

# Desenvolupament d'una lluminària multicolor amb LEDs amb sistema de control intel·ligent

Autor: Adrià Huguet Ferran

Tutor: Blas Garrido

Curs acadèmic: 2013-2014

Màster en Energies  
Renovables i  
Sostenibilitat Energètica



# Índex

0. Abstract.....	3
1. Introducció.....	4
2. Colorimetria.....	5
2.1. Cos negre i colors.....	5
2.2. CIE 1931: planckian locus.....	7
2.3. CRI i eficiència d'una font lluminosa.....	9
3. Sistema de control de la lluminària.....	12
3.1. Arduino ATmega328.....	12
3.2. LED.....	12
3.3. Construcció, disposició i connexions.....	13
3.4. Software de control.....	16
3.4.1. MATLAB GUI.....	18
3.4.2. Aplicació mòbil per a sistemes Android.....	19
4. Estudi energètic i d'eficiència.....	21
4.1. Flux lluminós.....	23
4.2. Potència consumida.....	24
4.3. CRI.....	25
4.4. Eficiència.....	27
5. Estudi econòmic.....	29
6. Conclusions.....	32
6.1. Avantatges.....	32
6.2. Inconvenients.....	32
6.3. Possibles millores.....	33
6. Bibliografia.....	35
Annex.....	37
Annex A. Material i especificacions.....	37
A1. Material utilitzat.....	37
A2. Especificacions dels LED.....	37
A3. Especificacions de l'Arduino Nano ATmega 328.....	38
Annex B. Relacions matemàtiques.....	39
B1. Generació del planckian locus al GUI del MATLAB.....	39
Annex C. Programació.....	41
C1. Càlcul de la intensitat dels diferents LED en funció de la temperatura de color d'un cos negre (MATLAB).....	41
C2. Generació del planckian locus (Python 2.7.).....	63
C3. Interfície gràfica desenvolupada en MATLAB (MATLAB).....	64
C4. Instruccions per descarregar l'aplicació per a dispositius Android.....	70
C5. Comunicació de l'Arduino amb sistemes Android (C).....	71
C6. Càlculs teòrics d'eficiència (Python 2.7.).....	73
Annex D. Resultats.....	75
D1. Simulació realitzada per ajustar els LED a l'espectre del cos negre ideal.....	75
D2. Simulació realitzada per obtenir l'eficiència teòrica del sistema.....	76
Annex E. Fotografies del prototip.....	77
Annex E1. Microcontrolador, plaques de connexions i LED.....	77
Annex E2. Fotografies del sistema en funcionament.....	78



## 0. Abstract

The aim of this project is to develop a system able to reproduce the visible light emitted by a blackbody in a temperature between 1000 and 8000K using the LED technology. This light has an ideal color rendering index (CRI), so it can be used for any application. The traditional and obsolete incandescent light bulbs, acting as a blackbody radiator, offer a very poor efficiency converting the electricity into light, producing only 10-20lm/W.

The developed smart system is controlled by an Arduino Nano equipped with an ATmega 328 microcontroller. It contains 6 different LED types: red, cyan, green, blue, warm white and neutral white. This selection of colors covers almost the whole visible band of the electromagnetic spectrum, allowing to fit the LED global light to the blackbody curve.

The color temperature can be selected through two different graphical user interfaces (GUI) and they are programmed in MATLAB and App Inventor, a programming language for Android devices developed by the Massachusetts Institute of Technology (MIT). On these interfaces the user can control the main parameters of the light, such as the color temperature and the global intensity. The intensity of the individual LED, the luminous efficacy and the CRI of each selection is given for informational purposes. Furthermore, the GUI developed in MATLAB shows a graphic with the color diagram and the Planckian locus plotted inside, changing the position of a pointer for each color temperature selected. Moreover, the MATLAB GUI requires a USB connexion while the Android GUI needs the bluetooth technology.

This microcontroller used can only provide 40mA of electrical intensity per pin. It causes several restrictions on the prototype, resulting in a low luminous flux (10-30lm) and a greater efficiency (30-50lm/W) than a halogen light bulb but smaller than a CFL. These issues can be solved changing the circuit, adding transistors and an external power source able to provide 700mA per pin, the nominal intensity of the used LED. This improvement increases the efficiency to values between 50 and 60lm/W, the typical performance of the CFL light bulb.

Finally, the economical study shows that the cost of the prototype is approximately 180€, but an industrial process can reduce the price more than 130€, giving a final cost of less than 50€ per unit. This can be achieved due to the reduction of the connections board cost and the use of the microcontroller instead of the entire Arduino.

# 1. Introducció

L'espectre solar s'ajusta gairebé a la perfecció a l'espectre del cos negre descrit per la llei de Planck de la mateixa manera que qualsevol altre objecte incandescent, com podrien ser les antigues bombetes ja obsoletes. La similitud entre la llum del sol i la de les bombetes incandescents fa que l'ull humà s'adapti molt bé a la llum d'aquestes, amb un índex de reproducció cromàtica ideal. El problema de les bombetes incandescents, per tant, no és la qualitat cromàtica de la seva llum, sinó la seva eficiència. Més d'un 95% de l'energia utilitzada es perd en forma de calor, de manera que com a màxim un 5% de l'energia s'emet en forma de llum visible.

L'objectiu d'aquest projecte és reproduir la llum emesa per un cos negre a una certa temperatura a través d'una tecnologia molt eficient: els LED (*Light Emitting Diode*). El color dels LED depèn principalment del material semiconductor utilitzat en la construcció, de manera que es poden aconseguir combinacions de LED amb una repartició espectral gairebé completa. Això permet, amb l'ajuda d'un microcontrolador i dels diferents elements que seran presentats, assolir el control i la reproducció de la llum emissora d'un cos negre a qualsevol temperatura i intensitat.

La melatonina, una hormona produïda a la glàndula pineal, participa en una gran varietat de funcions biològiques com podria ser la regulació de la pressió sanguínia o la reproducció estacional. Està caracteritzada per la variabilitat en el cicle de 24 hores (el cicle circadià) i la seva resposta sota canvis en la il·luminació ambiental [1] [2]. El sistema lluminós desenvolupat en aquest projecte permet realitzar, a voluntat de l'usuari, canvis del color de la llum i simular la llum d'un dia complet. Això pot ser interessant de cara a la regulació del ritme circadià, ja que la dependència de la melatonina en la llum ambiental pot ser controlada a través d'aquest sistema.

Al llarg d'aquest treball, doncs, s'exposa la base física necessària i s'expliquen tots els processos que han estat involucrats en el desenvolupament de la lluminària multicolor amb LED amb sistema de control intel·ligent: des dels circuits elèctrics i els components involucrats fins a tot el programari necessari pel correcte funcionament.

A l'annex E es poden trobar fotografies de les diferents parts que formen el prototip construït així com el propi sistema en funcionament.

## 2. Colorimetria

### 2.1. Cos negre i colors

La radiació tèrmica és la radiació electromagnètica que emet un cos pel fet d'estar a una certa temperatura. D'una forma anàloga, els cossos absorbeixen radiació tèrmica provinent de l'entorn. En cas que un cos estigui més calent que el seu entorn tendirà a refredar-se, ja que la potència emesa serà superior a la potència absorbida. Quan aquestes potències són iguals es parla d'equilibri termodinàmic. A temperatura ambient els cossos són visibles per la llum que reflecteixen, però a altes temperatures emeten llum visible, sent lluminosos per si mateixos. En general, la forma de l'espectre d'emissió d'un cos depèn de la seva temperatura i de la seva composició. Hi ha, però, una classe de cossos on l'espectre d'emissió depèn únicament de la temperatura. Aquests són coneguts amb el nom de cossos negres.

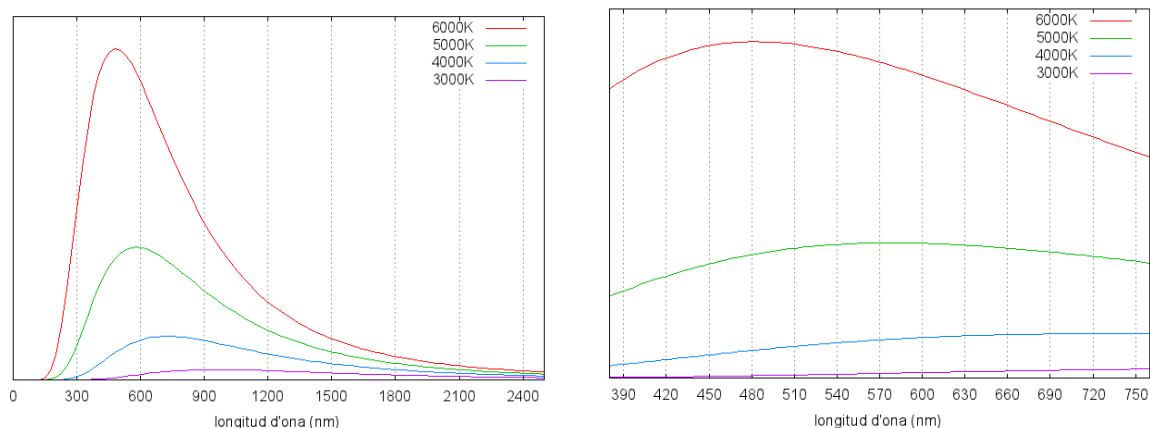
El cos negre és aquell que absorbeix tota la radiació tèrmica incident a la seva superfície, amb un coeficient d'absorció igual a 1 i un coeficient de reflexió igual a 0. Perquè un cos negre emeti llum visible ha d'estar a una temperatura superior a  $10^3\text{K}$  mentre que a temperatura ambient emet principalment radiació infraroja. El cas clàssic que es fa servir d'exemple d'un cos negre és una cavitat amb un petit orifici d'entrada. Un raig que entri per l'orifici té una probabilitat molt petita de tornar a sortir sense ser absorbit per les parets interiors de la cavitat, i per altra banda, la radiació que surt a través del forat s'ha hagut de generar a l'interior de la cavitat.

