ensayo realizado con luz solar..... | 40 |
| Fig.27 Interfaz gráfica vista por el usuario creada con LabVIEW. | 49 |
| Fig.28 Tabla resumen del ángulo abarcado por el sensor dependiendo del nº de medidas tomadas. | 49 |
| Fig.29 Diagrama de bloques LabVIEW en el caso de que la estructura sea TRUE. | 69 |
| Fig.30 Distribución PCB final del trabajo previo realizado. | 70 |
| Fig.31 Esquemático de la PCB final. | 71 |
| Fig.32 Distribución de la PCB final..... | 72 |

RESUMEN

En el presente trabajo aporta una solución novedosa a la medida de luz enfocada a la conservación preventiva de obras de arte. La iluminación en el ámbito de las obras de arte y patrimonio tiene un papel fundamental en dos aspectos: la alteración de la percepción de la obra y la conservación de la misma. En este caso el trabajo se centrará en la conservación de la obra tratando de evitar deterioros a causa de la iluminación.

En el ámbito de la conservación de obras de arte se necesita una correcta iluminación no sólo para la apreciación de la obra, también para no dañar el tejido sobre la que ésta está proyectada. Dependiendo del tipo de obra que queramos exponer se recomiendan unos límites de iluminación (ICOM, 2010). En este trabajo se pretende medir la iluminancia de una habitación para comprobar si su iluminación permite exponer determinadas obras de arte.

El objetivo de este trabajo es desarrollar un sensor de luz mediante el convertidor de color a frecuencia TCS3200D y dirigible mediante dos motores paso a paso. Se medirá la intensidad de la luz en las tres longitudes predominantes en la iluminación led, correspondientes a los colores rojo, verde y azul. La característica novedosa que aporta este prototipo es la capacidad de realizar lecturas a diferentes orientaciones, pudiendo obtener la orientación a la que el sensor capta la mayor intensidad lumínica y de ésta forma obtener el valor máximo de iluminancia recibida por la obra.

TRABAJOS PREVIOS

El propósito de este trabajo al comienzo fue construir un sensor de iluminación led el cual utilizase como captador diodos led. A pesar que finalmente no se pudo conseguir el objetivo se ha creído oportuno destacar los puntos más importantes que fueron realizados ya que parte de ellos se han utilizado en el trabajo final.

Un diodo led (light-emitting diode) es un diodo cuyo propósito es emitir luz cuando una corriente lo atraviesa, esta corriente atraviesa el diodo porque se aplica una diferencia de potencial entre sus dos patas de entrada mediante una fuente de alimentación externa. En el trabajo previo se quería utilizar el diodo de manera inversa, es decir cuando una intensidad lumínica incide sobre el led entre las patas de éste se genera una diferencia de potencial proporcional a la intensidad recibida. Los diodos led son sensibles a longitudes de onda más cortas que en las que el propio led puede emitir, es decir un led verde podrá medir intensidad proveniente de un led azul, pero no de uno rojo, mientras que el rojo podrá medir tanto la azul, como la verde como el propio rojo (Mims, s.f.).

Bajo esta premisa se elaboró un primer prototipo de sensor. Se diseñó y construyó una placa de circuito impreso, en adelante PCB (Printed Circuit Board) donde estaban conectados una circunferencia con 8 ledes, dos amplificadores de instrumentación en serie para amplificar la tensión de los ledes en dos etapas, los controladores de los motores que moverían el prototipo y un multiplexor. El multiplexor es un dispositivo que puede escoger entre varias señales y dar como salida sólo una, de esta manera los 8 ledes estaban conectados al multiplexor y mediante código informático.

Una vez construida la PCB se observó que los amplificadores no podían amplificar la diferencia de potencial que generaban los ledes porque ésta era demasiado pequeña y el multiplexor no la podía leer. Antes de hacer la PCB se probaron los ledes y si que se consiguió amplificar su señal, el problema es que en la prueba no se tuvo en cuenta el multiplexor. Los amplificadores eran capaces de amplificar la pequeña señal proporcionada por los ledes, sin embargo ésta era demasiada pequeña para que el multiplexor pudiera trabajar con ella.

Una vez localizado el problema se decidió abandonar el prototipo ya que era un problema de difícil solución, así que se optó por utilizar el sensor de color TCS3200D y utilizarlo como sensor de intensidad lumínica como más adelante se explicará. Lejos de ser una pérdida de tiempo las capacidades adquiridas para mover los motores, diseñar y crear la PCB y manejar la programación en Arduino fueron fundamentales para el trabajo final, es por eso que se ha creído oportuno comentarlo en la memoria.

1. INTRODUCCIÓN

A medida que avanzan los tiempos en el mundo de la conservación y restauración de patrimonio se ha descubierto que la iluminación tiene una gran influencia sobre las obras de arte, en mayor o menor medida dependiendo del material sobre las cuales están proyectadas, pero en cualquier caso una influencia notable la cual se ha de controlar.

Según su respuesta a la iluminación las obras de arte se pueden clasificar en tres grupos:

- Piezas con baja sensibilidad a la luz tales como porcelanas, vidrios, mármoles, etc., pueden ser iluminados hasta 300 lux.
- Piezas con sensibilidad media a la luz como pinturas al óleo y acrílicas pueden ser iluminadas con un máximo de 200 lux.
- Piezas con alta sensibilidad a la luz como acuarelas, tejidos materiales teñidos o pergaminos deben ser iluminados como máximo con 50 lux.

En las obras con sensibilidad media y alta se debe evitar iluminarlas con fuentes que tengan componentes ultravioletas o infrarrojas, ya que la radiación ultravioleta afecta químicamente a los colores y los tejidos sobre los cuales la obra se representa. En cuanto a la radiación infrarroja ésta eleva la temperatura de la obra provocando desgaste (ICOM, 2010).

El objetivo de este trabajo es desarrollar un prototipo de sensor fiable con el cual medir estos parámetros y poder hacer un buen trabajo de iluminación en recintos donde se expongan obras de arte y patrimonio.

La problemática a la que se enfrenta este trabajo es a la sustitución de las antiguas lámparas incandescentes por las actuales lámparas LED, las cuales tienen un comportamiento diferente en cuanto a su espectro de radiación. El espectro de radiación solar tiene la siguiente forma:

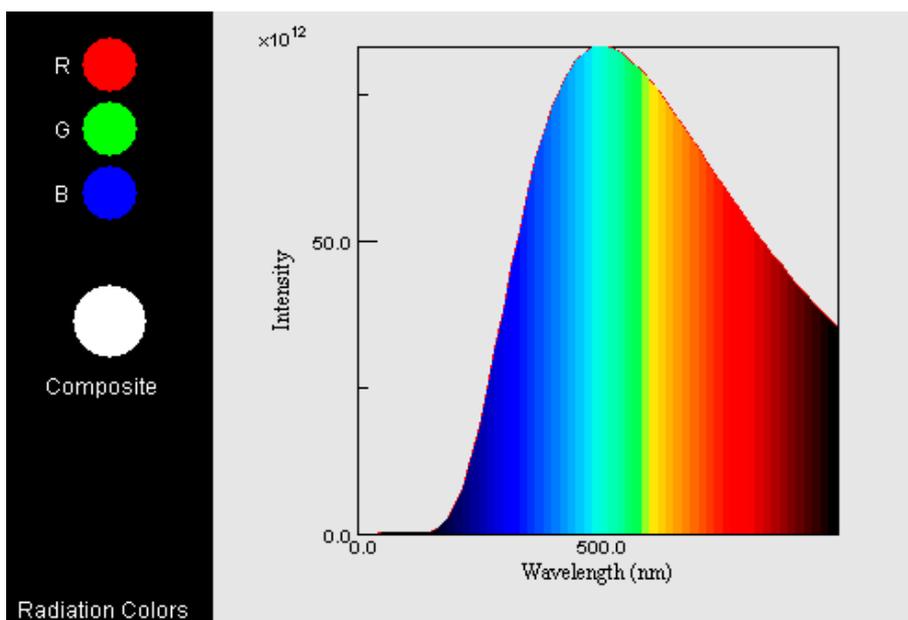


Fig.1 Espectro solar (Carlos, 2007)

Una vez tenemos la referencia solar la compararemos con el espectro de una luz incandescente y una luz LED. En el caso de la incandescente es el siguiente:

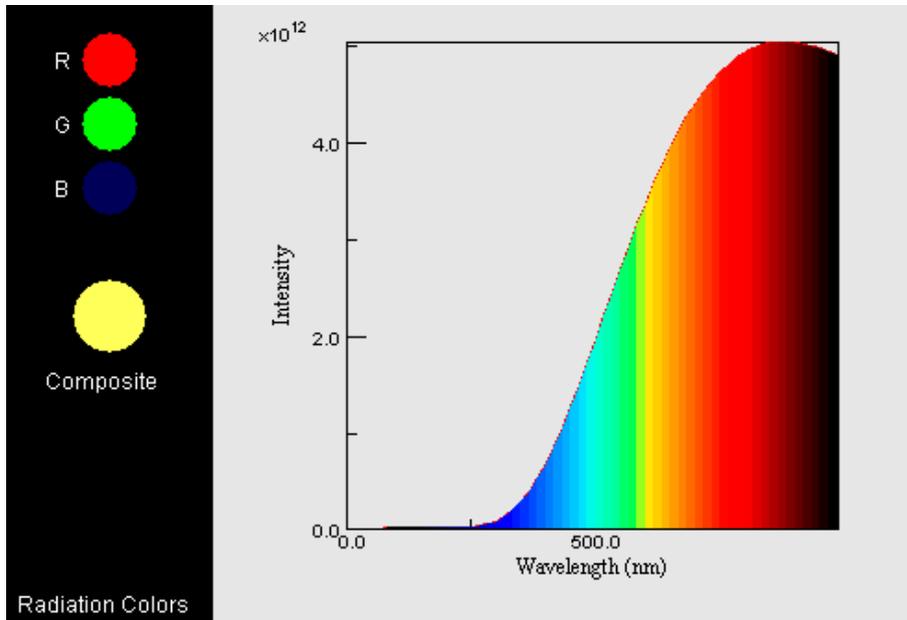


Fig.2 Espectro bombilla incandescente (Carlos, 2007)

En comparación con el espectro solar, la radiación de una bombilla incandescente tiene más intensidad en longitudes de onda próximas al infrarrojo, por lo que al hacer el estudio lumínico de una sala de exposición iluminada con bombillas incandescentes se debería prestar especial atención en las longitudes de ondas próximas al infrarrojo y a las propias longitudes infrarrojas.

Si ahora observamos el espectro obtenido con una bombilla basada en tecnología LED:

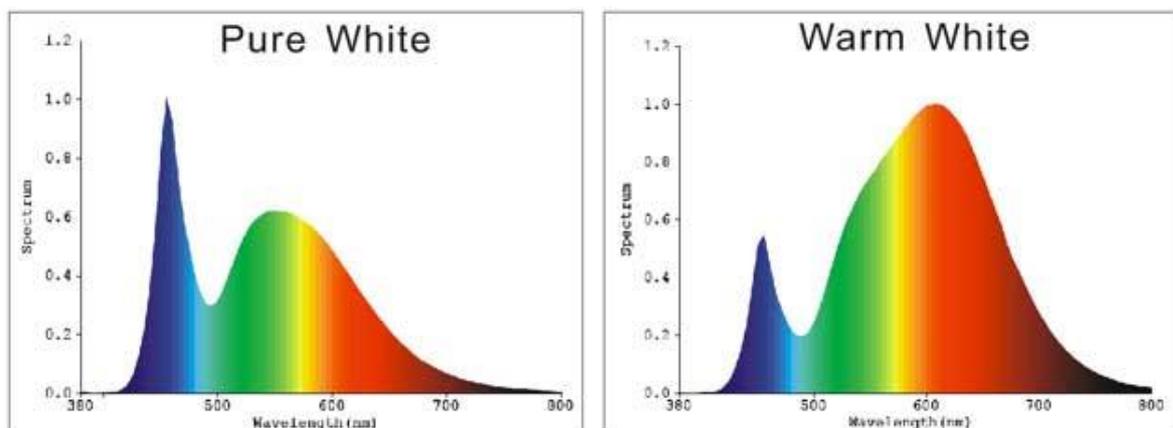


Fig.3 Espectro bombilla led (jeffkrol, 2013)

Se observa a primera vista que el espectro de una bombilla LED es muy diferente al de una bombilla incandescente, además también se aprecia una diferencia de espectros entre distintas lámparas LED. Para conseguir distintos tonos de blanco la combinación de diodos LED es diferente, por lo tanto cada una de estas combinaciones emitirá un espectro diferente el cual tendrá picos de intensidad a diferentes longitudes de onda.

En la actualidad existen dos métodos para fabricar una bombilla que emita luz blanca mediante tecnología led. El primero es el método LED RGB (red, green, blue) que consiste en hacer una combinación de tres ledes (rojo, verde y azul) y regular la intensidad de cada uno de ellos para generar una luz blanca. La ventaja de este método es que se pueden obtener diferentes tipos de blanco dependiendo de cómo se regule la intensidad de cada uno de los ledes mientras que como principal desventaja tiene el alto coste de fabricación. El segundo método es el led de fósforo convertido (pcLED) en el cuál se utiliza un led de longitud de onda corta, normalmente el azul, recubierto de fósforo. El fósforo absorbe una gran porción de luz azul y el led emite una luz blanca. Este método es el más usado debido a sus bajos costes de producción y como desventaja tienen la invariabilidad del tono de luz fácilmente como los ledes RGB.

Además de luz blanca con los ledes RGB los autores de las obras pueden elegir el color de la iluminación. Algunos autores contemporáneos prefieren iluminar sus obras con tonos fríos o cálidos para que su obra tenga más expresión.

Como los tres colores principales de iluminación led son el rojo, verde y azul se ha optado por un sensor de luz que registre la intensidad lumínica en las longitudes de onda en las que estos colores son emitidos.

Para entender el trabajo realizado en este proyecto hay que tener clara la diferencia entre unidades radiométricas y unidades fotométricas. Las unidades radiométricas son las que expresan la transferencia de energía desde un emisor a un receptor. Estas unidades asumen la teoría de la conservación de la energía a lo largo de un tubo de rayos, despreciando los fenómenos de interferencia y/o difracción. Cuando se tiene en cuenta la sensibilidad del ojo humano a las distintas longitudes de onda estas unidades se convierten en unidades fotométricas. Es decir las unidades fotométricas sólo se miden en la región visible del espectro, con longitudes de onda entre 360-760 nm aproximadamente (González, 2006).

Cuando hablamos de unidades fotométricas se ha de diferenciar entre respuesta fotópica y respuesta escotópica. En la retina del ojo humano existen dos tipos de células, las células tipo cono encargadas de la visión fotópica y las células tipo bastones encargadas de la visión escotópica. La visión fotópica se desarrolla en condiciones de alta luminosidad teniendo la respuesta máxima en la longitud de onda de 550 nm mientras que la visión escotópica se desarrolla en condiciones de baja luminosidad teniendo su respuesta máxima en 505 nm de longitud de onda. Si no se especifica lo contrario las unidades fotométricas están expresadas respecto a la visión fotópica (Roger Folch, Riera Guasp, & Roldán Porta, 2010).

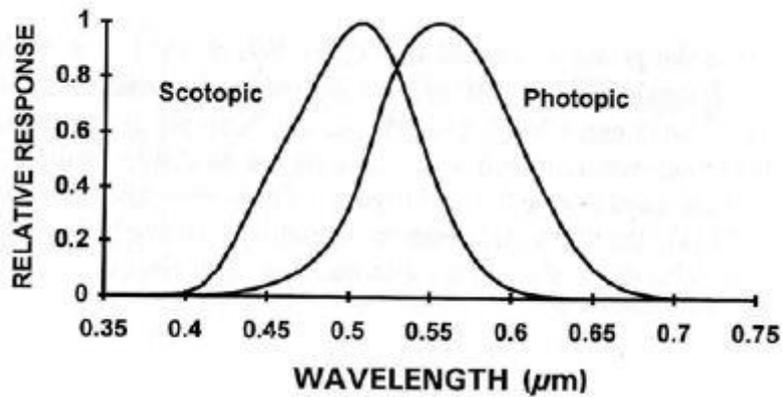


Fig.4 Respuesta visión fotópica y escotópica (González, 2006)

Este trabajo se centrará en dos unidades, una radiométrica y su equivalente fotométrica. Estas unidades son la irradiancia y la iluminancia.

La irradiancia es la densidad de flujo radiante por unidad de superficie que incide sobre un punto. Se mide en W/m^2 y es la unidad que va a registrar el sensor.

La iluminancia por su parte es el flujo luminoso por unidad de superficie que percibe el ojo humano. Se mide en lx (léase lux), la unidad de flujo luminoso es el lm (léase lumen) por lo tanto el $lx=lm/m^2$, es decir, la irradiancia y la iluminancia se relacionan mediante la eficacia luminosa de radiación que relaciona el flujo luminoso en lúmenes y el flujo radiante en vatios, por lo que si dividimos cada término por unidad de superficie obtendremos la relación iluminancia en luxes e irradiancia en vatios por metro cuadrado.

El motivo por el cual estas dos unidades han sido utilizadas es que el sensor utilizado mide la unidad radiométrica mientras que el ICOM (International Council Of Museums) recomienda los límites de iluminación expuestos anteriormente en la unidad fotométrica.

2. ANTECEDENTES, MOTIVACIÓN Y JUSTIFICACIÓN

2.1 Equipamiento disponible en el mercado

Para este apartado se ha estudiado el catálogo de referencia dentro de la industria fotométrica (Newport, 2011). Actualmente en el mercado se dispone de una gran variedad de dispositivos para medición de luz, para magnitudes tanto radiométricas como fotométricas, desde máquinas de sobre mesa hasta dispositivos portátiles fáciles de transportar.

Sin embargo en el referenciado catálogo se ha encontrado un posible nicho de mercado dentro del ámbito de la conservación de obras de arte. No se ha encontrado ningún dispositivo capaz de direccionar automáticamente el sensor para captar la intensidad de la luz recibida desde diferentes ángulos. Esta característica sería de mucha utilidad ya que se podría detectar con mayor precisión el pico de intensidad lumínica que recibe la obra de arte y desde que ángulo la recibe, facilitando el estudio lumínico.

2.2 Motivación y justificación

La motivación y justificación de este proyecto es la comprobación de la capacidad del alumno de GITI para ejecutar un proyecto de carácter científico-técnico utilizando las habilidades adquiridas a lo largo de sus estudios. Además de esto se ha querido impulsar la inserción de las lámparas led que son más eficientes y ecológicas que el resto de sistemas de iluminación, para ello debemos estudiarla mejor y buscar nuevos sensores especializados en este tipo de iluminación.

3. OBJETIVOS

Como se ha mencionado anteriormente este trabajo pretende ayudar a la conservación de las obras de arte con la construcción de un prototipo para medir la iluminación, especialmente iluminación led, potenciando el uso de esta nueva tecnología más eficiente y sensible con el medio ambiente.

Para la ejecución del trabajo se han marcado los siguientes objetivos específicos:

- Construir físicamente el prototipo que permita tanto la medición como la dirección del mismo.
- Programar la medición del sensor a las distintas longitudes de onda que más adelante serán explicadas.
- Programar los motores paso a paso para que ajusten la dirección del sensor correctamente.
- Interconectar ambos programas en un único programa mediante Arduino.
- Trasladar los datos desde Arduino a LabVIEW y exportarlos a un fichero Excel para hacerlos más manejables.

4. MATERIALES Y MÉTODOS

4.1 Software

4.1.1 Lenguaje de programación: Arduino programming language

Hoy en día hay muchos lenguajes de programación que pueden ser utilizados para leer las instrucciones indicadas por el usuario y traducirlas para que el ordenador pueda ejecutarlas. En este caso se ha decidido utilizar el lenguaje de programación propio de Arduino, ya que el microcontrolador utilizado ha sido también un Arduino.



Fig.5 Diferentes lenguajes de programación (campusMVP, 2014)

Arduino programming language es el lenguaje propio de Arduino basado en los lenguaje C y soporta todas sus funciones estándar. En la propia página web de Arduino se puede encontrar una referencia acerca del lenguaje con las funciones y características básicas.

La programación en Arduino se estructura en dos bloques: el setup y el loop. Dentro de la función setup se iniciará el programa, se declarará la función de los pines a utilizar (entrada o salida) y se inicializarán las variables a utilizar. Esta función se ejecutará una sola vez al inicio del programa. La función loop se ejecuta en bucle infinitamente, esta función es el programa en sí. Antes de estas dos funciones se declaran las variables a utilizar y se pueden crear funciones nuevas para utilizarlas posteriormente en la función loop.

Además de lo mencionado en el párrafo anterior también se pueden incluir librerías. Las librerías son archivos que incluyen funciones para ser utilizadas durante la programación. Cuando descargas el entorno Arduino hay varias librerías incluidas pero se pueden incluir otras librerías creadas por usuarios en internet o crear tu propia librería. Las librerías juegan un papel muy importante en el entorno Arduino, así hay librerías para manejar distintos periféricos, como teclados, ratones, motores paso a paso etc. Usuarios de todo el mundo crean sus librerías para cubrir sus necesidades y las comparten mediante internet, de modo que otros usuarios puedan beneficiarse de ellas e incluso mejorarlas.

4.1.2 Lenguaje de programación: LabVIEW

LabVIEW es un software para automatización y control de procesos. Con este software se puede programar de una forma intuitiva, ya que no se programa escribiendo código, como en la mayoría de entornos de programación.

En LabVIEW se programa gráficamente, existen dos ventanas, la ventana del diagrama de bloques y la ventana del panel frontal. El diagrama de bloques es donde está la programación deseada mientras que el panel frontal representa lo que el usuario ve cuando se ejecuta el programa. De este modo se puede crear fácilmente un sistema SCADA, acrónimo de Supervisory Control And Data Acquisition (Supervisión, Control y Adquisición de Datos). El sistema SCADA permite manejar el programa de una manera visual y atractiva, con una interfaz gráfica dinámica en la que queda claro el funcionamiento.

Dentro del diagrama de bloques, como su propio nombre indica se programa por bloques. Por ejemplo un bucle se construye como un cuadrado, lo que haya dentro de ese cuadrado se repetirá en bucle un número de veces dependiendo del tipo de bucle y las condiciones del mismo.

Además de las herramientas de programación dentro del diagrama de bloques también se incluyen los actuadores o indicadores que aparecen en el panel frontal. Para poner un ejemplo sencillo, se podría hacer un programa que midiera una tensión variable con un potenciómetro externo. En el diagrama de bloques habría un actuador que podría ser un botón y un indicador que podría ser una gráfica. De este modo en el panel frontal tendríamos un sistema SCADA donde pulsáramos un botón con ratón para empezar a medir y tendríamos una gráfica que nos mostrará la tensión leída.

4.2 HARDWARE

4.2.1 Microcontrolador: Arduino

Cuando se ha hecho referencia a un microcontrolador en el anterior apartado se refería a un dispositivo capaz de leer y ejecutar unas instrucciones previamente programadas. Para este cometido se necesita un microprocesador que interprete y ejecute las instrucciones, una memoria donde almacenar las instrucciones y una serie de puertos de entrada/salida, tanto digitales como analógicos para poder comunicarse con distintos periféricos. Estos tres elementos van integrados en un circuito integrado o chip.

El microcontrolador elegido ha sido el ATmega328 (Atmel Corporation, San José, Estados Unidos), que cumple con los requisitos del trabajo a realizar. Este microcontrolador cuenta con una memoria para almacenar el código de 32 kB y una memoria para el almacenamiento de datos de 1 kB suficientes ambas para nuestro cometido. Además cuenta con suficientes puertos tanto analógicos como digitales.

Para un uso más cómodo del microcontrolador se ha optado por utilizar una placa Arduino Duemilanove, en la que se encuentra el microcontrolador con fácil acceso a los puertos y a la alimentación que proporciona. Esta placa también cuenta con dos puertos de entrada al microcontrolador, un puerto USB con el que se puede establecer la comunicación entre el microcontrolador y el ordenador para cargar el programa y proporcionar una alimentación de 5V y una entrada de conector tipo jack en el caso de tener que alimentar a más de 5V. En este caso se ha debido utilizar esta última entrada ya que los motores paso a paso utilizados necesitaban 12 V para funcionar correctamente.

4.2.2 Sensor TCS3200

Para la medición se ha utilizado el sensor de color TCS3200 de la casa TAOS. Este sensor es un convertidor de luz a frecuencia formado por una matriz de 64 fotodiodos, 16 con filtro de color azul de una longitud de onda de 470 nm, 16 con filtro de color verde con una longitud de onda de 524 nm, 16 con filtro de color rojo con una longitud de onda de 640 nm y por último 16 sin filtro. Este sensor convierte la irradiancia emitida a las tres longitudes de onda en una onda cuadrada con una frecuencia proporcional a la irradiancia. El fabricante proporciona unas tablas para detectar colores. Son embargo en este caso se ha decidido calibrar manualmente una fuente lumínica para poder obtener lecturas de irradiancia para posteriormente obtener los valores equivalentes de iluminancia.

4.2.3 Motor paso a paso

La cantidad de luz que incide en una superficie es diferente dependiendo del ángulo de incidencia, es por eso que se ha decidido dotar al prototipo de dos motores paso a paso para poder direccionarlo en 360°.

Un motor paso a paso es un dispositivo electromecánico en el que un eje gira mediante impulsos eléctricos que excitan unas bobinas. Existen de varios tipos pero en este caso se ha optado por utilizar motores bipolares en los que existen dos bobinas y por lo tanto 4 cables, dos para cada bobina, que excitados correctamente hacen que el motor de un paso o un medio paso.

Para facilitar el manejo de estos motores se ha optado por utilizar el controlador DRV8825 denominado POLOLU comercialmente, el cual tiene como entrada los cuatro cables y proporciona como salida para manejar el motor dos cables, uno con el que se controla el paso y otro con el que se controla la dirección. De este modo con dos puertos digitales se puede controlar cada motor y se simplifica el número de puertos a utilizar. Este controlador ofrece la posibilidad de dividir los pasos, es decir aumentar la precisión del motor. El POLOLU tiene 3 pines de modo específico para elegir el número de micro pasos dados por cada paso entero del motor:

Table 1. Stepping Format

| MODE2 | MODE1 | MODE0 | STEP MODE |
|-------|-------|-------|-------------------------------------------------|
| 0 | 0 | 0 | Full step (2-phase excitation) with 71% current |
| 0 | 0 | 1 | 1/2 step (1-2 phase excitation) |
| 0 | 1 | 0 | 1/4 step (W1-2 phase excitation) |
| 0 | 1 | 1 | 8 microsteps/step |
| 1 | 0 | 0 | 16 microsteps/step |
| 1 | 0 | 1 | 32 microsteps/step |
| 1 | 1 | 0 | 32 microsteps/step |
| 1 | 1 | 1 | 32 microsteps/step |

Fig.6 Tabla para la selección del formato de pasos en el controlador PLOLU DRV8825 (Pololu, 2014)

Para el trabajo se ha seleccionado la opción de mayor precisión, es decir 32 micro pasos por cada paso entero del motor que se escoja. Para esta precisión se pueden usar tres combinaciones en los pines de modo, como se observa en la tabla de arriba, en este caso se ha optado por poner los 3 pines a 1 ya que se consideró la opción más fácil cuando se diseñó la PCB.

Para los motores se ha elegido el motor 17HS3430, el cual tiene un paso angular de 1.8°, es decir, cada paso recorre 1.8° y por lo tanto da una vuelta cada 200 pasos. Además al seleccionar la opción de dividir cada paso entero en 32 micro pasos para dar una vuelta completa el motor dará $200 \times 32 = 704$ micro pasos por vuelta.

4.2.4 PCB

Para la conexión de los distintos elementos del prototipo se ha diseñado e implementado una PCB. Esta placa consiste en distintos circuitos conductores que unen cada elemento y proporcionan un aspecto compacto al prototipo. Para diseñar la placa se ha utilizado el software libre Eagle.

4.2.5 Interruptor de final de carrera

Para establecer el origen a partir del cual los motores inician su recorrido se han utilizado finales de carrera. Estos dispositivos tienen una pestaña que está normalmente abierta, cuando el motor llega a la posición deseada, la estructura hace que esta pestaña se cierre pulsando un botón y cambiando el interruptor del final de carrera. Cuando el interruptor cambia el motor se para y ya tiene establecido su origen.



Fig.7 Interruptor final de carrera (cetronic, s.f.)

4.2.6 Estructura

La estructura para poder direccionar el prototipo se ha hecho en madera, utilizándose tornillos, cola blanca y pegamento termofusible para el ensamblaje y sujeción de todos los elementos.

En esta estructura se unen los motores, la PCB y los finales de carrera para obtener el prototipo final.

4.3 Calibración

Para la calibración del prototipo se optó por calibrar una fuente lumínica. Esta fuente lumínica consiste en una bombilla incandescente de 100W de potencia y un regulador de intensidad, midiendo la intensidad que circulaba por la bombilla con un multímetro. La calibración de la fuente se hizo en el Servicio de Conservación y Restauración de Bienes Culturales de la Diputación de Castellón con un espectrofotómetro, concretamente el modelo i1 basic pro de la casa x-rite.

Esta calibración se realizó para el primer prototipo explicado al principio del trabajo. Aunque no se pudo continuar con aquel prototipo la calibración ha sido un punto fundamental en la ejecución del trabajo final. Es por ello que se remarca una vez más la importancia del trabajo previo realizado.



Fig.8 Montaje para medir la intensidad de la fuente lumínica a la izquierda (Fig.8.a) y espectrofotómetro utilizado para calibrar la fuente a la derecha (Fig.8.b) (X-Rite, s.f.)

5. DESARROLLO Y RESULTADOS

En esta sección se procede a explicar el desarrollo del prototipo, tanto a nivel de hardware como software, así como los resultados obtenidos tras haber calibrado y puesto en funcionamiento el prototipo.

En primer lugar se explicará las características del prototipo en cuanto al hardware, los distintos elementos que lo componen, la relación entre ellos y los diversos métodos utilizados para su correcta conexión.

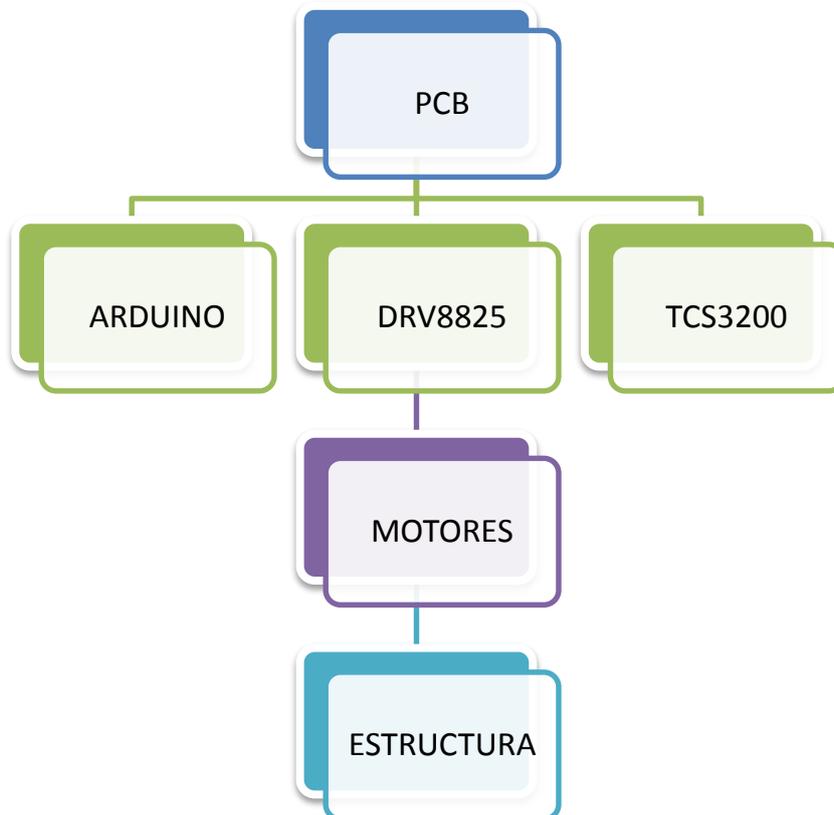
En segundo lugar se detallará el proceso de calibración del prototipo, el material utilizado, el código implementado para la calibración y los resultados de la calibración.