L'equació que descriu la radiació electromagnètica emesa per un cos negre en equilibri termodinàmic a una temperatura donada és la següent:

$$B_{\lambda}(T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1} \quad (1)$$

on  $B$  és la radiància espectral,  $T$  és la temperatura absoluta,  $h$  és la constant de Planck,  $c$  és la velocitat de la llum al buit,  $\lambda$  és la longitud d'ona de la radiació emesa i  $k_B$  és la constant de Boltzmann.

A continuació és representa la radiació electromagnètica emesa per un cos negre a diferents temperatures:



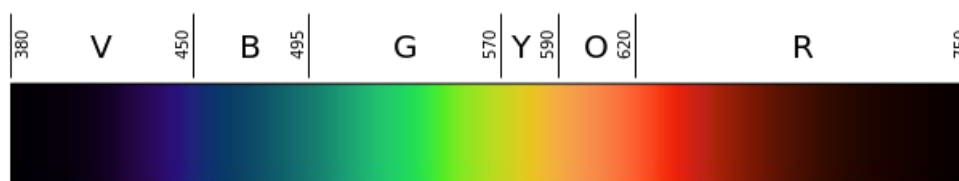
**Figura 1.** Radiació emesa per un cos negre a diferents temperatures. (a) de 0 a 2500nm (b) en el rang del visible.

A la pràctica, la llum emesa per les estrelles o els filaments de les bombetes incandescent és modelitzada com un cos negre, de manera que el seu espectre d'emissió és molt semblant al presentat.

L'espectre visible és la porció de l'espectre electromagnètic que pot ser detectat per l'ull humà. Els límits exactes de l'espectre visible no són del tot clars, però s'accepta que correspon a les longituds d'ona aproximadament situades entre 380nm i 740nm. Els diferents colors que es poden percebre es classifiquen de la següent forma:

**Taula 1.** Bandes espectrals dels diferents colors [3].

Color	Violeta	Blau	Cian	Verd	Groc	Taronja	Vermell
$\lambda$ (nm)	380-430	430-500	500-520	520-565	565-580	580-625	625-740



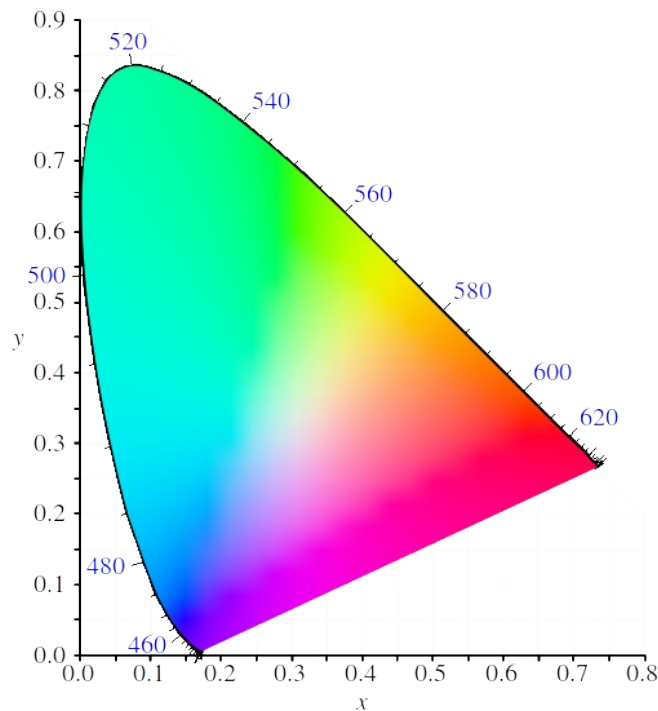
**Figura 2.** Colors i longituds d'ona [4].

Els cons són un dels dos tipus de cèl·lules fotorceptores existents a la retina de l'ull humà. Són els responsables de la visió en color i de la sensibilitat cromàtica. N'hi ha de tres tipus diferents i cadascun dels quals se n'encarrega de captar la llum en una determinada franja espectral. D'aquesta manera, amb la mescla dels tres estímuls provinents dels tres tipus diferents de cons es pot descriure qualsevol color.



## 2.2. CIE 1931: planckian locus

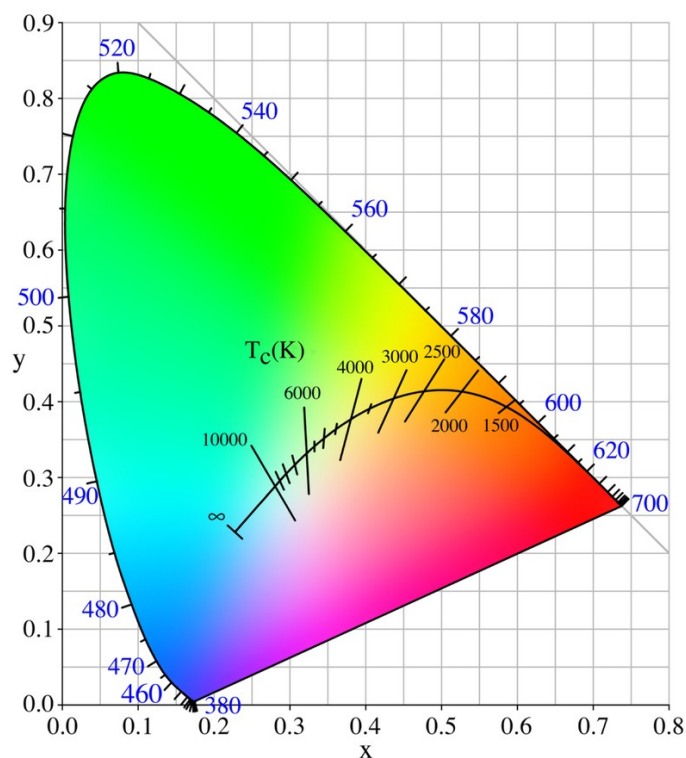
En un intent de quantificar els colors que es poden percebre, a l'any 1931 la *International Commission on Illumination* va crear el diagrama cromàtic CIE, que conté tots els colors que una persona estàndard pot resoldre. Serveix de referència a l'hora de quantificar un color ja que utilitza les corbes de sensibilitat dels tres tipus de cons existents a l'ull humà, permetent establir unes coordenades concretes  $(x, y)$  per un color qualsevol.



**Figura 3.** Diagrama cromàtic del model de color CIE 1931 [5].

Una font de llum blanca és una font de llum que avarca una àmplia banda òptica de l'espectre visible, normalment més de 100nm. Es defineix la temperatura de color d'una font blanca de llum com la temperatura d'un cos negre ideal que té la mateixa localització cromàtica que la font de llum blanca considerada.

A mesura que un cos negre s'escalfa i comença a emetre radiació dins del rang del visible, el color generat es va desplaçant a través del diagrama cromàtic començant pel vermell profund i passant pel el taronja, un blanc càlid, un blanc neutre i blanc blavós. Aquesta corba traçada sobre el diagrama es coneix amb el nom de *planckian locus*.



**Figura 4.** Diagrama cromàtic amb el *planckian locus* representat [6].

### 2.3. CRI i eficiència d'una font lluminosa

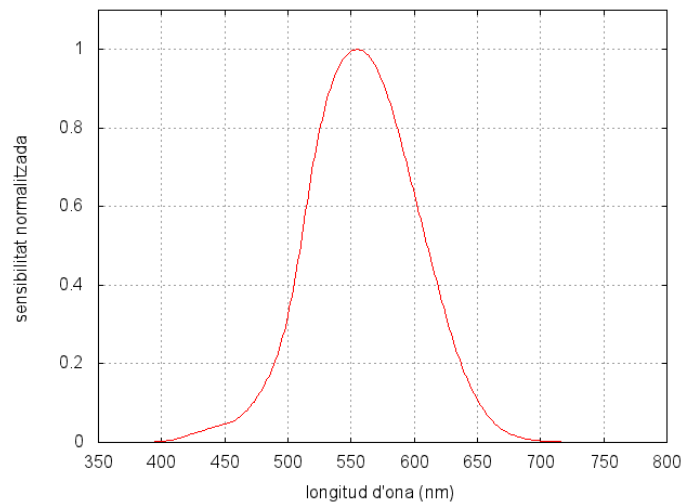
En colorimetria, una de les característiques més importants que se li atribueixen a la llum blanca és la capacitat de mostrar els colors reals dels objectes físics il·luminats. Aquesta propietat es coneix amb el nom de CRI (*Color-Rendering Index*) de manera que un CRI elevat, pròxim al màxim teòric (100), implica una molt bona reproducció dels colors. Per avaluar el CRI d'una font lluminosa es compara el canvi de color d'un objecte al ser il·luminat amb una llum de referència. A continuació es mostra una taula amb els CRI de diferents fonts lluminoses:

**Taula 2.** CRI de diferents fonts lluminoses [7].

Font lluminosa	CRI
Llum solar	100
Filament de wolframi (bombeta incandescent)	100
Fluorescent	60-85
LED basats en fòsfor (llum blanca)	60-90
Llum de vapor de sodi d'alta pressió	40
Llum de vapor de mercuri	20

En general, es considera que una font que sigui capaç d'assolir un CRI més gran de 85 és utilitzable per la gran majoria de les possibles aplicacions.