En tercer lugar se explicará el código construido para el prototipo, se enumeraran las principales funciones y se detallará su funcionamiento.

Por último se expondrán los resultados que se obtienen al ejecutar el programa que contiene el prototipo.

5.1 Hardware

La conexión del hardware está dispuesta según el siguiente organigrama:



Como se muestra en el organigrama en la PCB están conectados la placa Arduino, el sensor TCS3200 para captar la luz y los dos controladores para manejar los motores permitiendo la comunicación entre ellos. Además en la PCB también se encuentran las entradas de los motores conectadas cada uno a su controlador. Finalmente los motores se encuentran sobre una estructura, en la que están integrados los finales de carrera, que permite el correcto funcionamiento del prototipo.

Para el diseño de la PCB se ha utilizado el software libre Eagle, el cual permite diseñar una PCB de un tamaño 10 X 8 cm, suficiente para este caso. Este software te permite crear un esquemático, un diagrama donde seleccionas los elementos a utilizar en librerías y conectarlos mediante hilos. A partir de este esquemático se puede pasar a la pestaña board, donde aparecen todos los elementos con sus encapsulados y dimensiones reales. En esta pestaña se distribuyen los distintos elementos y pistas para que no se produzcan solapes entre las pistas. Los elementos utilizados en la PCB como se ha dicho anteriormente son: la placa Arduino, los controladores de los motores y el sensor TCS3200. Para los dos primeros elementos existían librerías en internet donde estaban creados ya que son elementos de uso extendido, pero para el último de ellos se ha tenido que crear una librería especial.

La creación de una librería se hace en el mismo programa, pudiendo crear tantos componentes como se quiera dentro de la misma librería. A la hora de crear un componente para una librería se debe crear tres elementos: symbol, package y device. El elemento symbol es el que aparecerá en el esquemático y debe ser sencillo y de fácil acceso a los pines del componente, el elemento package debe tener las dimensiones del componente real y aparecerá en la pestaña board para hacer la distribución de la PCB. Por último el elemento device es la unión entre el symbol y el package, es decir, la conexión entre los pines que aparecen en el symbol del esquemático y los pines del package que aparecerán en la pestaña board.

La librería creada se ha denominado lightsensor.lbr ya que ha sido creada para poder incluir el sensor de luz TCS3200. En ella se ha decidido crear dos componentes, el propio sensor y un módulo para incluirlo en la PCB. La creación del módulo es debida a las dimensiones del sensor, demasiado pequeñas para incluirlo directamente en la PCB, así con la creación de un módulo donde el TCS3200 va soldado y éste, de unas dimensiones más manejables, acoplado en la PCB, se consigue una mayor comodidad a la hora de ensamblar los componentes en la PCB.

Una vez creados tanto el esquemático como la distribución de la PCB en la pestaña board se ha procedido a la fabricación física de la PCB. Para ello en la pestaña board se dejan habilitadas únicamente las capas top, pads y dimensión creando de esta forma una máscara. En la capa top se encuentran las pistas, en la capa pads los agujeros necesarios para ensamblar los componentes y la capa dimensión es simplemente un recuadro con las dimensiones de la placa. Estando habilitadas estas capas se imprime en una hoja transparente. Una vez realizado este paso se procede a la insolación del material fotosensible. Este material tiene una placa de cobre en cada cara sobre la cual se extiende otra capa fotosensible. Al superponer la máscara con el material fotosensible e insolarlo toda la capa fotosensible se destruye menos las pistas y los agujeros necesarios para nuestra PCB, ya que están impresos en el papel transparente y no dejan que la luz del sol incida sobre la placa.

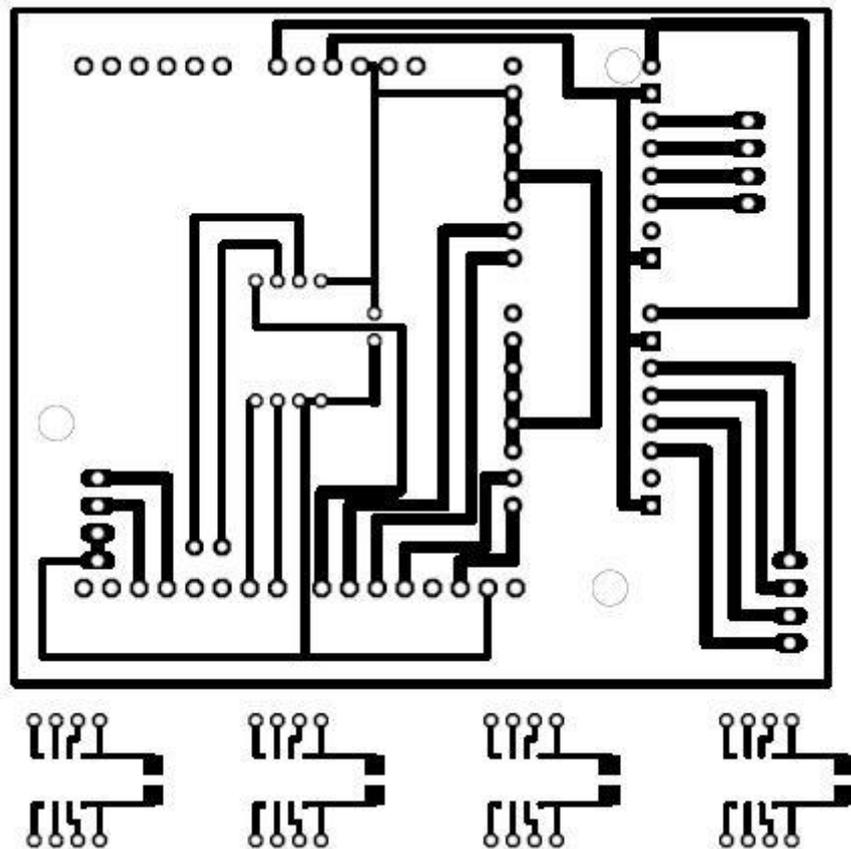


Fig. Máscara utilizada para la creación de la PCB final.

Tras el proceso de insolación se introduce la placa en ácido para eliminar todo el cobre que ha quedado al descubierto tras la insolación. Más tarde se retira el material fotosensible dejando al descubierto las pistas de cobre necesarias en nuestra PCB. Finalmente se perforan los agujeros y se sueldan los distintos componentes quedando conectados mediante las pistas.

Una vez acabado el proceso ensamblando todos los componentes la PCB queda del siguiente modo:

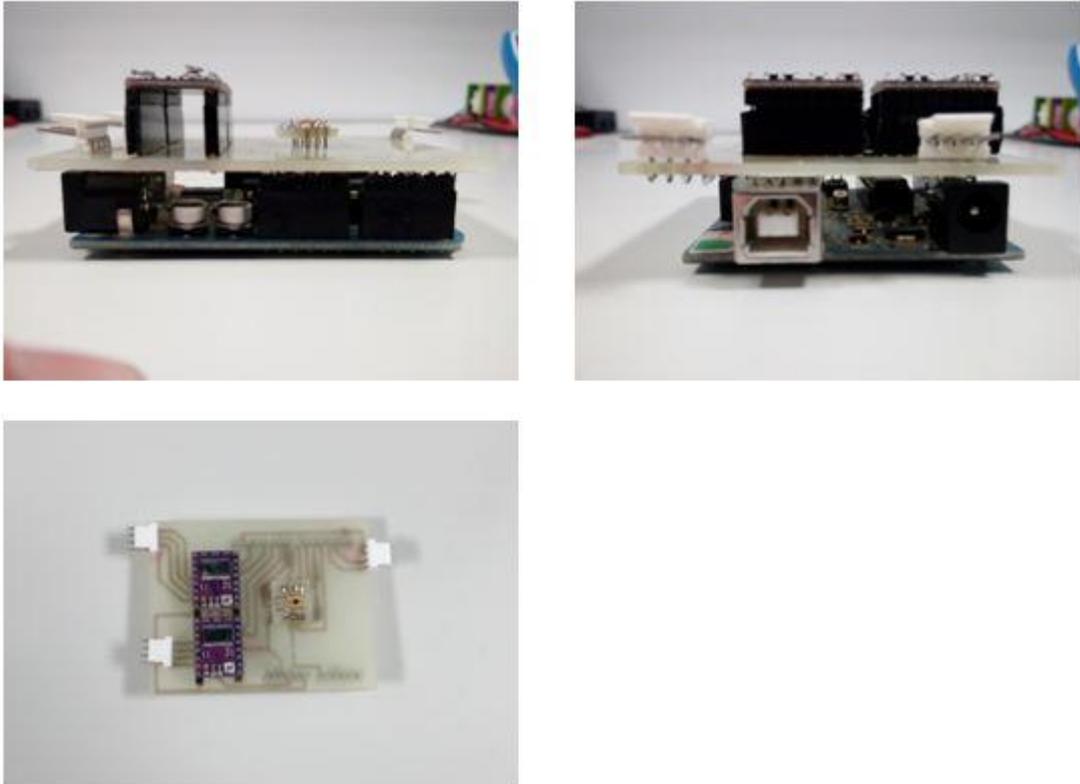


Fig.9 Vistas de la PCB acabada con todos los componentes ensamblados incluida la placa Arduino Duemilanove. Perfil derecho en la parte superior derecha (Fig.9.a), alzado en la parte superior izquierda (Fig.9.b) y planta en la parte inferior izquierda (Fig.9.c)

Con la PCB ya montada se ha realizado la estructura sobre la que ésta iba montada. La estructura consiste en dos maderas ensambladas en forma de “L”. Sobre una de estas maderas va montado el motor que se encarga de girar el prototipo verticalmente. La otra madera se ha perforado para poder introducir el eje del otro motor para realizar el giro en planta. Una vez hecho esto se ha perforado otro trozo de madera donde va montada la PCB, en este agujero va el eje del primer motor. Sobre la madera en la que va el eje del segundo motor se ha instalado el final de carrera correspondiente al giro en alzado y se ha provisto de una pequeña madera debajo de la PCB para que lo accione. Por otra parte el motor que controla el movimiento en planta reposa sobre una madera que se apoya en el suelo mediante dos listones para estabilizar todo el prototipo. En esta madera se sitúa el segundo final de carrera que se acciona con una varilla que va unida a la madera superior. Por último se ha envuelto el sensor con un tubo negro para que le llegue únicamente la luz desde donde está orientado. Una vez finalizado todo el montaje la estructura ha quedado de la siguiente manera:

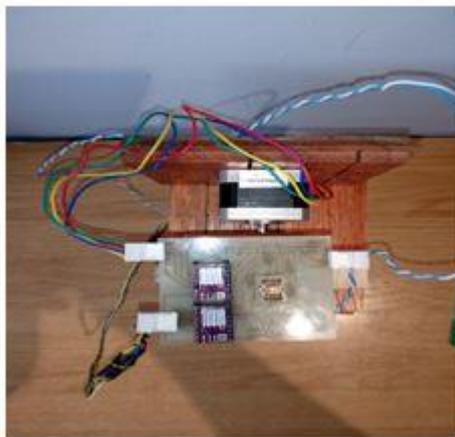
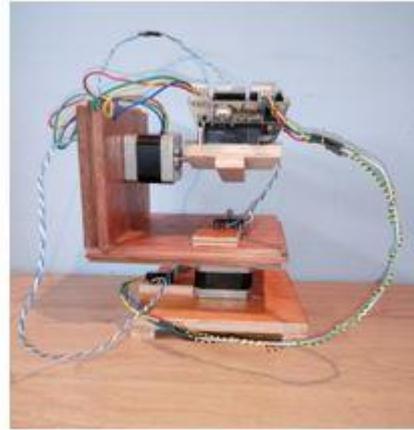
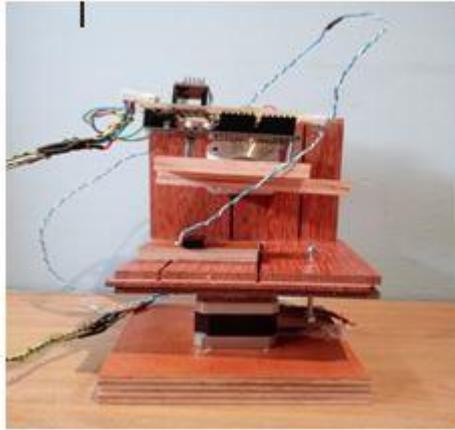


Fig.10 Estructura final. Vista en alzado en la parte superior izquierda (Fig.10.a), vista en planta en la parte inferior izquierda (Fig.10.b) y vista de perfil izquierdo en la parte superior derecha (Fig.10.c).

5.2 Calibración

La calibración del prototipo ha tenido dos fases: la primera fase fue la calibración de una fuente lumínica y la segunda fase fue una vez calibrada dicha fuente la calibración del prototipo con la ayuda de ésta.

La primera fase se realizó en el Servicio de Conservación y Restauración de Bienes Culturales de la Diputación de Castellón con la ayuda de un espectrómetro. La fuente lumínica a calibrar consiste en un portalámparas con una bombilla incandescente de 100 W, un regulador de intensidad puesto en serie y tras él se ha hecho un puente para insertar un multímetro en serie para medir la intensidad.

Una vez montada la fuente lumínica la calibración de la misma ha consistido en subir poco a poco la intensidad de la fuente tomando datos con el espectrómetro en diferentes rangos de intensidad. El espectrómetro mide la irradiancia espectral de la fuente en $\text{mW}/\text{m}^2\text{nm}$. La magnitud que nos interesa es la irradiancia, no la irradiancia espectral, ésta se consigue multiplicando la irradiancia espectral por la longitud de onda correspondiente. El espectrómetro utilizado para la calibración mide irradiancia espectral desde 380 nm hasta 730 nm en intervalos de 10 nm. En el caso del prototipo el sensor utilizado mide irradiancia en tres longitudes de onda: 470 nm, 524 nm y 640 nm. Así que se ha optado por utilizar los valores medidos por el espectrómetro a 470 nm, 520 nm y 640 nm por ser los que más se acercan a la medida real pudiendo despreciar el error cometido. Una vez realizada la evaluación las rectas de calibración obtenidas son las siguientes:

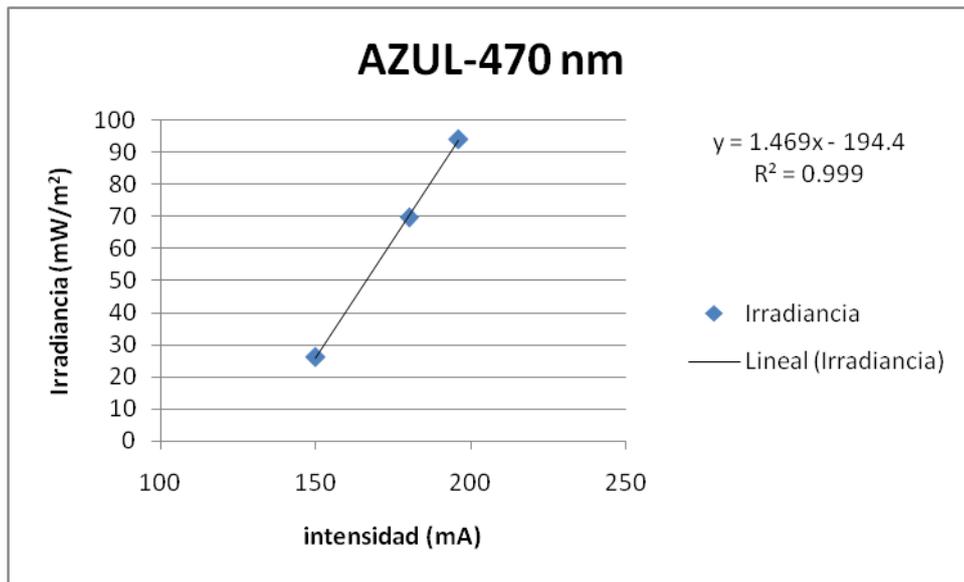


Fig.11 Calibrado de la fuente a 470 nm de longitud de onda.

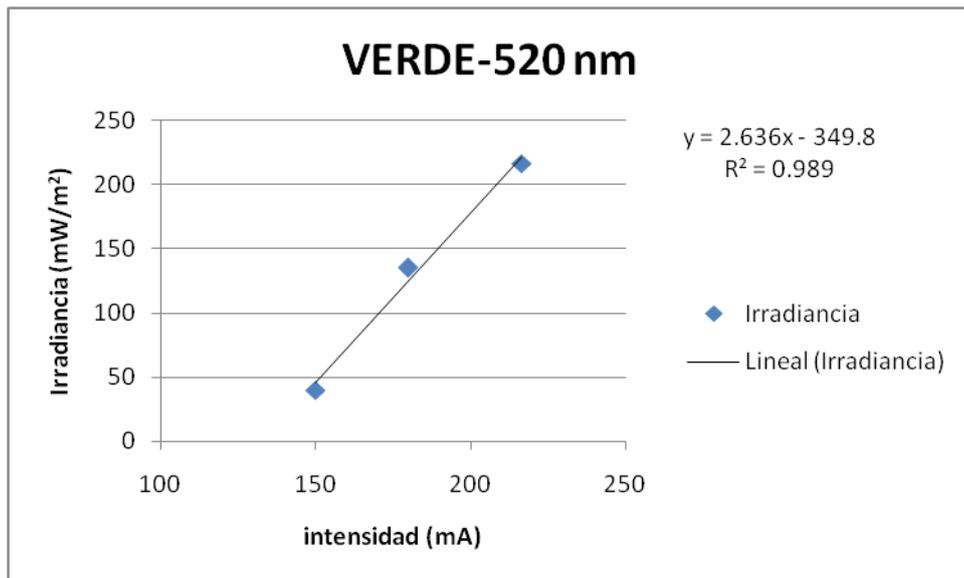


Fig.12 Calibrado de la fuente a 520 nm de longitud de onda.

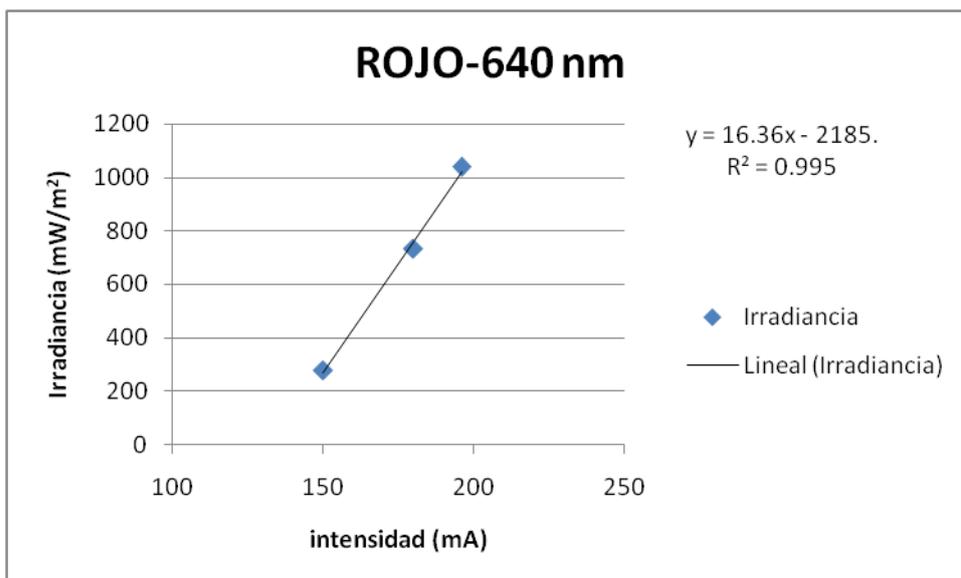


Fig.13 Calibrado de la fuente a 640 nm de longitud de onda.

Una vez realizada la calibración de la fuente lumínica se procede a calibrar el sensor. Como el sensor es un convertidor de intensidad lumínica a frecuencia necesitaremos medir la frecuencia que proporciona éste para calibrarlo.

La calibración del sensor se ha llevado a cabo en el departamento de física aplicada de la ETSII, intentando imitar las mismas condiciones de contorno en las que se calibró la fuente lumínica. Se han implementado tres programas sencillos, uno para cada longitud de onda en la que puede medir el sensor. Estos programas seleccionan la longitud de onda en la que el sensor mide la intensidad lumínica y toma datos ininterrumpidamente. Con la ayuda del regulador de intensidad de la fuente se ha seleccionado los distintos valores en los que la irradiancia es conocida gracias a la previa calibración de la fuente. Para cada valor de intensidad y con cada programa se han tomado datos durante 10 segundos. Posteriormente estos datos han sido promediados dando como resultado la frecuencia aceptada como la que el convertidor proporciona cuando la fuente funciona con cada intensidad.

Tras el proceso se obtienen relaciones entre intensidad de la fuente y frecuencia, como conocemos la irradiancia de la fuente a cada intensidad obtenemos la relación entre frecuencia e irradiancia. Se recuerda que la magnitud buscada es la iluminancia, no la irradiancia, por lo que se debe convertir la irradiancia a iluminancia según la siguiente tabla:

| λ (nm) | 400 | 420 | 440 | 460 | 480 | 500 | 520 | 540 | 560 | 580 | 600 | 620 | 640 | 660 | 680 | 700 | 720 |
|----------------|-----|-----|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|------|------|-----|-----|
| lm/W | 0.3 | 2.7 | 15.7 | 41.0 | 95.0 | 221 | 485 | 652 | 680 | 594 | 425 | 260 | 120 | 41.7 | 11.6 | 2.8 | 0.7 |

Fig.14 Tabla de eficacia luminosa de radiación (González, 2006)

Como se muestra en la tabla la conversión varía significativamente dependiendo de la longitud de onda. En la tabla se muestra el ratio lm/W , si recordamos la irradiancia ha sido medida en mW/m^2 y la iluminancia se mide en $\text{lx}=\text{lm/m}^2$ por lo que podemos obtener fácilmente la equivalencia entre los valores registrados de irradiancia y los valores de iluminancia correspondientes a éstos. Una vez finalizado el proceso de calibración los resultados obtenidos se reflejan en las siguientes gráficas:

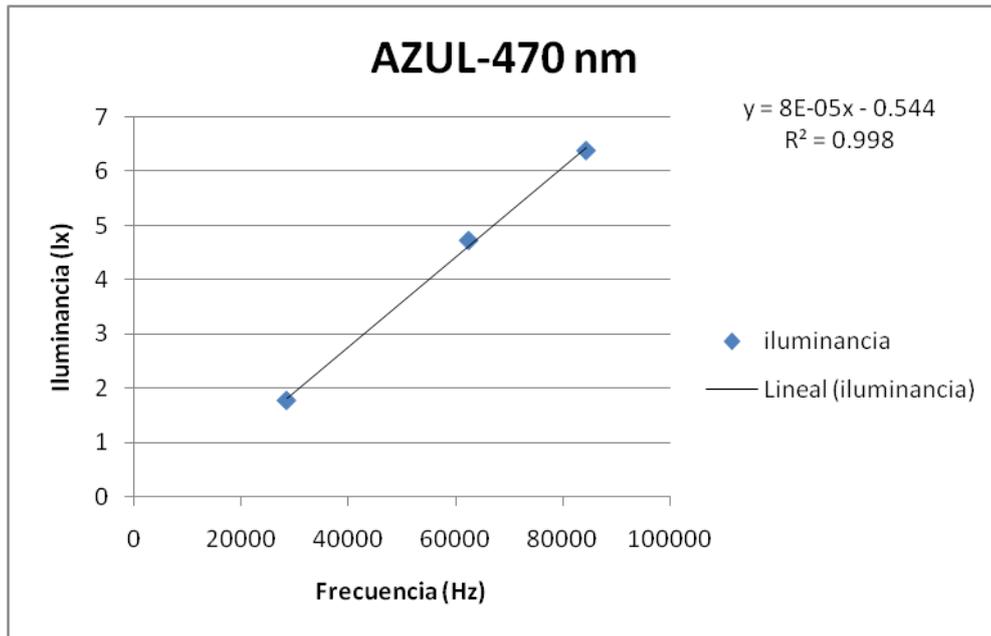


Fig.15 Calibrado del prototipo a 470 nm de longitud de onda

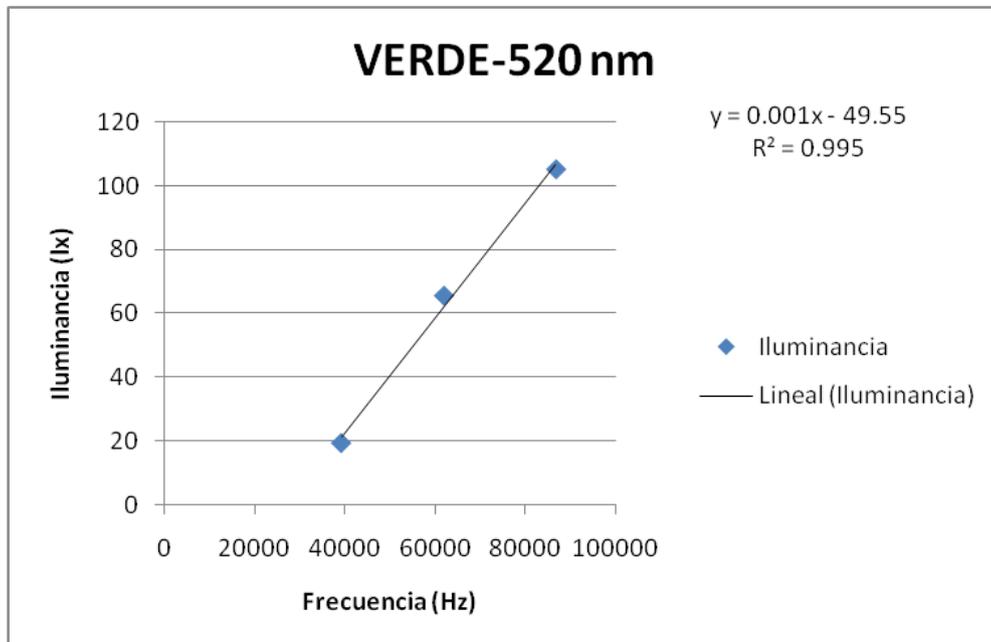


Fig.16 Calibrado del prototipo a 520 nm de longitud de onda

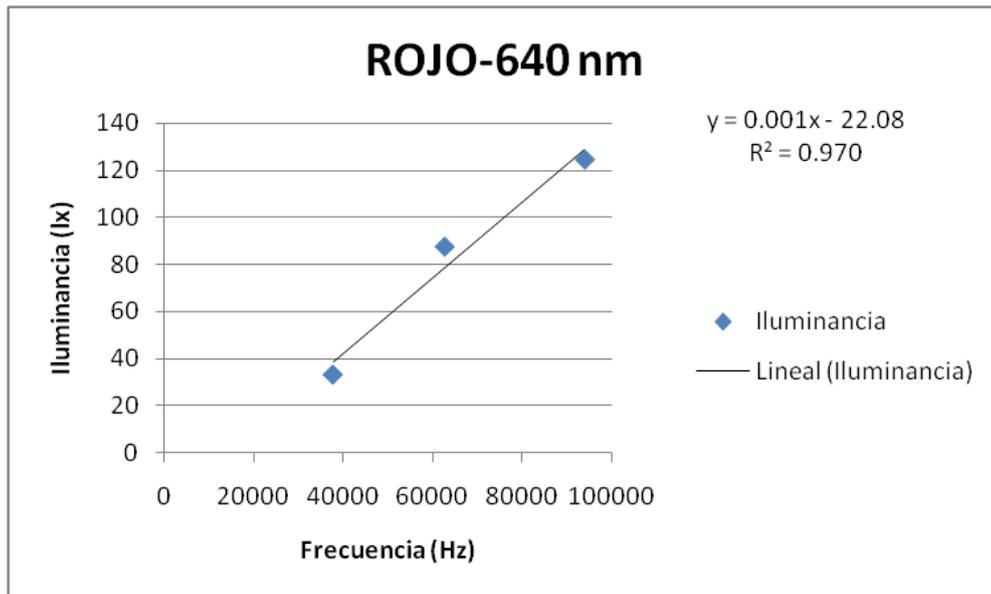


Fig.17 Calibrado del prototipo a 640 nm de longitud de onda

5.3 Software

5.3.1 Arduino

Para la programación en Arduino se han implementado dos funciones diferentes, una para mover los motores y otra para medir la intensidad de la luz.

La función que mueve los motores es la siguiente:

```
void paso(boolean dir,int steps){
  digitalWrite(dirPin,dir);
  delay(50);
  for(int i=0;i<steps;i++){
    digitalWrite(stepperPin, HIGH);
    delayMicroseconds(100);
    digitalWrite(stepperPin, LOW);
    delayMicroseconds(100);
  }
}
```

Fig.18 Función para controlar los motores paso a paso

Como se observa esta función necesita dos parámetros para funcionar, “dir” de tipo binario y “steps” de tipo entero. Lo primero que hace la función es escribir en el pin de dirección del controlador previamente declarado como dirPin la variable “dir”, ésta al ser binaria puede tener únicamente dos valores; “TRUE” o “FALSE”. Dependiendo del valor seleccionado el motor girara en un sentido o en el contrario. A continuación la función ejecuta el bucle for, cada vez que se ejecuta este bucle el motor dará un paso, poniendo en alto y bajo (5V y 0V) el pin del controlador, previamente declarado como stepperPin, que se encarga de los pasos. La función “delayMicroseconds” controla el tiempo de espera entre el cambio de alto a bajo en el pin de pasos en microsegundos, pudiendo cambiar de esta forma la velocidad de giro del motor. Cuanto menos sea el tiempo de espera más alta será la velocidad. Para el otro motor se ha utilizado la misma función añadiéndole a cada parámetro un dos indicando que es para el segundo motor.

Se han implementado tres funciones para la lectura de la intensidad de la luz, cada una en la longitud de onda correspondiente a su color: azul, verde o rojo. A continuación se explica la función que mide en la longitud de onda correspondiente al azul, siendo equivalentes las otras dos cambiando los parámetros necesarios.

```

void medirazul(){
  digitalWrite(S0, HIGH);
  digitalWrite(S1, HIGH);
  digitalWrite(S2, LOW);
  digitalWrite(S3, HIGH);
  while (FreqCount.available()!=true){
    delay(1);}

  unsigned long count = FreqCount.read();//lectura de la frecuencia
  azul=0.00008*count-0.544;// conversión de frecuencia a luxes
  Serial.print(azul);
  Serial.print(";");
}

```

Fig. 19 Función para medir la intensidad lumínica

El sensor utilizado tiene 4 pines de control: S0 y S1 seleccionan la escala en la que se muestra la frecuencia de salida a medir, mientras que S3 y S4 seleccionan el filtro con el que se mide:

Table 1. Selectable Options

| S0 | S1 | OUTPUT FREQUENCY SCALING (f_o) | S2 | S3 | PHOTODIODE TYPE |
|----|----|------------------------------------|----|----|-------------------|
| L | L | Power down | L | L | Red |
| L | H | 2% | L | H | Blue |
| H | L | 20% | H | L | Clear (no filter) |
| H | H | 100% | H | H | Green |

Fig.20 Parámetros para la selección de escala de salida y filtro de los fotodiodos en el sensor TCS3200D (TAOS, s.f.)

Lo primero que hace la función es seleccionar el valor de estos 4 parámetros. En todos los casos se utiliza la frecuencia de salida al 100%, es decir, no se modifica la frecuencia de la onda de salida del sensor. Para cada caso en el que estemos midiendo se seleccionara la combinación de S2 y S3 necesaria según la tabla de arriba, dejando sin utilizar los fotodiodos sin filtro, ya que no son de utilidad en este trabajo.