L'eficiència també és un factor molt important a l'hora de caracteritzar una font de llum. Es poden diferenciar, però, dos tipus d'eficiència diferents: una té en compte la potència elèctrica mentre que l'altra té en compte la potència radiativa emesa per la font lluminosa, que pot incloure longituds d'ona més enllà del rang del visible. Aquesta última també es coneix amb el nom d'eficàcia lluminosa. Normalment, si no s'expressa el contrari, l'eficiència serà la mesura del flux lluminós dividit per la potència elèctrica utilitzada. El flux lluminós és el flux radiatiu d'una font (el flux de radiació emès en totes les longituds d'ona) pesat per la corba de sensibilitat de l'ull humà i integrat per tot l'espectre, donant com a resultat la potència lluminosa percebuda. Es mesura en lúmens (*lm*).



**Figura 5.** Corba de sensibilitat de l'ull humà normalitzada en condicions fotòpiques [8].

El pic d'aquesta corba està situat als 555nm, corresponent al verd. Això significa que el nostre ull té respostes diferents per a les diverses longituds d'ona del rang del visible. Capta millor la llum verda que la llum vermella o la violeta, de manera que aquest factor afecta a l'eficiència d'una font lluminosa depenent del color que emeti.

Les eficiències de les diferents fonts lluminoses han anat millorant conforme la tecnologia ha anat progressant. Les antigues bombetes incandescents han estat substituïdes progressivament per noves formes d'il·luminació sovint basades en principis diferents. Actualment, els LED ofereixen grans possibilitats que fan pensar que serà una tecnologia que es consolidarà: treballen a baixa tensió, permeten un control acurat de la intensitat lumínica, són

susceptibles d'incorporar controls intel·ligents i, sobretot, tenen una eficiència força elevada i amb marge de millora.

**Taula 3.** Eficiència de diferents fonts lluminoses [9] [10] i [11].

<b>Font lluminosa</b>	<b>Eficiència (lm/W)</b>
Bombeta incandescent	10-20
Bombeta halògena	30
Fluorescent	Fins a 120
CFL	65
Llum de valor de sodi a baixa pressió	200
LED	Fins a 200

### **3. Sistema de control de la lluminària**

En aquest apartat s'exposa el material necessari pel desenvolupament de la lluminària, prestant especial atenció al microcontrolador utilitzat i a tot el programari involucrat en el procés.

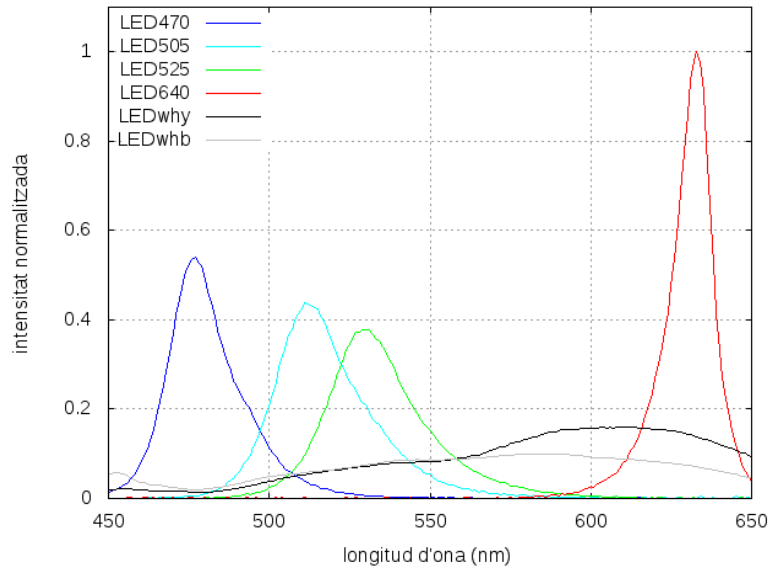
El material necessari pel desenvolupament de la lluminària i les especificacions dels diferents LED emprats es detallen a l'annex A1 i A2, respectivament.

#### **3.1. Arduino ATmega328**

Un microcontrolador és un petit ordinador connectat a un sol circuit integrat que consta d'un processador, una memòria interna i unitats d'entrada i sortida. En aquest cas s'ha fet servir un Arduino Nano basat en el microcontrolador ATmega328, de hardware i software lliure, ja que per les pretensions del projecte és força adient. Les prestacions més significatives d'un Arduino Nano es detallen a l'annex A3, juntament amb fotografies il·lustratives.

#### **3.2. LED**

Els LED és un dels elements clau per entendre el funcionament del sistema d'il·luminació. La seva correcta tria és crucial per tal d'oferir una bona reproducció dels diferents espectres i, forçosament, han de formar un grup que garanteixi una repartició suficientment àmplia com per abastir tot l'espectre visible. La figura 6 mostra els espectres mesurats dels diferents LED utilitzats. En aquestes mesures s'ha utilitzat l'espectròmetre *Ocean Optics USB2000+* i filtres de densitat òptica 2 pels LED vermell, cian, blau i verd; densitat òptica 1.3 pel blanc càlid; i dos filtres de densitat òptica 0.6 i 0.4 pel LED blanc neutre. Els filtres s'han utilitzat per disminuir el flux lluminós de les fonts per tal que l'espectròmetre pugui captar la llum sense sortir del rang de mesures. Posteriorment això s'ha tingut en compte en els càlculs de la simulació multiplicant pels factors corresponents.



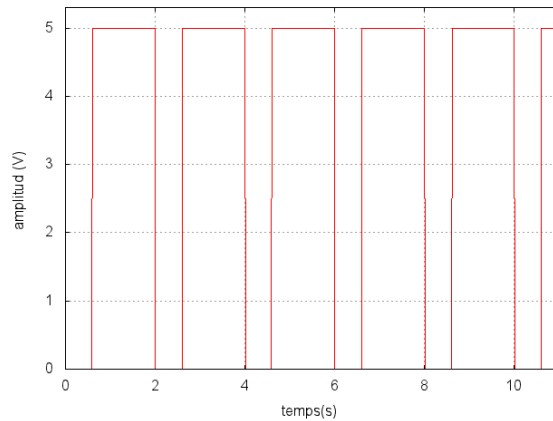
**Figura 6.** Espectres normalitzats dels LED utilitzats i mesurats al prototip.

A més a més cal tenir en compte les limitacions del microcontrolador, que només és capaç de donar 40mA d'intensitat per cada sortida i el control analògic del senyal (PWM) està disponible només en 6 canals. Això fa que els LED no treballin a potència nominal i la seva lluminositat potencial queda disminuïda de forma molt notòria. Paral·lelament, poder utilitzar només 6 LED diferents implica una molt bona elecció d'aquests per tal que es pugui reproduir els diferents espectres amb la màxima fidelitat.

Un altre factor que cal tenir present és la disposició dels LED i l'element difusor de la llum. És molt important que la llum surti correctament mesclada, de forma isotròpica, ja que sinó el seu espectre diferiria en funció de l'angle. Per tal de garantir un correcte funcionament, els LED s'han situat en forma circular de manera que els diferents colors no quedin agrupats a una regió concreta, tal i com es pot veure a la figura A3 de l'annex E1. La difusió la duu a terme un element esfèric semitranslúcid de vidre, que per proposicions demostratives compleix a la perfecció.

### 3.3. Construcció, disposició i connexions

L'Arduino disposa de 6 sortides de tipus PWM (*Pulse-Width Modulation*). Aquest tipus d'ona es caracteritza per permetre el control de l'amplada d'un pols elèctric periòdic i de forma quadrada. La freqüència del pols és molt superior a la freqüència de resolució de l'ull humà, sent utilitzable pel control de la intensitat lumínica de LED. L'amplitud de l'ona és de 5V, la diferència de potencial nominal dels diferents canals del microcontrolador.



**Figura 7.** PWM de període 2s, amplada d'1.3s i amplitud de 5V.

Per tal que tot funcioni correctament, els LED han d'anar connectats als pins D3, D5, D6, D9, D10 i D11 de l'Arduino, que són els que proporcionen aquest tipus de sortida. A més a més, donat que els LED tenen una resistència interna molt baixa cal limitar el corrent circulant mitjançant resistències addicionals. D'aquesta forma s'eviten intensitats molt elevades que podrien danyar el microcontrolador. Per cada LED diferent, doncs, es calcula la resistència necessària a través de:

$$R_i = \frac{V_{sortida} - V_{LED,i}}{I} \quad (2)$$

on  $V_{sortida}$  és el potencial del senyal de sortida (5V),  $V_{LED,i}$  és la caiguda de potencial de cada LED i  $I$  és la intensitat que circula per cada LED (40mA).