Una vez escogido la escala de salida y el tipo de fotodiodo a utilizar se procede a medir la frecuencia de salida. Para ello se ha utilizado la librería FreqCount.h, que dispone de las funciones necesarias para la lectura de la frecuencia. Primero el programa comprueba si la lectura de la frecuencia está disponible, si es así la frecuencia se almacena en la variable "count". Una vez almacenada la frecuencia en Hz se traslada a valores de iluminancia (lx) mediante la recta de calibración obtenida en el apartado anterior. Por último se muestra por pantalla el valor de la iluminancia medida y se añade un ";" a continuación. El motivo de este ";" es que se van a exportar los datos para poder leerlo mediante Excel con el formato "texto separado por ";"

Las dos funciones anteriormente explicadas son las principales en las que se basa el programa. Como se ha mencionado anteriormente la programación en Arduino consiste en dos bloques: el "setup" y el "loop".

En el setup a parte de declarar los pines del Arduino como salida o entrada se lleva el sensor a la posición marcada como origen, esta posición será a 0° respecto a la vertical. Esta acción se hace mediante los finales de carrera, moviendo los motores hasta que se activen el final correspondiente a cada uno de ellos.

Una vez ejecutado el setup empieza la función loop la cual se ejecuta en bucle infinitamente. Ésta función es el programa en sí mismo, dentro de ella el sensor es direccionado y tomando medidas a diferentes orientaciones. El recorrido que hace el sensor es el siguiente:

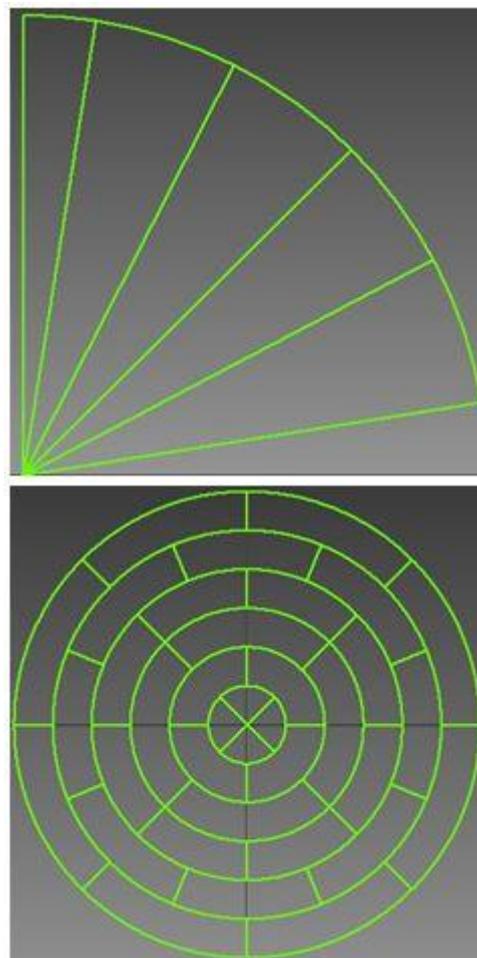


Fig.21 Recorrido del prototipo a la hora de hacer mediciones. Recorrido en alzado en la parte superior (Fig.21.a) y en planta en la parte inferior (Fig.21.b)

Como se observa en la imagen, la posición inicial es a 0° respecto a la vertical, en esta posición el segundo motor gira el sensor en intervalos de 45° y se toman medidas en cada uno de ellos. Este recorrido corresponde a la circunferencia más grande de la vista en planta del recorrido. A continuación el sensor vuelve a la posición inicial de la siguiente trayectoria circular, que como se observa la siguiente trayectoria consiste en los mismos intervalos pero con diferente referencia. Una vez en dicha posición el primer motor cambia la inclinación de la vertical, pasando a ser ésta de 9°. En ese momento el segundo motor vuelve a girar en intervalos de 45°. Éste proceso se repite cambiando la inclinación de la vertical a 27°, 45°, 63° y 81°. A partir de 45° los intervalos de medición del segundo motor se aumentan a 90°, reduciendo el número de medidas a 4 para cada inclinación, tal y como se observa en la fig.18. Para cambiar la orientación y realizar una nueva medida hay que introducir el carácter “M”. En cada medida Arduino escribe por el puerto serie 6 datos, la medida de cada uno de los filtros, la orientación respecto a la vertical en alzado y la posición angular respecta a la planta y una “X”, separado cada uno de estos datos por un “;”.

5.3.2 LabVIEW

En LabVIEW se ha programado la interfaz gráfica con la que el usuario manejará el programa. Cómo se ha comentada en el apartado [4.1.2](#) LabVIEW es un software que permite programar gráficamente a la vez el código y la interfaz gráfica.

En este caso se ha implementado una interfaz muy sencilla en el que se tiene que elegir el puerto serie y el número de medidas.

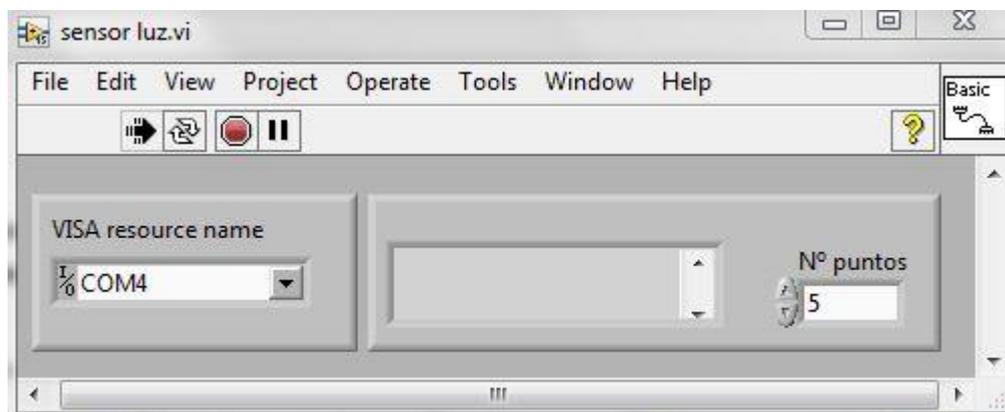


Fig.22 Interfaz gráfica vista por el usuario creada con LabVIEW

El diagrama de bloque correspondiente es también muy sencillo y a continuación se procede a explicarlo. El programa empieza estableciendo la comunicación con Arduino que se explicará en el siguiente apartado. Una vez establecida la conexión con Arduino se entra en un bucle while y a continuación a una estructura TRUE/FALSE. El bucle while es un bucle que se ejecuta infinitamente hasta que ocurra una condición, este bucle se representa con un recuadro, tiene un contador de ciclo que comienza en 0 y un botón rojo de fin de bucle que será activado cuando la condición de fin de bucle se cumpla. Todo lo que haya dentro del recuadro se ejecutará en bucle. La estructura TRUE/FALSE es también un recuadro y dentro de él se ejecutará un proceso u otro dependiendo si la

entrada que se proporciona a este recuadro es TRUE o FALSE. En el caso de que la entrada a esta estructura sea TRUE existe otro bucle while dentro de ella, por el contrario si la entrada es FALSE dentro de la estructura se encuentra un bucle while y una estructura secuencial, que como su nombre indica ejecuta una serie de órdenes consecutivas y dentro de ella un último bucle while.

La entrada a la estructura TRUE/FALSE será TRUE cuando el primer bucle while se encuentre en su primer ciclo, el resto de iteraciones será FALSE. Por lo tanto el primer ciclo se entrará en la estructura TRUE donde hay un bucle while que inicializa la entrada del puerto serie de Arduino y acaba cuando esta entrada ha sido inicializada por completo. Además de este bucle while en la estructura TRUE se escribe la primera línea del archivo en el que se guardarán los datos del sensor. En esta línea se escribe: "Azul;Verde;Rojo;Vertical;Horizontal" que será la cabecera del archivo en la cual debajo de cada columna irán las mediciones con los filtros azul, verde y rojo y las orientaciones del sensor vertical y horizontal. Antes de entrar en el primer bucle while se ha introducido el archivo en blanco en el cual se escriben estos datos. Una vez escritos se pasa a la siguiente línea del archivo.

En el caso de que la estructura sea FALSE, es decir en todos los ciclos del bucle menos en el primero, existe un indicador que pide un número, este será el número de medidas que tome el programa. Tras elegir este número se entra en un nuevo bucle while que durará hasta que se complete el número de mediciones, para esto se le resta 1 la indicador, ya que el contador de ciclos empieza en 0. Una vez dentro de este bucle se ejecuta la estructura secuencial, en el que existen dos etapas. En la primera etapa se introduce el carácter "M" en Arduino para poder realizar la siguiente medida. En la segunda etapa existe un último bucle while que lee los caracteres que envía Arduino por el puerto serie hasta que se encuentra con la "X" que marca el final de la medida. Cada vez que finaliza la medida se escribe en el fichero introducido al principio y se pasa a la siguiente línea del mismo. Por último se inicializa la comunicación entre Arduino y LabVIEW.

5.3.3 Comunicación Arduino-LabVIEW

La comunicación entre Arduino y LabVIEW se establece utilizando los drivers NI VISA para LabVIEW. Estos drivers permiten la comunicación de LabVIEW con un puerto serie. Un puerto serie es un sistema de comunicación de datos, en el caso de este trabajo el puerto serie un puerto USB. La característica de este tipo de puertos es que transmiten los datos bit a bit, transmitiendo solamente un bit en cada envío (Pérez Porto & Merino, 2015).

La comunicación mediante puerto serie es la utilizada por Arduino tanto para enviar como recibir datos. De este modo al utilizar VISA para LabVIEW podemos seleccionar el mismo puerto serie que utiliza Arduino y enviarle y recibir datos.

Para utilizar VISA se debe configurar previamente el puerto serie con el que se quiere establecer la conexión. Para esto LabVIEW te pide distintos parámetros como el nombre del puerto serie o la velocidad de transmisión. Una vez configurado el puerto podemos escribir o leer en el puerto. Además podemos borrar lo que haya en el puerto para hacer una nueva lectura o una nueva escritura. Por último también podemos cerrar el puerto para establecer la conexión con otro puerto (National Instruments, 2008).

5.4 Resultados

Una vez montado el prototipo se han realizado tres ensayos con diferentes fuentes lumínicas: la fuente utilizada para la calibración que consiste en una bombilla incandescente de 100 W, una segunda fuente con una bombilla led y por último se ha realizado un ensayo con luz solar.

Para el primer ensayo se utilizó la fuente calibrada, se colocó el prototipo en el suelo y se realizó la medida con la única iluminación que proporcionaba la fuente lumínica. Tras varias pruebas corroborando que los datos obtenidos tras varias medidas eran iguales se obtuvieron los siguientes resultados:

| Azul | Verde | Rojo | Vertical | Horizontal | | | | | | |
|-------|--------|--------|----------|------------|-----------|----------|------------|--------|----|-----|
| -0.54 | -7.75 | -0.82 | 0 | 0 | Max azul | Vertical | Horizontal | | | |
| 1.24 | -29.13 | -1.45 | 0 | 45 | | | | 20.46 | 81 | 135 |
| 1.02 | -32.14 | -4.32 | 0 | 90 | | | | | | |
| 0.9 | -31.45 | -3.33 | 0 | 135 | Max verde | Vertical | Horizontal | | | |
| 1.02 | -30.36 | -2.4 | 0 | 180 | | | | 185.3 | 81 | 135 |
| 0.98 | -32.67 | -4.94 | 0 | 225 | | | | | | |
| 0.75 | -35.23 | -7.35 | 0 | 270 | Max rojo | Vertical | Horizontal | | | |
| 0.68 | -31.69 | -3.34 | 0 | 315 | | | | 609.08 | 81 | 45 |
| 1.05 | -28.59 | -0.01 | 0 | 22 | | | | | | |
| 1.14 | -29.81 | 0.11 | 9 | 22 | | | | | | |
| 1.03 | -32.29 | -4.7 | 9 | 67 | | | | | | |
| 0.79 | -33.82 | -6.03 | 9 | 112 | | | | | | |
| 0.83 | -30.94 | -2.12 | 9 | 157 | | | | | | |
| 1.04 | -31.49 | -3.87 | 9 | 202 | | | | | | |
| 0.84 | -34.76 | -7.28 | 9 | 247 | | | | | | |
| 0.6 | -35.36 | -7.29 | 9 | 292 | | | | | | |
| 0.75 | -30.13 | -2 | 9 | 337 | | | | | | |
| 1.05 | -28.05 | -0.08 | 9 | 0 | | | | | | |
| 1.1 | -30.65 | -0.69 | 27 | 0 | | | | | | |
| 1.02 | -31.59 | -3.32 | 27 | 45 | | | | | | |
| 1 | -33.35 | -4.98 | 27 | 90 | | | | | | |
| 0.76 | -32.78 | -4.6 | 27 | 135 | | | | | | |
| 0.83 | -33.53 | -5.65 | 27 | 180 | | | | | | |
| 0.72 | -35.32 | -7.64 | 27 | 225 | | | | | | |
| 0.56 | -36.63 | -8.72 | 27 | 270 | | | | | | |
| 0.6 | -32.72 | -4.4 | 27 | 315 | | | | | | |
| 1.37 | -28.94 | -2.35 | 27 | 45 | | | | | | |
| 6.05 | 55.31 | 302.47 | 45 | 45 | | | | | | |
| 8.32 | 61.79 | 89.11 | 45 | 135 | | | | | | |
| 7.22 | 36.06 | 60.01 | 45 | 225 | | | | | | |
| 5.49 | 33.5 | 70.88 | 45 | 315 | | | | | | |
| 8.57 | 61.4 | 82.9 | 45 | 0 | | | | | | |
| 9.99 | 90.28 | 384.79 | 63 | 0 | | | | | | |
| 11.44 | 102.18 | 239.08 | 63 | 90 | | | | | | |
| 10.98 | 91.36 | 122.97 | 63 | 180 | | | | | | |
| 10.26 | 70.99 | 101.45 | 63 | 270 | | | | | | |
| 13.09 | 109.82 | 447.2 | 63 | 45 | | | | | | |
| 13.06 | 121.14 | 609.08 | 81 | 45 | | | | | | |
| 20.46 | 185.3 | 444.54 | 81 | 135 | | | | | | |
| 17.11 | 134.23 | 360.24 | 81 | 225 | | | | | | |
| 14.64 | 152.73 | 522.03 | 81 | 315 | | | | | | |

Fig.23 Resultados obtenidos en el primer ensayo con bombilla incandescente.

La medición se realizó con la fuente justo encima del prototipo, por lo que las primeras medidas, las cuales se hacen a 0° respecto de la vertical, es decir el vector de superficie que es perpendicular a la horizontal está a 0° con la horizontal, son notablemente más bajas y van aumentando conforme va cambiando el ángulo del vector de superficie.. Aunque es lógico que las primeras medidas sean las más bajas, se registraron varios valores negativos, esto muestra claramente una calibración no adecuada del prototipo, por lo que aunque el prototipo responda bien a los cambios de orientación debería hacerse una buena calibración para poder tener datos fiables.

Aún sabiendo que la calibración no es la mejor posible se van a analizar los datos más relevantes obtenidos:

| | | |
|-----------|----------|------------|
| Max azul | Vertical | Horizontal |
| 20.46 | 81 | 135 |
| Max verde | Vertical | Horizontal |
| 185.3 | 81 | 135 |
| Max rojo | Vertical | Horizontal |
| 609.08 | 81 | 45 |

Fig.24 Valores representativos del primer ensayo.

Como se observa los máximos valores de iluminancia en las tres longitudes de onda medidas se encuentran en un ángulo de 81° del vector de superficie con la horizontal, es decir cuando el sensor está casi enfocado directamente a la fuente, ya que el vector de superficie es perpendicular a la superficie del sensor, que se encontraba justo encima del prototipo. Este dato nos demuestra que a pesar de la calibración no adecuada el prototipo responde bien en términos de buscar la orientación en la que el sensor está más expuesto.

El otro dato relevante es el máximo de iluminancia, que corresponde a 609.08 lx en la longitud de onda del color rojo. Este dato también es lógico ya que en el espectro de la luz incandescente en la Fig.2 se observa que la longitud de onda correspondiente al rojo es predominante respecto a las demás. A pesar de no estar calibrado adecuadamente si siguiéramos las recomendaciones expuestas en la [introducción](#) esta fuente no serviría para ningún tipo de obra.

El segundo ensayo se realizó en las mismas condiciones que el primero cambiando la fuente por una bombilla led. Los resultados fueron los siguientes:

| Azul | Verde | Rojo | Vertical | Horizontal | | | |
|-------|--------|--------|----------|------------|-----------|----------|------------|
| -0.54 | -47.79 | -18.81 | 0 | 0 | Max azul | Vertical | Horizontal |
| -0.35 | -47.3 | -20.09 | 0 | 45 | 6.51 | 81 | 225 |
| -0.4 | -47.95 | -20.64 | 0 | 90 | Max verde | Vertical | Horizontal |
| -0.42 | -47.56 | -20.19 | 0 | 135 | 39.98 | 81 | 315 |
| -0.38 | -47.13 | -19.78 | 0 | 180 | Max rojo | Vertical | Horizontal |
| -0.36 | -47.42 | -20.11 | 0 | 225 | 235.59 | 81 | 45 |
| -0.4 | -47.98 | -20.65 | 0 | 270 | | | |
| -0.43 | -47.61 | -20.24 | 0 | 315 | | | |
| -0.39 | -47.05 | -19.81 | 0 | 22 | | | |
| -0.37 | -47.11 | -19.94 | 9 | 22 | | | |
| -0.39 | -47.89 | -20.61 | 9 | 67 | | | |
| -0.44 | -48.1 | -20.7 | 9 | 112 | | | |
| -0.42 | -47.52 | -20.14 | 9 | 157 | | | |
| -0.38 | -47.43 | -20.06 | 9 | 202 | | | |
| -0.39 | -47.95 | -20.57 | 9 | 247 | | | |
| -0.44 | -48.11 | -20.72 | 9 | 292 | | | |
| -0.42 | -47.32 | -19.96 | 9 | 337 | | | |
| -0.38 | -47.14 | -19.76 | 9 | 0 | | | |
| -0.32 | -43.54 | -18.01 | 27 | 0 | | | |
| -0.03 | -38.72 | -10.3 | 27 | 45 | | | |
| 0.37 | -34.63 | -6.7 | 27 | 90 | | | |
| 0.79 | -33.67 | -7.68 | 27 | 135 | | | |
| 0.71 | -35.85 | -8.92 | 27 | 180 | | | |
| 0.47 | -40.09 | -13.1 | 27 | 225 | | | |
| 0.09 | -44.77 | -17.5 | 27 | 270 | | | |
| -0.25 | -46.07 | -18.53 | 27 | 315 | | | |
| 0.51 | -38.52 | -11.68 | 27 | 45 | | | |
| 2.01 | -6.99 | 51.87 | 45 | 45 | | | |
| 3.03 | -4.14 | 23.74 | 45 | 135 | | | |
| 2.9 | -9.32 | 17.03 | 45 | 225 | | | |
| 2.48 | -8.7 | 16.39 | 45 | 315 | | | |
| 3.27 | -4.97 | 21.03 | 45 | 0 | | | |
| 4.2 | 18.63 | 163.51 | 63 | 0 | | | |
| 5.18 | 24.13 | 51.3 | 63 | 90 | | | |
| 5.12 | 20.07 | 44.9 | 63 | 180 | | | |
| 4.45 | 9.91 | 37.61 | 63 | 270 | | | |
| 5.48 | 26.2 | 187.68 | 63 | 45 | | | |
| 6.12 | 38.51 | 235.59 | 81 | 45 | | | |
| 6.32 | 36.8 | 65.98 | 81 | 135 | | | |
| 6.51 | 39.72 | 65.89 | 81 | 225 | | | |
| 6.43 | 39.98 | 71.32 | 81 | 315 | | | |

Fig.25 Resultados obtenidos en el segundo ensayo con bombilla led.

Se observa que este caso sigue el mismo patrón que el primer ensayo, obteniéndose los valores más altos en el ángulo más alto del vector de superficie respecto a la horizontal.

Por último se realizó un ensayo con luz solar dando los siguientes resultados:

| Azul | Verde | Rojo | Vertical | Horizontal | | | |
|-------|--------|--------|----------|------------|-----------|----------|------------|
| -0.54 | 629.07 | 759.23 | 0 | 0 | Max azul | Vertical | Horizontal |
| 61.96 | 731.67 | 759.08 | 0 | 45 | 61.96 | 0 | 45 |
| 61.95 | 731.6 | 759.11 | 0 | 90 | Max verde | Vertical | Horizontal |
| 61.95 | 731.61 | 759.06 | 0 | 135 | 731.67 | 0 | 45 |
| 61.94 | 731.52 | 758.92 | 0 | 180 | Max rojo | Vertical | Horizontal |
| 61.93 | 731.41 | 758.89 | 0 | 225 | 759.23 | 0 | 0 |
| 61.93 | 731.38 | 758.78 | 0 | 270 | | | |
| 61.92 | 731.26 | 758.76 | 0 | 315 | | | |
| 61.9 | 731.07 | 758.5 | 0 | 22 | | | |
| 61.9 | 731.03 | 758.52 | 9 | 22 | | | |
| 61.9 | 730.89 | 758.22 | 9 | 67 | | | |
| 61.88 | 730.71 | 758.22 | 9 | 112 | | | |
| 61.88 | 730.74 | 758.22 | 9 | 157 | | | |
| 61.88 | 730.71 | 758.21 | 9 | 202 | | | |
| 61.88 | 730.63 | 758.15 | 9 | 247 | | | |
| 61.87 | 730.62 | 758.11 | 9 | 292 | | | |
| 61.87 | 730.6 | 758.08 | 9 | 337 | | | |
| 61.85 | 730.39 | 757.88 | 9 | 0 | | | |
| 61.85 | 730.47 | 757.88 | 27 | 0 | | | |
| 61.85 | 730.3 | 757.6 | 27 | 45 | | | |
| 61.83 | 730.06 | 757.56 | 27 | 90 | | | |
| 61.82 | 729.94 | 757.51 | 27 | 135 | | | |
| 61.82 | 730.07 | 757.55 | 27 | 180 | | | |
| 61.83 | 730.09 | 757.53 | 27 | 225 | | | |
| 61.83 | 730.09 | 757.55 | 27 | 270 | | | |
| 61.83 | 730.05 | 757.53 | 27 | 315 | | | |
| 61.82 | 729.88 | 757.28 | 27 | 45 | | | |
| 61.8 | 729.67 | 757.1 | 45 | 45 | | | |
| 61.78 | 729.5 | 757.08 | 45 | 135 | | | |
| 61.8 | 729.74 | 757.24 | 45 | 225 | | | |
| 61.8 | 729.81 | 757.23 | 45 | 315 | | | |
| 61.79 | 729.6 | 757.1 | 45 | 0 | | | |
| 61.79 | 729.63 | 757.09 | 63 | 0 | | | |
| 61.78 | 729.46 | 756.92 | 63 | 90 | | | |
| 61.78 | 729.53 | 757.04 | 63 | 180 | | | |
| 61.79 | 729.6 | 757.05 | 63 | 270 | | | |
| 61.78 | 729.43 | 756.88 | 63 | 45 | | | |
| 61.77 | 729.27 | 756.69 | 81 | 45 | | | |
| 61.75 | 729.14 | 756.56 | 81 | 135 | | | |
| 61.76 | 726.97 | 751.42 | 81 | 225 | | | |
| 61.34 | 724.03 | 751.55 | 81 | 315 | | | |

Fig.26 Resultados obtenidos en el tercer ensayo realizado con luz solar.

En este caso se observa como el sensor satura, es decir llega al límite de su capacidad de lectura y los valores se repiten. Debido a estos resultados se concluye que el prototipo con el sensor elegido no puede medir luz solar, únicamente luz de interior.

6. CONCLUSIONES

6.1 Objetivos superados

Como se observa el prototipo cumple todos los objetivos propuestos en el apartado 3, sin embargo se observa que el sensor da lecturas negativas, esto se debe a una calibración no adecuada. Desafortunadamente no se pudo realizar una segunda calibración, ya que el espectrofotómetro utilizado para la calibración se encontraba en el Servicio de Conservación y Restauración de Bienes Culturales de la Diputación de Castellón y por diversos motivos no se pudo concertar una segunda cita para una calibración más precisa. Aunque la calibración no era un objetivo en sí es una parte muy importante del trabajo que no se ha podido realizar con la exactitud deseada. Por lo tanto se debería volver a calibrar la fuente y modificar los datos de esta calibración en la programación.

6.2 Propuestas de mejora

Tras analizar las características del prototipo una vez finalizado se ha concluido que tiene mucho margen de mejora, estas son las propuestas de mejora que se han elegido:

- Realizar una calibración con más exactitud de la fuente para poder realizar lecturas fiables.
- Gestionar mejor el cableado en la estructura, ya que además de que estéticamente no es agradable puede ocasionar problemas de enredo durante el funcionamiento del prototipo.
- Realizar la conexión con el ordenador de forma inalámbrica así como alimentarlo con una batería externa y de este modo poder tomar medidas de una forma más cómoda.
- Mejorar la interfaz con el usuario aportando más datos como gráficas o diagramas que muestren la iluminancia en función de la orientación, dando más información y haciéndola más agradable visualmente.
- Trasladar la interfaz gráfica a un software propio eliminando la necesidad de obtener LabVIEW por parte del usuario.
- Realizar una estructura más compacta, minimalista y práctica.

7. PRESUPUESTO

Para confeccionar el presupuesto pese a que se trata de un trabajo académico se ha decidido asemejarlo al trabajo que podría realizar una oficina técnica, teniendo en cuenta partidas como los gastos generales y el beneficio industrial además del 21% de IVA en vigor.

A la hora de elaborar el presupuesto se han tenido en cuenta las 300 horas realizadas por el alumno así como ciertas horas realizadas por el tutor que se han considerado reflejar. Estas horas corresponden a trabajos de montaje de prototipo ya que el tutor hizo una gran aportación en este apartado. En el presupuesto el tutor aparece como “colaborador”.

En el presente apartado se exponen el cuadro de precios unitarios y el cuadro resumen del presupuesto. Para una descripción más detallada del presupuesto se adjunta el cuadro de precios descompuestos y el fichero Excel utilizado para realizar el presupuesto en el [Anexo XI](#).

| CUADRO PRECIOS UNITARIOS | | | | | | |
|--------------------------|------|-----|------------------------------------------------|----------|---------|----------------|
| Capítulo | U.O. | Ud. | Descripción | Medición | Precio | Importe(€) |
| 1 | | | Diseño e implementación de algoritmos | | | |
| | 1 | h | Arduino | 70 | 36.50 | 2555.14 |
| | 2 | h | Labview | 55 | 35.50 | 1952.61 |
| | | | Total Capítulo 1 | | | 4507.75 |
| 2 | | | Diseño y montaje del prototipo | | | |
| | 1 | Ud | Diseño y montaje PCB | 1 | 2016.54 | 2016.54 |
| | 2 | Ud | Elementos para la automatización | 1 | 88.82 | 88.82 |
| | 3 | Ud | Estructura | 1 | 454.52 | 454.52 |
| | | | Total Capítulo 2 | | | 2559.88 |
| 3 | | | Calibración | | | |
| | 1 | Ud | Calibración fuente lumínica | 1 | 183.6 | 183.6 |
| | 2 | Ud | Calibración prototipo | 1 | 357 | 357 |
| | | | Total Capítulo 3 | | | 540.6 |
| 4 | | | Investigación y búsqueda de información | | | |
| | 1 | Ud | Investigación | 1 | 3570 | 3570 |
| | | | Total Capítulo 4 | | | 3570 |

CUADRO RESUMEN PRESUPUESTO

| Capítulo | Descripción | Importe (€) |
|----------|-------------------------------------------|-----------------|
| 1 | Diseño e implementación de algoritmos | 4507.75 |
| 2 | Diseño y montaje del prototipo | 2559.88 |
| 3 | Calibración | 540.6 |
| 4 | Investigación y búsqueda de información | 3570 |
| | PRESUPUESTO EJECUCIÓN MATERIAL | 11178.23 |
| | Gastos Generales (12%) | 1341.39 |
| | Beneficio Industrial (6%) | 670.69 |
| | PRESUPUESTO EJECUCIÓN POR CONTRATA | 13190.32 |
| | IVA (21%) | 2769.97 |
| | PRESUPUESTO BASE LICITACIÓN | 15960.28 |

8. BIBLIOGRAFÍA

Aliexpress. (s.f.). *5 unids Impresora 3D StepStick DRV 8825 Conductor Del Motor de Pasos Soporte de módulo de capas 4 capas PCB Reprap RAMPAS 1.4 + disipador de calor*. Recuperado el 13 de Junio de 2016, de Aliexpress: http://es.aliexpress.com/store/product/5pcs-lot-3D-Printer-StepStick-DRV8825-Stepper-Motor-Driver-Module-Carrier-Reprap-4-layer-4-layer/1757110_32351427821.html?spm=2114.04010208.3.2.eH5LzC&ws_ab_test=searchweb201556_0,searchweb201602_3_10017_405_10044_10

Aliexpress. (s.f.). *Envío gratis 10 unids/lote nuevo Micro Roller Lever Arm normalmente abierto cerrar Limit Switch KW12-3*. Recuperado el 13 de Junio de 2016, de Aliexpress: <http://es.aliexpress.com/item/Free-Shipping-10pcs-lot-New-Micro-Roller-Lever-Arm-Normally-Open-Close-Limit-Switch-KW12-3/32657449949.html?spm=2114.43010308.4.41.eUQedi>

Aliexpress. (s.f.). *Motor de pasos 2PHASE 17HS3430 longitud 34 mm angle 1.8 paso 28 n. cmholding torque (min) 1.6 n. cm retén torque (max)*. Recuperado el 13 de Mayo de 2016, de Aliexpress: <http://es.aliexpress.com/item/Stepper-Motor-2PHASE-17HS3430-1-8-step-angle-28-N-cmholding-torque-min-1-6-N/646805486.html>

Aliexpress. (s.f.). *Muestra, 50 sets 4 Pin XH 90 ángulo de pliegue Pin Connector 2.54 mm XH-4P Kits para PCB / automotriz / placa de circuito electrónico ect. envío gratis*. Recuperado el 10 de Mayo de 2016, de Aliexpress: http://es.aliexpress.com/store/product/Sample-50-sets-4-Pin-XH-90-angle-bend-Pin-Connector-2-54mm-XH-4P-Kits/335862_32329276081.html?spm=2114.04010208.3.2.Uwi983&ws_ab_test=searchweb201556_0,searchweb201602_3_10017_10044_10043_406_10041_10042,searchweb201

Aliexpress. (s.f.). *Nueva 10 Pairs 40 Pin 2.54 mm Male & Female SIL Header Socket fila PCB conector caliente venta del producto*. Recuperado el 13 de Junio de 2016, de Aliexpress: <http://es.aliexpress.com/item/New-10-Pairs-40-Pin-2-54mm-Male-Female-SIL-Header-Socket-Row-Strip-PCB-Connector/32545598891.html?spm=2114.43010208.4.71.tUOPJH>

Arduino. (s.f.). *Arduino Duemilanove*. Recuperado el 4 de Abril de 2016, de Arduino: <https://www.arduino.cc/en/Main/ArduinoBoardDuemilanove>

Arduino. (s.f.). *Language Reference*. Recuperado el 4 de Abril de 2016, de Arduino: <https://www.arduino.cc/en/Reference/HomePage>

BOE. (2013 de Octubre de 25). *XVII CONVENIO COLECTIVO NACIONAL DE EMPRESAS DE INGENIERÍA Y OFICINAS*. Recuperado el 4 de Junio de 2016, de BOE: <https://www.boe.es/boe/dias/2013/10/25/pdfs/BOE-A-2013-11199.pdf>

campusMVP. (24 de Febrero de 2014). *Los lenguajes de programación más demandados por las empresas*. Recuperado el 10 de Mayo de 2016, de campusmvp: <http://www.campusmvp.es/recursos/post/Los-lenguajes-de-programacion-mas-demandados-por-las-empresas.aspx>