Així doncs, obtenim les següents resistències addicionals a incorporar a partir de l'equació (2):

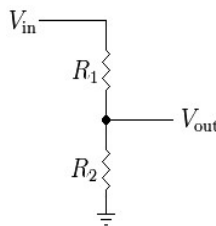
**Taula 4.** Resistències utilitzades per limitar el corrent.

Color del LED	Caiguda de potencial (polarització directa) (V)	Resistència addicional (ohms)
Blau	3.3	43
Cian	3.25	44
Verd	3.25	44
Vermell	2.9	53
Blanc càlid	2.81	55
Blanc neutre	3	50

El dispositiu Bluetooth, per altra banda, es comunica amb el microcontrolador a través dels pins RX0 i TX1 de l'Arduino. Aquests dos pins són els encarregats de rebre i transmetre dades, ja sigui amb un ordinador extern a través de la connexió USB o bé qualsevol altre dispositiu mitjançant la connexió per cables.

En aquest cas el dispositiu Bluetooth utilitzat és el *JY-MCU Bluetooth to UART Wireless Serial Port Module* [16]. Necessita una alimentació de 5V, cosa que el fa ideal per utilitzar-lo juntament amb l'Arduino, però l'entrada de dades ha d'estar connectada a 3.3V. Per salvar aquesta dificultat s'utilitza un divisor de tensió per tal de disminuir el potencial d'entrada.

Un divisor de tensió consta de dues resistències connectades en sèrie tal i com es mostra al següent esquema:



**Figura 8.** Divisor de tensió [17].

$V_{in}$  és el potencial sortint de l'Arduino i  $V_{out}$  és el potencial desitjat. La relació entre el potencial  $V_{in}$  i  $V_{out}$  és:

$$V_{in} = I \cdot (R_1 + R_2)$$

$$V_{out} = I \cdot R_2$$

$$\frac{V_{out}}{V_{in}} = \frac{R_2}{R_1 + R_2} \quad (3)$$

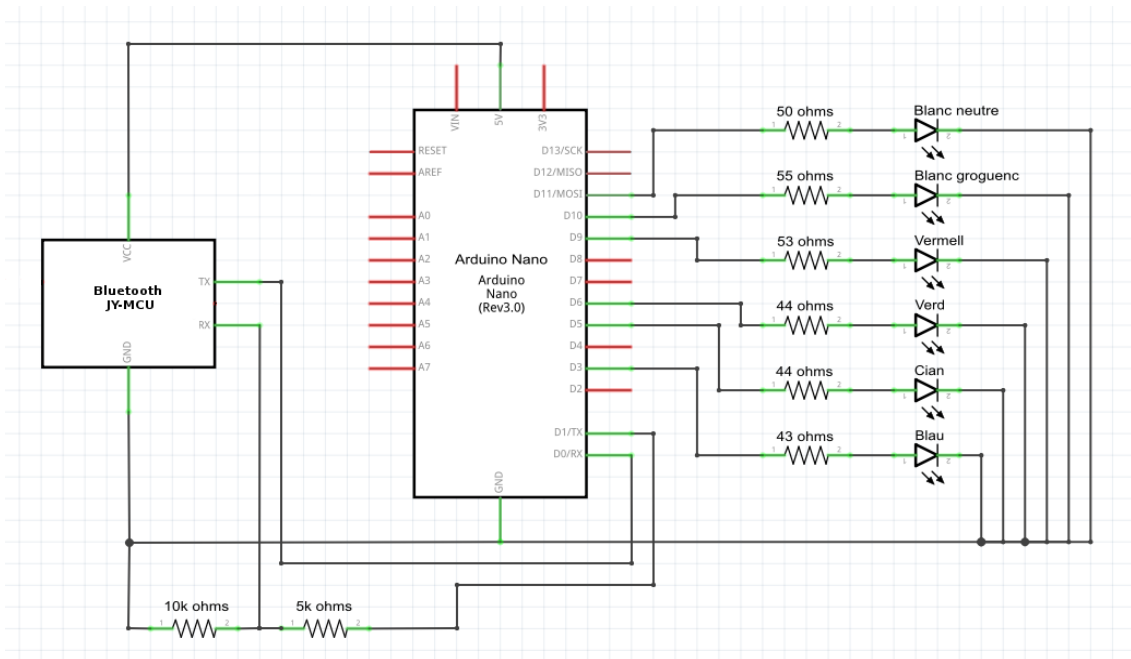
Per tal que  $V_{out}$  sigui 3.3V s'ha d'escollir el valor de les resistències segons l'equació (3) de manera que:

$$\frac{V_{out}}{V_{in}} = \frac{3.3V}{5V} = \frac{R_2}{R_1 + R_2} \rightarrow \frac{R_1}{R_2} \simeq 0.52$$

Per limitar el corrent s'escullen uns valors alts de les resistències. Per aquest cas,  $R_1 = 5k\Omega$  i  $R_2 = 10k\Omega$ .



L'esquema del circuit sencer, incloent l'Arduino, els LED i el dispositiu Bluetooth es detalla a continuació:



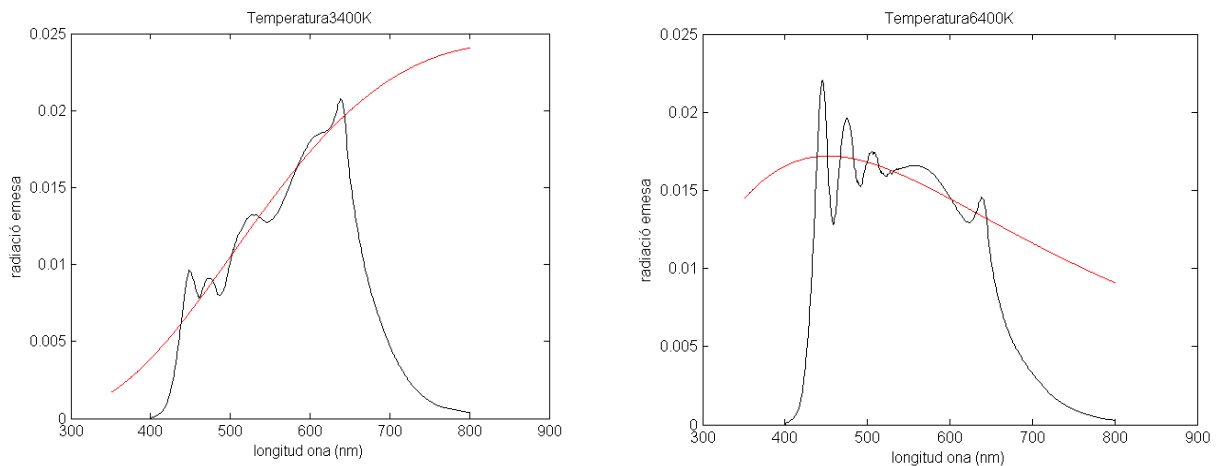
**Figura 9.** Esquema del circuit utilitzat. Per generar-lo s'ha utilitzat el software Fritzing [18] i Gimp[19].

### 3.4. Software de control

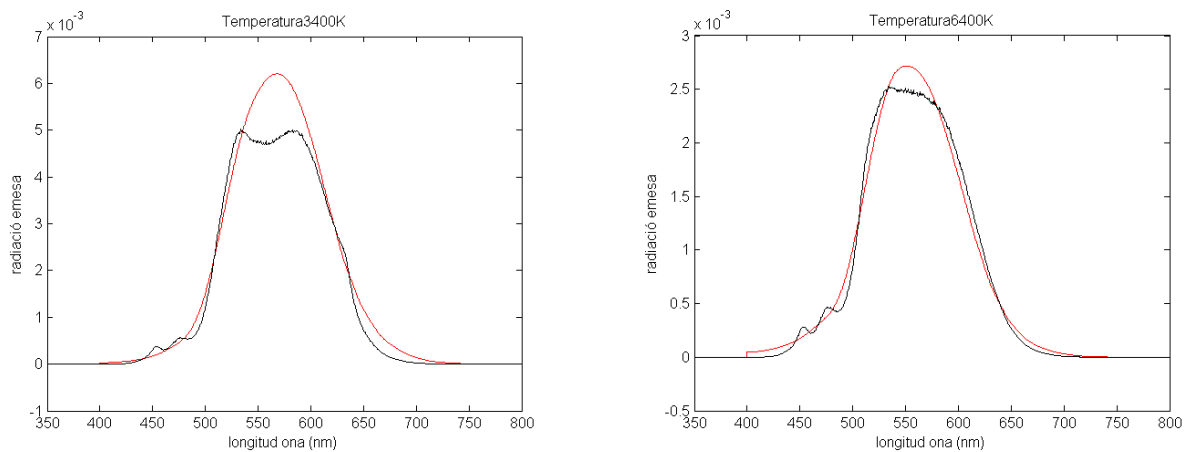
Primerament s'ha desenvolupat un programa escrit en MATLAB [20] que utilitza l'espectre dels LED descrits a l'apartat 3.2. i els ajusta a un espectre de cos negre a una temperatura donada.

Cada LED pateix una variació de la intensitat de corrent  $i$ , conseqüentment, una variació de la intensitat lumínica d'emissió. Així doncs, a través del control dels 6 LED diferents, el programa és capaç de simular la intensitat de corrent necessària per cada LED per tal que, en conjunt, s'ajustin a l'espectre d'emissió d'un cos negre a diferents temperatures. Cal notar que el microcontrolador no ofereix canvis en la intensitat sinó en l'amplada de l'ona PWM, però això no és un impediment a l'hora de controlar la intensitat lumínica.

Aquesta simulació parteix de l'espectre d'un cos negre ideal a una temperatura de 1000K  $i$ , en passos de 100K, arriba fins als 8000K. Els resultats es guarden en forma de matriu que posteriorment serà llegida pels diferents entorns gràfics que permeten una comunicació entre els resultats i l'Arduino, evitant la realització de repetitius càlculs i el lent processament de la informació. La matriu de resultats es pot trobar a l'annex D1.



**Figura 10.** Ajust dels diferents LED a l'espectre visible d'un cos negre a (a) 3400K i (b) 6400K.



**Figura 11.** Ajust dels diferents LED a l'espectre visible d'un cos negre multiplicat per la sensibilitat de l'ull humà a (a) 3400K i (b) 6400K.

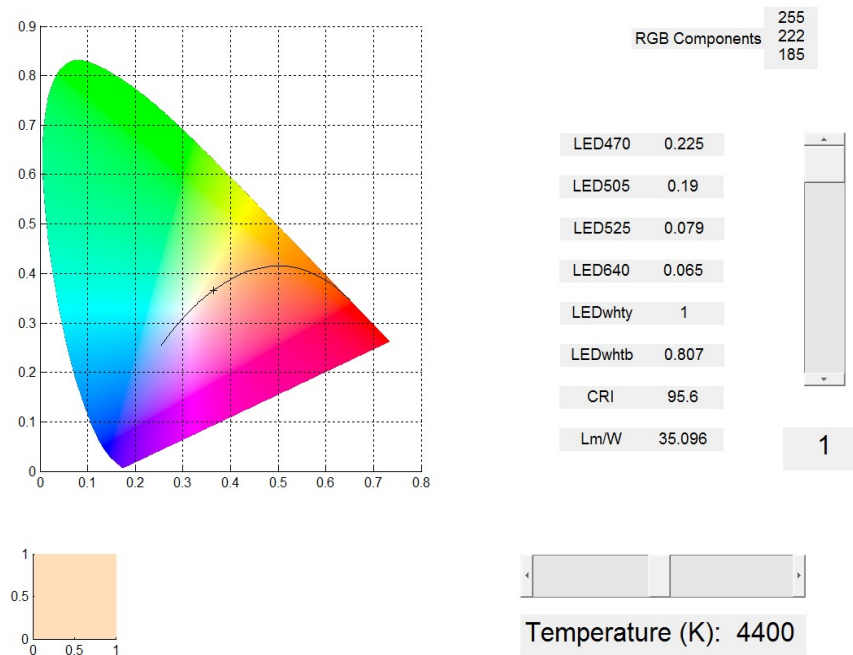
D'aquesta manera s'aconsegueix simular la llum que emetria un cos negre a una determinada temperatura a través dels 6 LED diferents, que són regulats a través del microcontrolador. El fet que el sistema estigui controlat per LED implica una alta eficiència a l'hora de transformar electricitat en llum i el fet que la llum emesa reproduïx l'espectre d'un cos negre dins del rang del visible dóna com a resultat un CRI molt elevat. A l'annex C1 es pot trobar el codi font del programa utilitzat.

Per tal de poder tenir un control acurat del sistema d'il·luminació intel·ligent s'han tingut en compte dues possibilitats diferents. Per una banda s'ha realitzat una GUI (*Graphical User Interface*) programada en MATLAB que permet canviar els paràmetres característics del sistema

a través d'un cable USB i, per altra banda, s'ha desenvolupat una aplicació per dispositius Android que permet un control sense fils mitjançant la tecnologia Bluetooth.

### 3.4.1. MATLAB GUI

A través de la taula generada pel programa que s'encarrega de determinar la intensitat individual de cada LED s'ha creat una interfície gràfica que permet obtenir un control de la temperatura de color i la intensitat del conjunt de LED. Aquesta, a més a més, mostra per pantalla el *planckian locus* superposat al diagrama cromàtic CIE 1931 amb la posició actualitzada de la temperatura de color escollida, les coordenades RGB de cada color, el CRI, l'eficàcia lluminosa en lm/W de la llum generada (no té en compte l'eficiència elèctrica dels LED) i les intensitats normalitzades que circulen pels diferents LED.



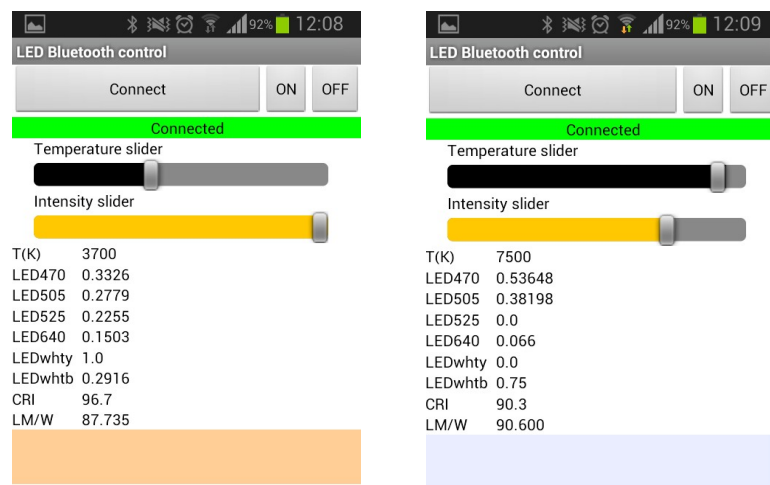
**Figura 12.** Interfície gràfica creada en MATLAB.

Per tal que aquesta interfície gràfica funcioni correctament i sigui capaç de comunicar-se amb el microcontrolador encarregat de fer funcionar el sistema és necessari establir una connexió, via USB, entre l'Arduino i l'ordinador utilitzat. Això suposa una limitació de les possibilitats del sistema, ja que implica la necessitat de posseir un ordinador amb el sistema operatiu Windows o una llicència MATLAB funcional a una distància suficientment curta del sistema lluminós. A més a més és necessari instal·lar el paquet addicional ArduinoIO per permetre una comunicació entre l'Arduino i l'ordinador, d'accés lliure i gratuït [21].

A l'annex B1 s'hi pot trobar una descripció detallada de com s'ha fet la representació del *planckian locus* sobre el diagrama CIE 1931 i els canvis de coordenades utilitzats. De la mateixa manera, a l'annex C2 es pot trobar el programa que s'ha utilitzat per generar els diferents punts de la corba, escrit en Python 2.7 [24], mentre que a l'annex C3 es pot trobar el codi font de la interfície gràfica desenvolupada en MATLAB.

### 3.4.2. Aplicació mòbil per a sistemes Android

Per tal de superar la limitació de programari i d'espai ja contemplades en la GUI desenvolupada en MATLAB s'ha desenvolupat una aplicació simplificada per sistemes Android que permet un control, de la mateixa manera, dels diferents paràmetres del sistema lluminós. Aquesta aplicació funciona de forma similar que l'anterior però sense mostrar el gràfic per pantalla degut a les dificultats tècniques que això suposaria. A estat desenvolupada a través de l'*App Inventor* [25], una aplicació web gestionada pel MIT (*Massachusetts Institute of Technology*) que permet crear aplicacions per a sistemes Android basat en el llenguatge de programació *Scratch* [26].



**Figura 13.** Interfície gràfica creada per dispositius Android.

Inicialment es carrega la matriu generada pel programa que determina la intensitat individual de cada LED. Seguidament, per a cada temperatura de color escollida, s'envia una cadena de caràcters (*string*) mitjançant la tecnologia Bluetooth a l'Arduino, que separa i filtra els diferents elements de la cadena. S'ha triat aquest procediment degut a que és la manera més fàcil d'enviar l'estat dels 6 LED en una sola variable. Això suposa una càrrega addicional de càlculs

pel microcontrolador, però tècnicament és més fiable i la programació es simplifica enormement. Cal afegir, a més a més, que el microcontrolador té la capacitat de càlcul suficient com per afrontar moltes més exigències que les que s'utilitzen en aquesta aplicació, sent capaç de resoldre la cadena de caràcters enviada i emetre els diferents senyals rebuts a través l'aplicació mòbil sense cap inconvenient.

Finalment, l'aplicació mostra per pantalla la temperatura de color seleccionada, les intensitats de corrent dels diferents LED, el CRI i l'eficiència teòrica de cada una de les llums reproduïdes, a més a més del color teòric de la llum resultant, tal i com es pot veure a la figura 13.

Les instruccions per descarregar el codi font de l'aplicació per dispositius Android es poden trobar a l'annex C4. El programa de funcionament de l'Arduino per aquesta modalitat es pot consultar a l'annex C5.

## 4. Estudi energètic i d'eficiència

Un cop explicat detalladament el funcionament bàsic del sistema d'il·luminació intel·ligent, es procedeix a realitzar un petit estudi energètic i d'eficiència per poder-ho comparar amb una làmpada de taula tradicional. Els valors finals són òbviament orientatius, ja que es poden considerar moltes tecnologies diferents i el projecte desenvolupat també és susceptible de patir modificacions importants de disseny. A més a més cal destacar que els espectres dels LED utilitzats per fer aquest estudi són els donats pel fabricant, que per aquest propòsit i en aquesta aproximació es poden considerar perfectament vàlids. Tampoc s'ha tingut en compte el dispositiu Bluetooth, ja que és un element opcional i la seva inclusió no faria canviar les conclusions finals.