Carlos. (2 de Marzo de 2007). *Bombillas fluorescentes: ¿es oro todo lo que reluce?* Recuperado el 10 de Mayo de 2016, de Wordpress: <https://singularidad.wordpress.com/2007/03/02/bombillas-fluorescentes-%C2%BFes-oro-todo-lo-que-reluce/>

cetronic. (s.f.). *DC1C-A1LC INTERRUPTOR FINAL DE CARRERA DE PALANCA*. Recuperado el 10 de Mayo de 2016, de cetronic: <http://www.cetronic.es/sqlcommerce/disenos/plantilla1/seccion/producto/DetalleProducto.jsp?idIoma=&idTienda=93&codProducto=354253011&cPath=1160>

González, E. G. (2006). *GUÍA BÁSICA DE CONCEPTOS DE RADIOMETRÍA Y FOTOMETRÍA*. Recuperado el 18 de Mayo de 2016, de UNIVERSIDAD DE SEVILLA: <http://laplace.us.es/campos/optica/general/opt-guia2.pdf>

ICOM. (2010). *Appllyiing Preventtiive Conservattiion Recommendatttiions ffor Siillk iin*. Recuperado el 30 de Marzo de 2016, de ICOM: https://www.icom-cc.org/54/document/applying-preventive-conservation-recommendations-for-silk-in-historic-houses/?action=Site_Downloads_Downloadfile&id=813

jeffkrol. (Junio de 2013). *light spectrum question*. Recuperado el 10 de Mayo de 2016, de plantedtank: <http://www.plantedtank.net/forums/10-lighting/1057362-light-spectrum-question.html>

Leroy Merlin. (s.f.). *Paquete de 150 tuercas SUKI INTERNATIONAL GMBH HEXAGONAL ACERO ZINCADO*. Recuperado el 3 de Junio de 2016, de Leroy Merlin: <http://www.leroymerlin.es/fp/15659371/paquete-de-150-tuercas-suki-international-gmbh-hexagonal-acero-zincado>

Leroy Merlin. (s.f.). *Paquete de 35 tornillos tirafondo CABEZA REDONDA HENDIDA ACERO ZINCADO*. Recuperado el 3 de Junio de 2016, de Leroy Merlin: <http://www.leroymerlin.es/fp/15680546/paquete-de-35-tornillos-tirafondo-cabeza-redonda-hendida-acero-zincado>

Leroy Merlin. (s.f.). *Tablero contrachapado CONTRACHAPADO*. Recuperado el 3 de Junio de 2016, de Leroy Merlin: <http://www.leroymerlin.es/fp/18896773/tablero-contrachapado-contrachapado>

Leroy Merlin. (s.f.). *Varilla CQFD ROSCADAS ACERO ZINCADO*. Recuperado el 3 de Junio de 2016, de Leroy Merlin: <http://www.leroymerlin.es/fp/57743/varilla-roscadas-acero-zincado?pathFamiliaFicha=420108>

Mánuel Lázaro, A., & del Río Fernández, J. (2005). *LabView 7.1 Programación Gráfica para el Control de Instrumentación*. Madrid: Thomson.

Mercadé, A. (12 de Julio de 2011). *PROPIEDADES DE LA LUZ*. Recuperado el 10 de Mayo de 2016, de Transformando el infierno: <https://transformandoelinfierno.com/2011/07/12/propiedades-de-la-luz/>

Mims, F. M. (s.f.). *How to Use LEDs to Detect Light*. Recuperado el 3 de Marzo de 2016, de makezine: <http://makezine.com/projects/make-36-boards/how-to-use-leds-to-detect-light/>

MotionKing. (s.f.). *2 Phase Hybrid Stepper Motor*. Recuperado el 10 de Mayo de 2016, de datasheet: <http://www.datasheet.es/PDF/928658/17HS3430-pdf.html>

National Instruments. (19 de Septiembre de 2008). *Comunicación Serial Utilizando LabVIEW con un Microcontrolador*. Recuperado el 13 de Junio de 2016, de National Instruments: <http://www.ni.com/white-paper/7907/es/>

Newport. (2011). *THE NEWPORT RESOURCE*.

Pérez Porto, J., & Merino, M. (2015). *Puerto seral*. Recuperado el 10 de Mayo de 2016, de definición: <http://definicion.de/puerto-serial/>

Pololu. (Julio de 2014). *DRV8825 Stepper Motor Controller IC*. Recuperado el 11 de Abril de 2016, de Pololu: <https://www.pololu.com/file/0J590/drv8825.pdf>

Roger Folch, J., Riera Guasp, M., & Roldán Porta, C. (2010). *Tecnología eléctrica*. Madrid: Síntesis.

RS. (s.f.). *Alojamiento de PCB Molex 22-01-3047, Serie KK 254, paso: 2.54mm, 4 contactos, 1 fila, Recto, Hembra, Montaje de Cable*. Recuperado el 13 de Junio de 2016, de RS: <http://es.rs-online.com/web/p/carcasas-de-conectores-para-pcb/6795388/>

RS. (s.f.). *Conector macho para PCB Molex Ángulo de 90° 4 pines 1 fila paso 2.54mm, Terminación Soldada*. Recuperado el 13 de Junio de 2016, de RS: es.rs-online.com/web/p/conectores-macho-para-pcb/1732944/?origin=PSF_435704|acc

RS. (s.f.). *Sensor de colores, TCS3200D, Luz de color a frecuencia 470 nm, 524 nm, 640 nm SOIC 8 pines*. Recuperado el 11 de Abril de 2016, de RS: http://es.rs-online.com/web/p/sensores-de-luz-y-de-color/6424339/?origin=null|fp&cm_sp=featureproducts--FeaturedProductsContent--6424339

solostocks. (s.f.). *Tarima madera machihembrada*. Recuperado el 4 de Junio de 2016, de solostocks: <http://www.solostocks.com/venta-productos/suelos-revestimiento/suelos-madera/tarima-madera-machihembrada-6545538>

TAOS. (s.f.). *TCS3200 datasheet*. Recuperado el 11 de Abril de 2016, de RS: <http://docs-europe.electrocomponents.com/webdocs/0892/0900766b80892846.pdf>

Voipminic. (s.f.). *Placa PCB fotosensible positiva Fibra 1 Cara*. Recuperado el 12 de Mayo de 2016, de Voipminic: <http://shop.voipminic.com/circuitos-electronicos/2139-pcb-fibra-1-cara.html>

X-Rite. (s.f.). *Manual de uso*. Recuperado el 16 de Mayo de 2016, de X-Rite: http://www.xrite.com/documents/apps/public/Manuals/EO2UV-QSG_i1Pro_UVcut_User_Manual.pdf

9. ANEXOS

Anexo I - Manual de usuario

Como se ha mostrado en el apartado [5.3.2](#) la interfaz gráfica es muy sencilla:



Fig.27 Interfaz gráfica vista por el usuario creada con LabVIEW.

Para realizar una medición es necesario disponer de un ordenador con el software Arduino y LabVIEW. Además se debe disponer de un cable USB para conectar el prototipo al ordenador y un transformador de 12 V con puerto jack para alimentar el prototipo.

Una vez conectado al ordenador se ejecuta el programa sensor luz.vi y aparece la pantalla que se muestra en la Fig. En esta pantalla elegimos el nombre del puerto que hemos conectado al ordenador, el número de puntos en el que queremos medir y accionamos el programa con la flecha negra que hay en la parte superior.

Lo primero que hace el programa es pedirnos la ubicación en la que se va a guardar el fichero creado y el tipo de fichero mediante la extensión. Se recomienda nombrar al fichero con la extensión .xls ya que de este modo se podrá abrir mediante Excel y los datos se analizarán cómodamente.

Una vez hecho esto el programa empieza a tomar medidas, dependiendo del ángulo que se quiera abarcar se deberá seleccionar un número de puntos representado en el siguiente cuadro:

| Ángulo girado respecto a la vertical | Nº de medidas |
|--------------------------------------|---------------|
| 0º | 8 |
| 9º | 17 |
| 27º | 26 |
| 45º | 31 |
| 63º | 36 |
| Recorrido completo | 41 |

Fig.28 Tabla resumen del ángulo abarcado por el sensor dependiendo del nº de medidas tomadas.

Según se van tomando medidas aparecen en pantalla y una vez que el proceso finaliza quedan registradas en el fichero previamente elegido.

Anexo II – Documentación técnica sensor de color TCS3200D

Para una completa información véase el datasheet en:

https://drive.google.com/open?id=0B_B1eCxfBX73amRGdDVubDdmZGs

(TAOS, s.f.)

Anexo III – Documentación técnica placa Arduino

Microcontrolador ATmega328

Voltaje de funcionamiento: 5V

Voltaje de entrada (recomendado): 7-12V

Voltaje de entrada (límites): 6-20V

Pines digital I/O: 14 (de los cuales 6 proporcionan salidas PWM)

Pines de entrada analógicos: 6

Corriente DC por los pines I/O: 40mA

Corriente DC por el pin de 3.3V: 50mA

Memoria Flash: 32KB (ATmega328) de los cuales 2KB son usados para el programa de arranque

Memoria SRAM: 2KB (ATmega328)

Memoria EEPROM: 1KB (ATmega328)

Velocidad de reloj: 16MHZ

(Arduino, s.f.)

Anexo IV – Documentación técnica motor paso a paso

Ángulo de paso: 1.8 °

Longitud del motor: 34 mm

Corriente nominal: 0.4 A

Resistencia por fase 30 Ω

Par de mantenimiento: 28 N.cm Min

Par de retención: 1.6 N.cm

Inercia del rotor: 34 g.cm²

Número de cables: 4

Peso del motor: 220 g

(MotionKing, s.f.)

Anexo V – Documentación técnica controlador motor paso a paso POLOLU DRV8825

Para una completa información véase el datasheet en:

https://drive.google.com/open?id=0B_B1eCxfBX73eThjMXVWYUVhMms

(Pololu, 2014)

Anexo VI – Código Arduino

```
#include <FreqCount.h>

//declaración de los pins a utilizar del arduino
int i;
int fc1 = 2;
int fc2 = 3;
int S0 = 6;
int S1 = 7;
int S2 = 4;
int S3 = 8;
int pin = 5;
int dirPin = 10;
int stepperPin = 9;
int dirPin2 = 13;
int stepperPin2 = 11;
//declaración variables de lectura
float azul;
float verde;
float rojo;

// funciones para manejar los motores
void paso(boolean dir,int steps){
  digitalWrite(dirPin,dir);
  delay(50);
  for(int i=0;i<steps;i++){
    digitalWrite(stepperPin, HIGH);
    delayMicroseconds(100);
    digitalWrite(stepperPin, LOW);
    delayMicroseconds(100);
  }
}

void paso2(boolean dir2,int steps2){
  digitalWrite(dirPin2,dir2);
  delay(50);
  for(int i=0;i<steps2;i++){
    digitalWrite(stepperPin2, HIGH);
    delayMicroseconds(100);
    digitalWrite(stepperPin2, LOW);
    delayMicroseconds(100);
  }
}
```

```

//funciones para medir a las distintas longitudes de onda
void medirazul(){
  digitalWrite(S0, HIGH);
  digitalWrite(S1, HIGH);
  digitalWrite(S2, LOW);
  digitalWrite(S3, HIGH);
  while (FreqCount.available() != true){
    delay(1);}

  unsigned long count = FreqCount.read();//lectura de la frecuencia
  azul=0.00008*count-0.544;// conversión de frecuencia a luxes
  Serial.print(azul);
  Serial.print(";");

}

void medirverde(){
  digitalWrite(S0, HIGH);
  digitalWrite(S1, HIGH);
  digitalWrite(S2, HIGH);
  digitalWrite(S3, HIGH);
  while (FreqCount.available() != true){
    delay(1);}

  unsigned long count = FreqCount.read();//lectura frecuencia
  verde=0.001*count-49.55;//conversion de frecuencia a luxes
  Serial.print(verde);
  Serial.print(";");

}

void medirrojo(){
  digitalWrite(S0, HIGH);
  digitalWrite(S1, HIGH);
  digitalWrite(S2, LOW);
  digitalWrite(S3, LOW);
  while (FreqCount.available() != true){
    delay(1);}

  unsigned long count = FreqCount.read();
  rojo=0.001*count-22.08;
  Serial.print(rojo);
  Serial.print(";");

}

```

```

void setup() {

  Serial.begin(57600);//
  FreqCount.begin(1000);
  //establecer el modo entrada/salida de cada pin para medir los luxes
  pinMode(pin, INPUT);
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  //establecer el modo salida de cada pin para manejar el motor
  pinMode(dirPin, OUTPUT);
  pinMode(stepperPin, OUTPUT);
  pinMode(dirPin2, OUTPUT);
  pinMode(stepperPin2, OUTPUT);
  //establecer el modo entrada para los finales de carrera
  pinMode(fc1, INPUT_PULLUP);
  pinMode(fc2, INPUT_PULLUP);
  while (Serial.available()==0){
    delay(1);}
  while (Serial.read()!=83){
    delay(1);}
  while (digitalRead(fc2)==HIGH){
    paso(false,32);}
  while(digitalRead(fc1)==HIGH){
    paso2(false,32);}

void loop() {

while (Serial.available()==0){
  delay(1);}

  while (Serial.read()!=77){
    delay(1);}

    medirazul();
    medirverde();
    medirrojo();
    Serial.print("0;");
    Serial.print("0;");
    Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

  for(i=0;i<25;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("0;");
  Serial.print("45;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

```

```

while (Serial.read()!=77){
  delay(1);}

```

```

  for(i=0;i<25;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("0;");
  Serial.print("90;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

```

```

while (Serial.read()!=77){
  delay(1);}

```

```

  for(i=0;i<25;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("0;");
  Serial.print("135;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}
  for(i=0;i<25;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("0;");
  Serial.print("180;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

  for(i=0;i<25;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("0;");
  Serial.print("225;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

  for(i=0;i<25;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("0;");
  Serial.print("270;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

  for(i=0;i<25;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("0;");
  Serial.print("315;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

  for(i=0;i<163;i++){
    paso(false,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("0;");
  Serial.print("22;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

  for(i=0;i<5;i++){
    paso2(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("9;");
  Serial.print("22;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<25;i++){
        paso(true,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("9;");
  Serial.print("67;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<25;i++){
        paso(true,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("9;");
  Serial.print("112;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<25;i++){
        paso(true,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("9;");
  Serial.print("157;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

  for(i=0;i<25;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("9;");
  Serial.print("202;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

  for(i=0;i<25;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("9;");
  Serial.print("247;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

  for(i=0;i<25;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("9;");
  Serial.print("292;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<25;i++){
        paso(true,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("9;");
  Serial.print("337;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<187;i++){
        paso(false,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("9;");
  Serial.print("0;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<10;i++){
        paso2(true,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("27;");
  Serial.print("0;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<25;i++){
        paso(true,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("27;");
  Serial.print("45;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<25;i++){
        paso(true,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("27;");
  Serial.print("90;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<25;i++){
        paso(true,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("27;");
  Serial.print("135;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read() !=77){
  delay(1);}

  for(i=0;i<25;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("27;");
  Serial.print("180;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read() !=77){
  delay(1);}

  for(i=0;i<25;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("27;");
  Serial.print("225;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read() !=77){
  delay(1);}

  for(i=0;i<25;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("27;");
  Serial.print("270;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<25;i++){
        paso(true,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("27;");
  Serial.print("315;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<150;i++){
        paso(false,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("27;");
  Serial.print("45;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<10;i++){
        paso2(true,32);
      }

  medirazul();
  medirverde();
  medirrojo();
  Serial.print("45;");
  Serial.print("45;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

  for(i=0;i<50;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("45;");
  Serial.print("135;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

  for(i=0;i<50;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("45;");
  Serial.print("225;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

  for(i=0;i<50;i++){
    paso(true,32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("45;");
  Serial.print("315;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read() !=77){
  delay(1);}

  for(i=0;i<175;i++){
    paso(false, 32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("45;");
  Serial.print("0;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read() !=77){
  delay(1);}

  for(i=0;i<10;i++){
    paso2(true, 32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("63;");
  Serial.print("0;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read() !=77){
  delay(1);}

  for(i=0;i<50;i++){
    paso(true, 32);
  }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("63;");
  Serial.print("90;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<50;i++){
        paso(true,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("63;");
  Serial.print("180;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<50;i++){
        paso(true,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("63;");
  Serial.print("270;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<125;i++){
        paso(false,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("63;");
  Serial.print("45;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<10;i++){
        paso2(true,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("81;");
  Serial.print("45;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<50;i++){
        paso(true,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("81;");
  Serial.print("135;");
  Serial.print("X;");

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}
      for(i=0;i<50;i++){
        paso(true,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("81;");
  Serial.print("225;");
  Serial.print("X;");