Primer de tot es procedeix a simular, a nivell teòric, el flux lluminós, la potència consumida, el CRI i l'eficiència del sistema per diferents temperatures de color. Com que per cada temperatura de color diferent s'utilitzen diferents LED, aquests paràmetres no seran constants. Les eficiències dels diferents LED utilitzats es poden consultar a la següent taula:

**Taula 5.** Eficiències teòriques dels diferents LED utilitzats segons les dades del fabricant a 350mA [12] [13].

Color	Eficiència (lm/W)
Blau	38
Cian	75
Verd	93
Vermell	75
Blanc càlid	93
Blanc neutre	107

En primera aproximació es pot dir que la caiguda de potencial de cada LED varia segons quin sigui el seu color d'emissió. No obstant, degut a l'existència de les resistències encarregades de limitar el corrent afegides juntament amb les resistències de seguretat que incorpora l'Arduino, la caiguda de potencial total per a cada canal és de 5V.

La intensitat que circula per cada LED, per contra, varia en funció de la temperatura de color seleccionada. Això es tradueix en una variació de la potència consumida per cada temperatura de color i cal tenir-ho en compte a l'hora de realitzar els càlculs d'eficiència.

Per calcular la intensitat que passa per cada canal s'utilitza la matriu de dades resultants de l'ajust dels LED amb el cos negre de l'annex D1. D'aquesta manera:

$$I_i = I_{norm,i} \cdot 40\text{mA}$$

on  $I_i$  és la intensitat que passa per cada canal,  $I_{norm,i}$  és la intensitat normalitzada que passa per cada canal i 40mA és la intensitat màxima que dona el microcontrolador.

La potència consumida per cada canal es calcula a través de:

$$P_{cons,i} = 5\text{V} \cdot I_i$$

on 5V és el potencial de caiguda de cada canal.

S'ha mesurat la potència que consumeix l'Arduino, que en funcionament és de 0.125W. Així doncs:

$$P_{TOTAL} = P_{arduino} + \sum P_{cons,i}$$

Per altra banda, el flux lluminós emès serà:

$$F_i = E_i \cdot P_{LED,i} = E_i \cdot V_{LED,i} \cdot I_i$$

$$F_{TOTAL} = \sum F_i$$

on  $E_i$  és l'eficiència de cada LED i  $V_{LED,i}$  és el potencial de caiguda de cada LED.

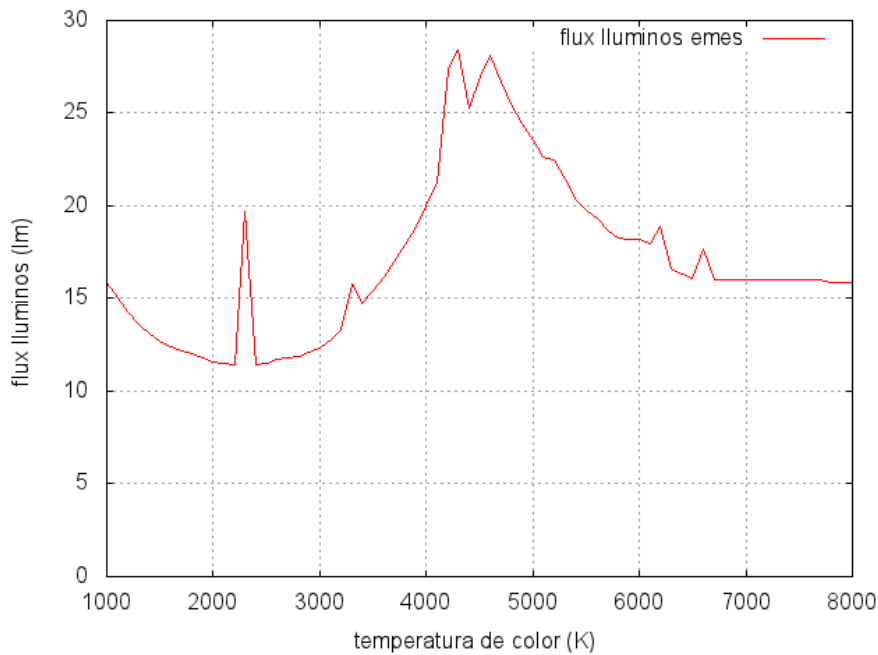
Finalment, per a calcular l'eficiència total de cada temperatura de color:

$$E_{TOTAL} = \frac{F_{TOTAL}}{P_{TOTAL}}$$

Així doncs, agafant les dades proporcionades a l'annex A2 i D1 i combinant-les amb les de la taula 5 amb els pertinents càlculs matemàtics ja presentats, s'arriba als resultats experimentals presentats als següents apartats.

La taula de resultats que han servit per generar les figures 14, 15, 16 i 17 es pot trobar a l'annex D2. El programa utilitzat per realitzar els càlculs es pot trobar a l'annex C6.

## 4.1. Flux lluminós



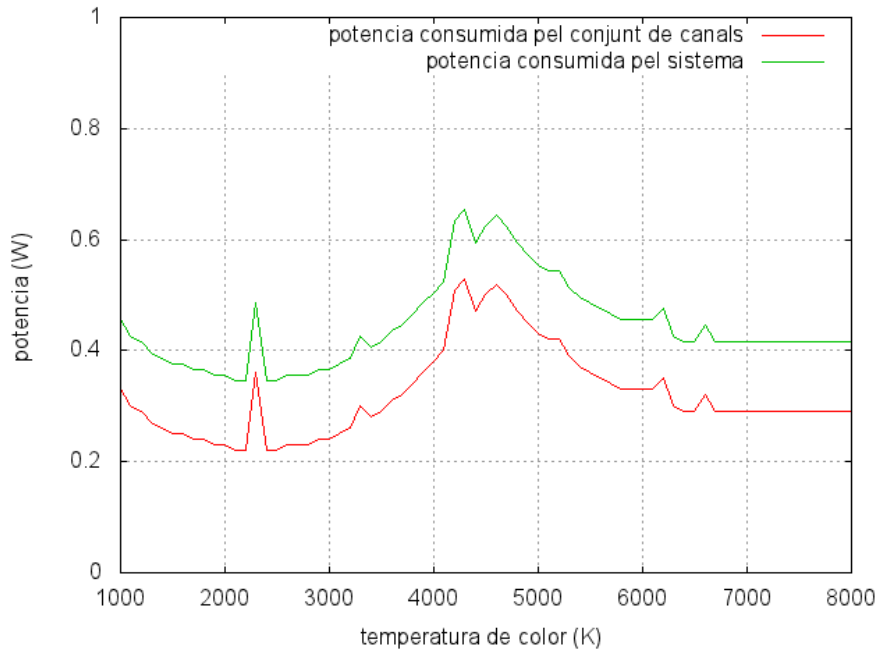
**Figura 14.** Flux lluminós emès pel sistema en funció de la temperatura de color.

La variació del flux lluminós en funció de la temperatura de color és deguda a la diferent resposta individual dels LED utilitzats. Per temperatures de color baixes s'utilitzen LED que emeten un flux lluminós inferior als que s'utilitzen per temperatures de color d'entre 4000 i 5000K. El mateix passa per temperatures de color elevades. El pic central es troba situat vora els 4500K i el rang de flux lluminós varia entre els 10 i 30lm, una quantitat bastant petita si es compara amb la capacitat total d'emissió dels LED utilitzats. L'explicació rau en la baixa intensitat utilitzada sortint del microcontrolador, 40mA. Aquests LED estan preparats per ser utilitzats a 700mA, podent emetre fins a uns 180lm pel cas del LED blanc neutre. No obstant, com que s'està treballant amb intensitats de 17.5 vegades inferiors a la nominal, el flux lluminós també es veu reduït d'una forma molt notòria.

Les diferents oscil·lacions que es poden observar, com ara el pic situat als 2300K, sorgeixen de la dificultat d'ajustar 6 LED a l'espectre d'un cos negre. Els LED blancs tenen un espectre amb dos pics i això comporta moltes inestabilitats a l'ajust, provocant un efecte en cadena i implicant altres LED, que han de modificar la intensitat per tal d'adaptar-se en conjunt.



## 4.2. Potència consumida



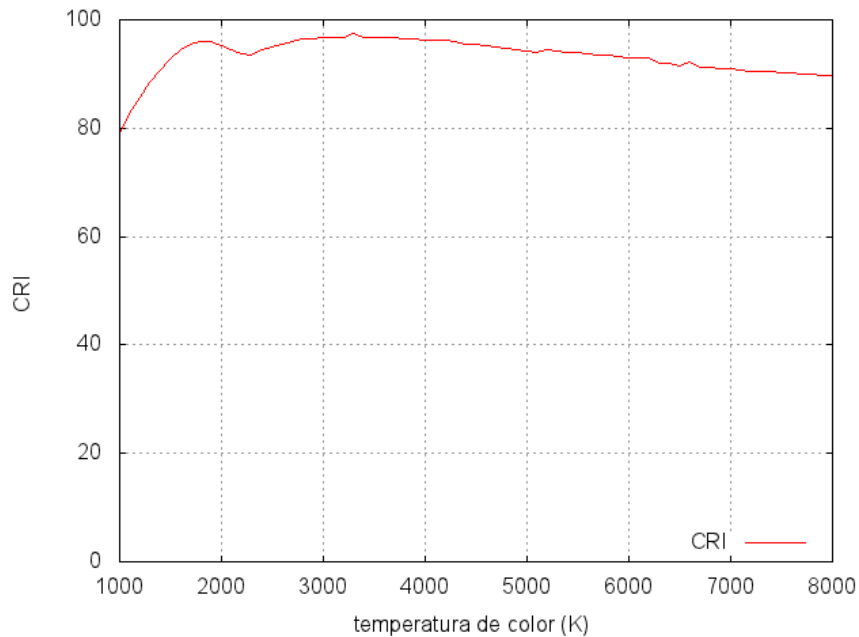
**Figura 15.** Potència consumida pel conjunt de canals i pel sistema lluminós en funció de la temperatura de color.

Tal i com es pot esperar la potència consumida no és constant en les diferents temperatures de color. El motiu és el mateix que en el cas anterior: la diferent resposta individual dels LED. En aquest cas la diferència entre la potència consumida pel conjunt de canals, on es troben els LED i hi ha l'emissió de llum, i la potència total del sistema és només la potència consumida per l'Arduino. Aquesta és considerada constant degut a que sempre ha de realitzar els mateixos processos independentment de la temperatura de color a la que treballi.

De forma similar amb el que succeïa amb el flux lluminós, es pot trobar un pic clarament diferenciat al voltant dels 4500K. Això és degut a que la potència consumida està directament relacionada amb el flux lluminós emès, ja que com més quantitat de llum emeti el sistema més energia consumirà.

La potència consumida pel sistema es troba entre els 0.3 i els 0.7W. Això és molt poca potència per un sistema lluminós, i novament s'explica per la baixa intensitat que circula per l'interior dels LED.

### 4.3. CRI

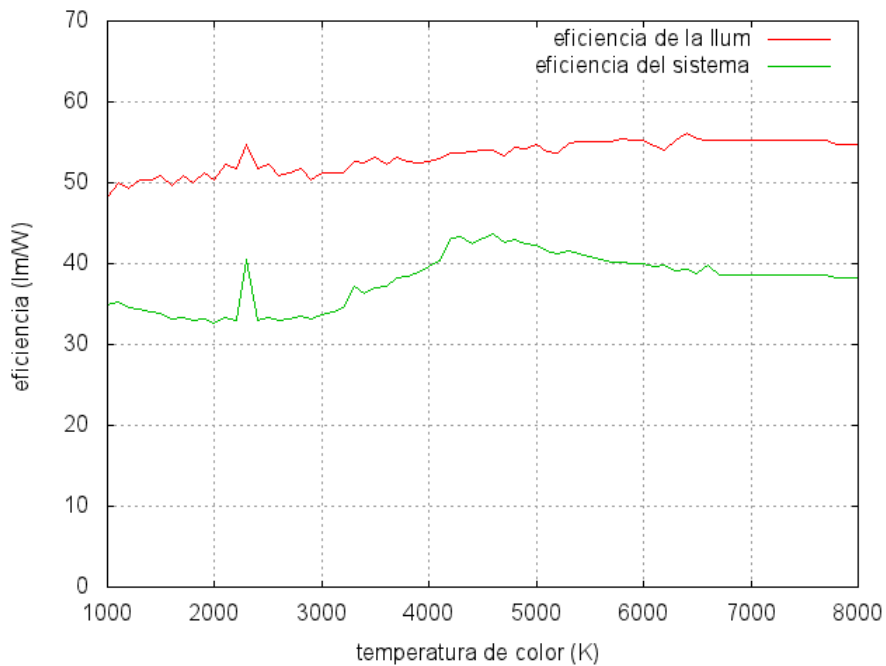


**Figura 16.** CRI del sistema lluminós en funció de la temperatura de color.

Tal i com s'observa a la figura 16, el CRI en funció de la temperatura de color descriu una corba bastant plana. Entre 1400 i 7700K es manté per sobre de 90 i el pic està situat als 3300K amb un valor de 97.5. Aquests valors són molt bons, cosa que permet utilitzar aquest sistema per qualsevol aplicació pràctica. A més a més s'aconsegueix, tal i com es pretenia des d'un primer moment, superar els índexs cromàtics de reproducció dels LED blancs individuals, situades al voltant del 90.

El baix rendiment del CRI de les temperatures de color més pròximes als 1000K s'expliquen per la forta presència del LED vermell combinat amb el blanc càlid. El LED vermell per si sol té un CRI molt baix i només combinant-lo amb altres colors és quan pot aportar una contribució positiva, però en aquest règim les opcions de combinació són molt baixes, de manera que el CRI global es veu fortament penalitzat. A temperatures més altes, per contra, els dos LED blancs agafen el protagonisme i passen a ser els dominants, de manera que el CRI es veu compensat positivament.

## 4.4. Eficiència



**Figura 17.** Eficiència del conjunt de canals i del sistema en funció de la temperatura de color.

En aquest cas és interessant analitzar per separat l'eficiència de la part encarregada de generar llum de l'eficiència de tot el sistema, ja que si s'analitza juntament amb la potència consumida es poden extreure conclusions importants i possibles solucions.

Per un banda es pot veure que l'eficiència de la part destinada a la il·luminació en funció de la temperatura de color és pràcticament constant. Això s'explica a través de les corbes observades de potència i flux lluminós i a través de l'eficiència dels diferents LED involucrats a cada temperatura. El pic que s'observava al voltant dels 4500K ha desaparegut perquè al voltant d'aquestes temperatures hi ha una forta utilització dels dos LED blancs, molt més eficients que la resta. La potència consumida augmenta en paral·lel amb el flux lluminós i, per tant, el quocient es manté constant.

Per altra banda, si es té en compte la contribució de tot el sistema, el pes de l'eficiència dels LED disminueix i la potència lluminosa ja no creix de la mateixa manera que ho fa la potència consumida, ja que el canvi relatiu és més petit si es considera tot el sistema. A més a més, la figura 17 també explica amb claredat l'anomalia situada als 2300K. En aquest punt, el programa d'ajust es veu obligat a realitzar una important aportació dels dos LED blancs, cosa que comporta un augment del flux lluminós i de l'eficiència.

Cal destacar que l'eficiència d'aquest sistema no és molt bona: es situa al voltant dels 40lm/W, un valor superior al d'una bombeta halògena però inferior al d'una bombeta CFL. Amb els LED utilitzats seria d'esperar obtenir eficiències molt més elevades, situades per sobre dels 60lm/W. Aquest fet s'explica amb la impossibilitat de tenir un control del potencial dels pins de sortida (fixat a 5V) i, novament, la baixa intensitat que circula pels LED, que fa que la potència consumida per aquests sigui comparable amb la potència consumida per l'Arduino. Si s'augmentés la intensitat fins a 700mA, la potència consumida per aquests també augmentaria un factor 17.5, de manera que la contribució de l'Arduino en la potència total consumida seria molt baixa i el sistema s'acostaria als 50-60lm/W que s'atribueix a la part encarregada d'il·luminar de la figura 17. Aquesta eficiència encara es pot veure incrementada ajustant el potencial de sortida dels pins a les caigudes de tensió dels diferents LED, ja que una bona part de l'energia es perd a les resistències addicionals afegides. Aquestes dues consideracions comportarien un aprofitament òptim de l'energia elèctrica per part dels LED, que podrien operar en condicions nominals i no es perdria energia ajustant el sistema. No obstant, treballar en aquest règim faria necessària la instal·lació de dissipadors tèrmics ja que sinó augmentaria la temperatura de treball dels LED degut a l'efecte Joule i el seu rendiment disminuiria dràsticament. Aquests dissipadors no suposen una despesa d'energia extra durant l'ús, però seria un element més a tenir en compte a l'hora de desenvolupar el projecte.

## 5. Estudi econòmic

En aquest apartat es realitza un estudi sobre un possible procés de producció industrial suposant una fabricació de 1000 unitats. S'ha tingut en compte que els elements han estat adquirits a l'empresa Farnell [27], un distribuïdor mundial de productes electrònics. El preu de les plaques de connexions, per contra, han estat estimades ja que les ha dissenyat i manufacturat personal del departament d'electrònica de la UB.

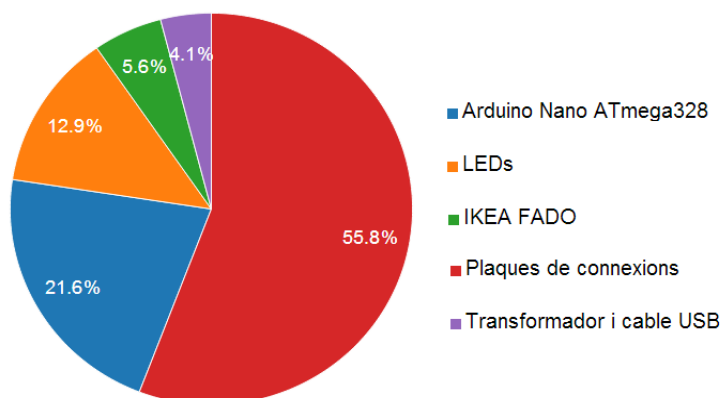
L'objectiu principal d'aquest estudi és trobar una cota superior del preu d'un possible procés de producció industrial d'un sistema com el presentat. Novament, el preu del dispositiu Bluetooth no ha estat inclòs a l'estudi pel fet de ser un element opcional. En cap cas, però, modifica l'anàlisi que es vol fer ja que el seu preu no és elevat.