```

```

while (Serial.available()==0){
  delay(1);}

while (Serial.read()!=77){
  delay(1);}

      for(i=0;i<50;i++){
        paso(true,32);
      }
  medirazul();
  medirverde();
  medirrojo();
  Serial.print("81;");
  Serial.print("315;");
  Serial.print("X;");

while (digitalRead(fc2)==HIGH){
  paso(false,32);
  }

while(digitalRead(fc1)==HIGH){
  paso2(false,32);}

}

```


Anexo VIII - PCB trabajo previo

Como se ha comentado al principio del documento hubo un trabajo previo considerado de importancia, en ese trabajo se diseñaron varias PCB, solucionando fallos hasta llegar a un prototipo final, en este anexo se quiere mostrar el prototipo final fruto de varias iteraciones hasta conseguir un modelo correcto:

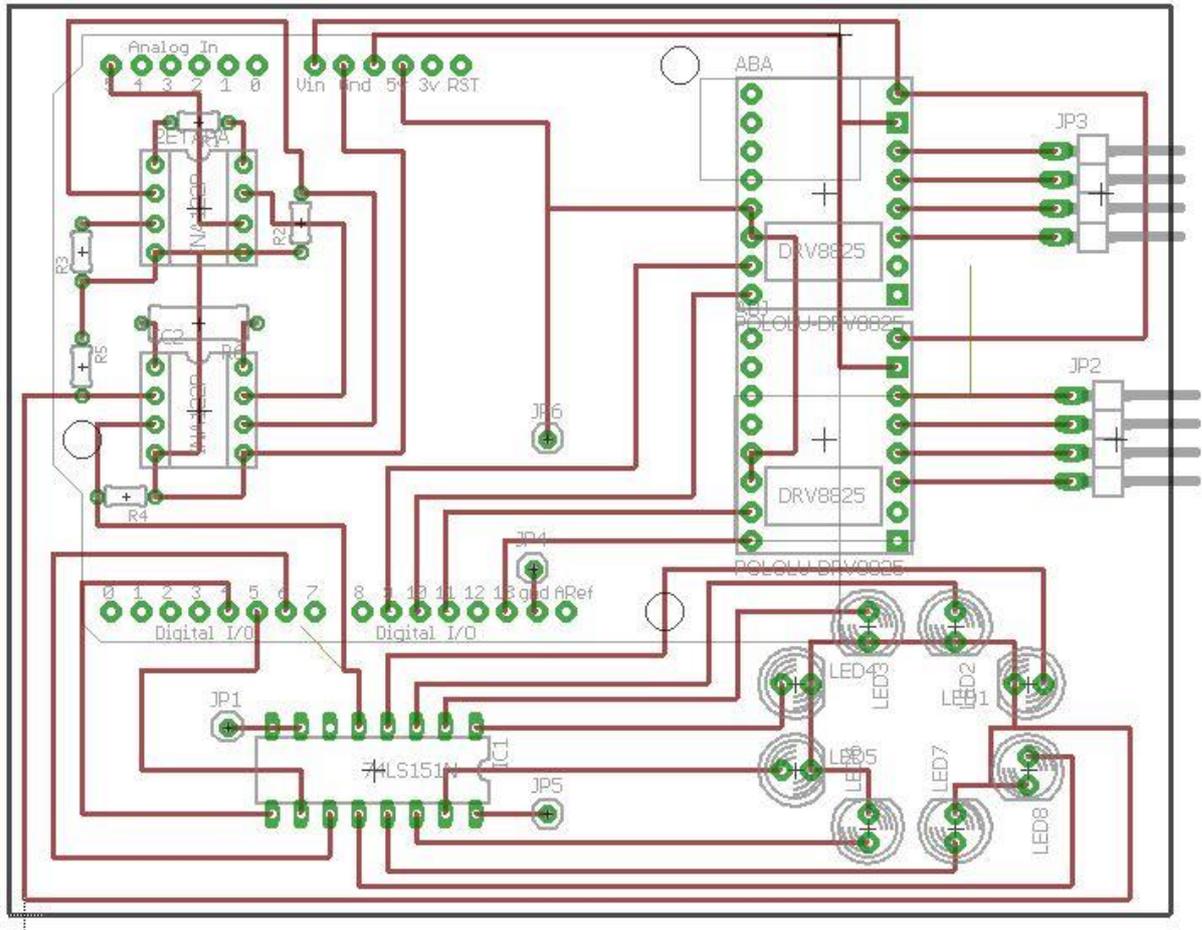


Fig.30 Distribución PCB final del trabajo previo realizado.

Anexo IX – Esquemático PCB

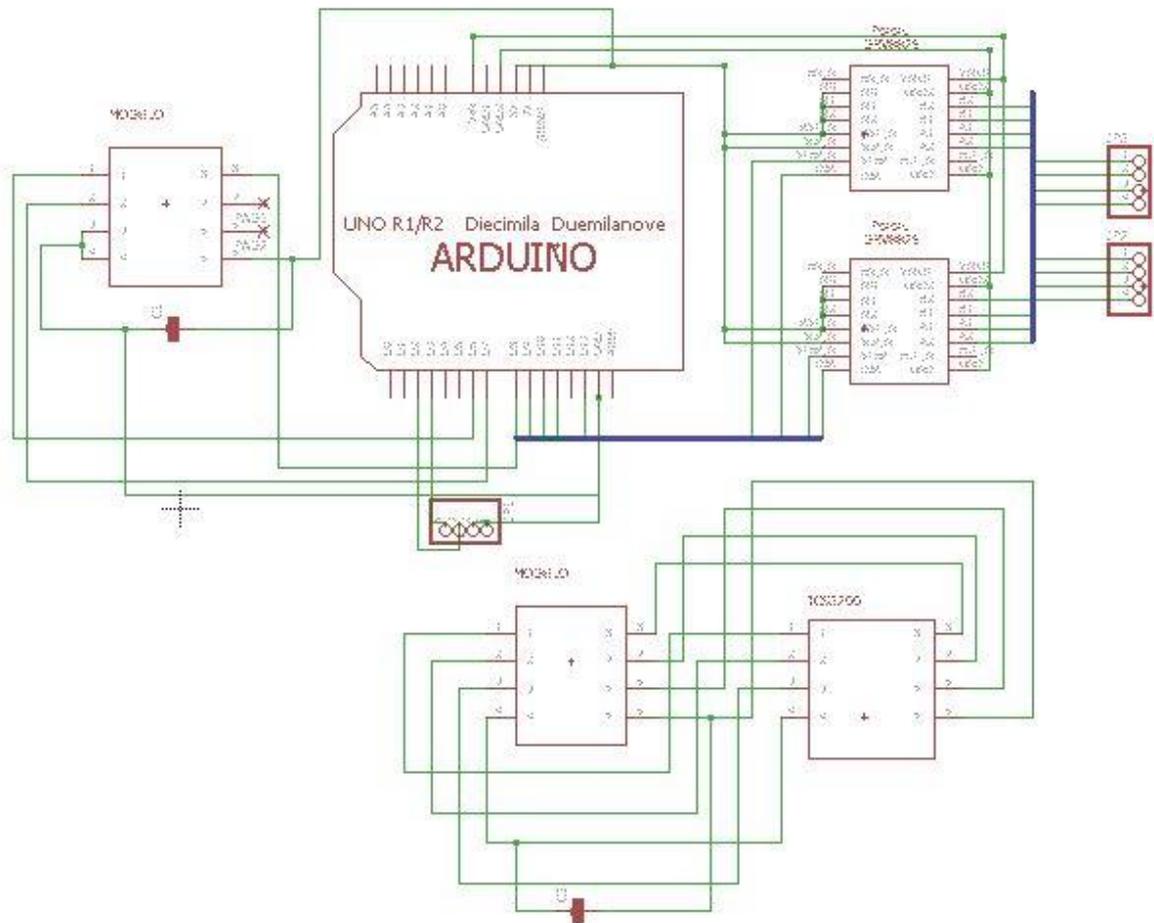


Fig.31 Esquemático de la PCB final.

Para una información más detallada se puede consultar el archivo que debe abrirse con el software Eagle en el siguiente enlace:

https://drive.google.com/open?id=0B_B1eCxfBX73VTdQRVJ5NVpkSDg

Anexo X – Distribución PCB

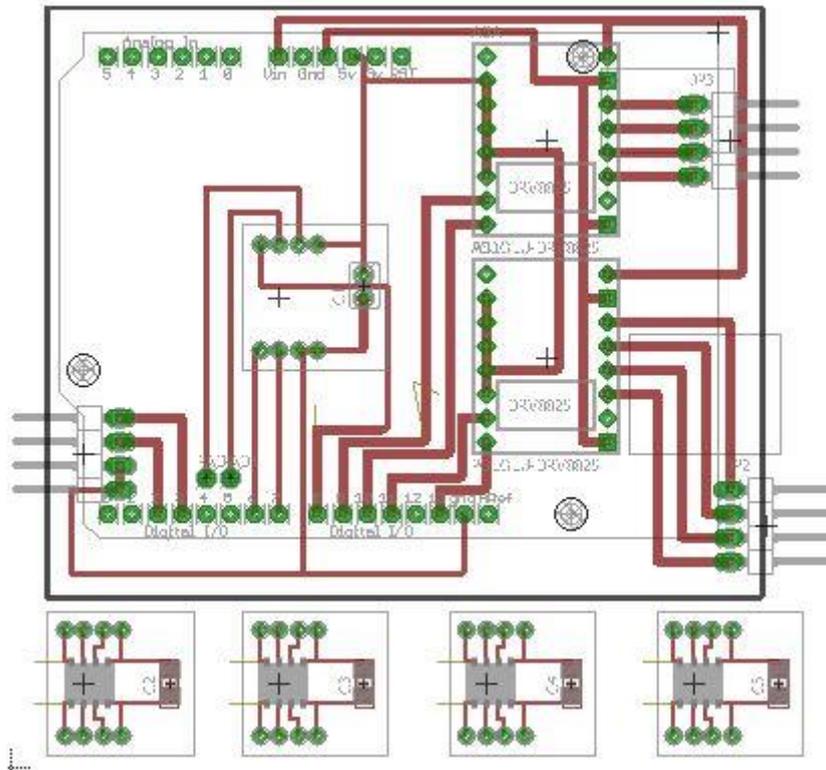


Fig.32 Distribución de la PCB final.

Para una información más detallada se puede consultar el archivo que debe abrirse con el software Eagle en el siguiente enlace:

https://drive.google.com/open?id=0B_B1eCxfBX73TExRVEFTdzdYNFU

Anexo XI – Cuadro de precios descompuestos

CUADRO DE PRECIOS DESCOMPUESTOS

| Capítulo | U.O. | Ud. Descripción | Rendimiento | Precio | Importe(€) |
|----------|------|----------------------------------------------------------|-------------|--------|---------------|
| 1 | | Diseño e implementación de algoritmos | | | |
| | 1 | Arduino | | | |
| | | Ud Software | 1.0 | 0.0 | 1.0 |
| | | h Amortización PC | 1.0 | 0.1 | 0.1 |
| | | h Graduado en GITI | 1.0 | 35.0 | 35.0 |
| | | Costes directos | | | 36.1 |
| | | Costes directos complementarios (2% herramientas) | 0.02 | 20.1 | 0.40 |
| | | Coste total | | | 36.50 |
| | 2 | LabVIEW | | | |
| | | Ud Software | 1.0 | 0.0 | 0.0 |
| | | h Amortización PC | 1.0 | 0.1 | 0.1 |
| | | h Graduado en GITI | 1.0 | 35.0 | 35.0 |
| | | Costes directos | | | 35.1 |
| | | Costes directos complementarios (2% herramientas) | 0.02 | 20.1 | 0.40 |
| | | Coste total | | | 35.50 |
| 2 | | Diseño y montaje del prototipo | | | |
| | 1 | Diseño y montaje PCB | | | |
| | | Ud Placa PCB fotosensible positiva fibra 1 cara | 1.0 | 1.9 | 1.9 |
| | | h Amortización insoladora | 1.0 | 0.1 | 0.1 |
| | | h Graduado en GITI | 50.0 | 35.0 | 1750.0 |
| | | h Colaborador | 15.0 | 15.0 | 225.0 |
| | | Costes directos | | | 1977.0 |
| | | Costes directos complementarios (2% herramientas) | 0.02 | 1977.0 | 39.54 |
| | | Coste total | | | 2016.5 |
| | 2 | Elementos para la automatización | | | |
| | | Ud Placa Arduino Duemilanove Atmega 328 | 1.0 | 25.0 | 25.0 |
| | | Ud Sensor TCS3200D | 1.0 | 4.24 | 4.24 |
| | | m Cable conexiones | 1.5 | 2.8 | 4.2 |
| | | Ud Librería Arduino para medir frecuencia | 1.0 | 0.0 | 0.0 |
| | | Ud Pack pines de conexión para PCB | 1.0 | 2.35 | 2.35 |
| | | Ud Motor de pasos 17HS3430 | 2.0 | 17.02 | 34.04 |
| | | Ud Interruptor final de carrera | 2.0 | 1.27 | 2.54 |
| | | Ud Pack 5 unidades POLOLU DRV8825 | 1.0 | 7.5 | 7.5 |
| | | Ud Pack 50 90º 4 pin connector 2.54mm | 1.0 | 7.21 | 7.21 |
| | | Costes directos | | | 87.08 |
| | | Costes directos complementarios (2% herramientas) | 0.02 | 87.08 | 1.7416 |
| | | Coste total | | | 88.82 |

| | | | | |
|---|----------------------------------------------------------|-------|--------|---------------|
| 3 | Estructura | | | |
| | Ud Tarima madera machihembrada | 1.0 | 10.9 | 10.9 |
| | m ² Tablero contrachapado | 1.0 | 6.21 | 6.21 |
| | Ud Varilla roscada 3 mm 1 m | 1.0 | 0.72 | 0.72 |
| | Ud Pack 35 tornillos 4 x 20 mm | 1.0 | 1.13 | 1.13 |
| | Ud Pack 150 tuercas 3 mm | 1.0 | 1.65 | 1.65 |
| | h Graduado GITI | 10.0 | 35.0 | 350.0 |
| | h Colaborador | 5.0 | 15.0 | 75.0 |
| | Costes directos | | | 445.6 |
| | Costes directos complementarios (2% herramientas) | 0.02 | 445.6 | 8.9122 |
| | Coste total | | | 454.5 |
| 3 | Calibración | | | |
| 1 | Calibración fuente lumínica | | | |
| | Ud Fuente lumínica | 1.0 | 5.0 | 5.0 |
| | h Graduado en GITI | 5.0 | 35.0 | 175.0 |
| | Costes directos | | | 180.0 |
| | Costes directos complementarios (2% herramientas) | 0.02 | 180.0 | 3.6 |
| | Coste total | | | 183.6 |
| 2 | Calibración prototipo | | | |
| | h Graduado en GITI | 10.0 | 35.0 | 350.0 |
| | Costes directos | | | 350.0 |
| | Costes directos complementarios (2% herramientas) | 0.02 | 350.0 | 7.0 |
| | Coste total | | | 357.0 |
| 4 | Investigación y búsqueda de información | | | |
| 1 | Investigación | | | |
| | h Graduado en GITI | 100.0 | 35.0 | 3500.0 |
| | Costes directos | | | 3500.0 |
| | Costes directos complementarios (2% herramientas) | 0.02 | 3500.0 | 70 |
| | Coste total | | | 3570.0 |

Si se quiere consultar el archivo Excel del presupuesto puede consultarse el siguiente enlace:

https://drive.google.com/open?id=0B_B1eCxfBX73dIlkTjhjG1WZGM