La llista de preus dels diferents components que formen el sistema és la següent:

**Taula 6.** Preus i cost relatiu dels diferents elements del sistema lluminós.

Element	Preu	Cost relatiu (%)
Arduino Nano Atmega328	38.67	22
LED blau	4.48	3
LED cian	4.09	2
LED verd	3.95	2
LED vermell	3.29	2
LED blanc càlid	3.56	2
LED blanc neutre	3.82	2
IKEA FADO	9.99	6
Plaques de connexions	100	56
Transformador	1.3	1
Cable USB	5.99	3

Si es representen els preus relatius de la taula 6 en forma de gràfic:



**Figura 18.** Preus relatius del sistema lluminós.

Com es pot observar a la figura 18 els principals components que encareixen el sistema són les plaques de connexions i l'Arduino. El cost total és de gairebé 180€, dels quals 140€ corresponen a aquests dos elements. Es fa evident, doncs, pensar que si es vol optimitzar el sistema econòmicament el primer que cal fer és rebaixar el preu d'aquests dos components.

Pel que fa les plaques de connexions, un 80% correspon al disseny i el 20% restant a la manufactura. La part corresponent al disseny és una quantitat fixa ja que aquest serà el mateix per cadascuna d'elles i per tant la seva contribució al preu final per placa dependrà del nombre total de plaques fabricades.

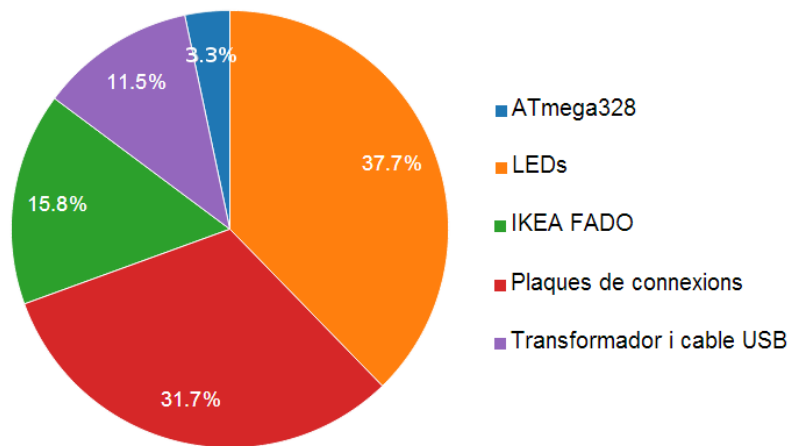
L'Arduino, per altra banda, és un projecte que no està pensat per ser utilitzat en aplicacions industrials. És molt útil per desenvolupar prototips perquè la seva arquitectura permet una continua variació dels circuits implicats sense necessitat de soldadures. A nivell industrial, però, caldria pensar que ha de ser substituït per un microcontrolador sense els avantatges addicionals que dóna l'Arduino. Aquest microcontrolador pot ser el mateix ATmega 328, amb un preu de mercat de 2.49€ [28].

Finalment cal afegir que el preu per unitat de cada component disminueix a mesura que s'incrementa el nombre d'unitats adquirides. D'aquesta manera, consultant els preus per 1000 unitats a l'empresa Farnell i suposant un estalvi del 30% en la resta de components:

**Taula 7.** Preu per unitat, estalvi respecte el prototip i cost relatiu dels diferents elements del circuit suposant una comanda de 1000 unitats.

Element	Preu per unitat (€)	Estalvi respecte el prototip (%)	Cost relatiu (%)
ATmega 328	1.47	94	3
LED blau	3.41	24	8
LED cian	2.62	36	6
LED verd	2.48	37	6
LED vermell	2.56	22	6
LED blanc càlid	2.57	28	6
LED blanc neutre	3.10	19	7
IKEA FADO	6.99	30	16
Plaques de connexions	14.08	30	32
Transformador	0.91	30	2
Cable USB	4.19	30	9
<b>TOTAL</b>	<b>44.39</b>	<b>75</b>	

Representant de nou els costos relatius:



**Figura 19.** Preus relatius del sistema lluminós per 1000 unitats.

Així doncs s'obté un sistema lluminós per menys de 50€ la unitat en primera aproximació.

Amb aquest nou escenari es pot observar com el preu dels LED guanya protagonisme mentre que el del microcontrolador queda fixat en un 3.3% del preu final. Aquest últim és un factor clau per conèixer la viabilitat del projecte ja que és el principal element que diferencia aquest sistema intel·ligent d'un convencional, juntament amb altres detalls econòmics menors com podria ser el disseny de les plaques de connexions o elements del circuit que per força s'haurien d'introduir, com ara resistències o capacitats. El preu de les plaques de connexions també pot patir una forta reducció en un procés d'industrialització, cosa que disminuiria encara més el cost final.

## 6. Conclusions

Després d'haver desenvolupat un primer prototip i haver realitzat un estudi teòric sobre la viabilitat energètica i econòmica del sistema d'il·luminació es poden treure una sèrie de conclusions rellevants.

### 6.1. Avantatges

- En primer lloc cal destacar la singularitat d'aquest sistema. Actualment importants empreses del món de la il·luminació estan començant a desenvolupar lluminàries semblants però que ofereixen menys oportunitats. La majoria de sistemes existents es basen en LED tricolor RGB, més senzills d'implementar però que no són capaços de reproduir espectres amb fidelitat. Amb l'addició de més LED es poden aconseguir fonts lluminoses amb un CRI pròxim a 100 amb una eficiència de 50-60lm/W si es disposa dels components adequats.
- Ofereix una gran versatilitat, podent ser utilitzat en moltes aplicacions diferents, com podria ser un llum de taula, una llanterna, un focus per una galeria d'art, il·luminació per biblioteques o hospitals, etc.
- El cost de producció, en contra del que es podria esperar, no difereix de forma notòria del d'una lluminària amb la mateixa potència ja que el preu del microcontrolador és reduït.
- La durabilitat dels LED en comparació amb les bombetes incandescents o halògenes és molt superior i consumeixen molta menys electricitat [29] [30], per la qual cosa la forta inversió inicial queda compensada amb un cost de funcionament molt petit. Malgrat tot, com que la llum emesa depèn del conjunt de LED, el mal funcionament o el desgast d'un sol d'ells provocaria la necessitat de substituir tot el grup, suposant un cost addicional.

### 6.2. Inconvenients

Per contra, el prototip també inclou una sèrie de factors limitants que poden ser solucionats modificant el sistema.

- La baixa eficiència del prototip és sens dubte el principal problema ja que és la magnitud que, des d'un principi, s'intentava maximitzar. Malgrat que l'eficiència del prototip és superior a la d'una bombeta halògena, és molt més baixa de la que es podria esperar per un sistema basat en la tecnologia LED. Com ja s'ha explicat, això és degut a la baixa intensitat



que proporciona el microcontrolador per cada canal. D'aquesta manera, els LED no poden funcionar a potència nominal i la contribució en el consum per part del microcontrolador és molt elevada. Una possible solució per tal de superar aquest contratemps seria a través de la incorporació de transistors que controlessin una font externa al sistema capaç de proporcionar el voltatge i la intensitat necessària per tal de garantir un funcionament a potència nominal dels LED.

- Un altre inconvenient és el programari involucrat per a la utilització del sistema. Les dues interfícies gràfiques de control impliquen la necessitat de posseir un ordinador equipat amb el sistema operatiu Windows o una llicència MATLAB, que no són gratuïts, o bé un dispositiu Android. Aquesta dependència externa pot suposar un cost econòmic addicional, fent la lluminària menys atractiva comercialment. Malgrat tot, actualment existeixen llenguatges de programació multiplataforma d'alt nivell amb els quals es poden escriure programes equivalents creant les interfícies gràfiques pertinents, com per exemple el Python. Una altra opció que evita la utilització de dispositius externs és la inserció d'un potenciòmetre que permeti una variació d'una resistència en un canal diferent. La caiguda de potencial pot ser mesurable pel microcontrolador i d'aquesta manera es poden associar successius valors de les resistències a diferents temperatures de color d'una manera semblant al que fa un control del volum en un aparell de reproducció d'àudio.

### 6.3. Possibles millores

Per acabar s'hauria de destacar les possibles millores a incorporar en un futur. Aquest és un projecte inacabat i que constantment pot anar millorant, ja sigui amb noves funcionalitats o a través de la substitució de tecnologia més innovadora.

- El primer que caldria fer és augmentar, com s'exposa a l'apartat 4.1. i 6.2., la intensitat que circula pels LED per tal de millorar l'eficiència i la lluminositat total. Aquesta millora no suposa un canvi radical del sistema ja que la programació del microcontrolador és exactament la mateixa i tan sols s'hauria de canviar el circuit introduint transistors i una font externa, que podria ser la xarxa elèctrica.
- A nivell de millorar l'ajust dels diferents LED a l'espectre desitjat s'hauria de considerar l'addició de més LED de diferents colors o temperatures de color. El sistema actual, format per 6 LED, realitza un ajust de gran qualitat per temperatures de color baixes. Per temperatures de color mitjanes i altes, però, no és tant bo tal i com es pot observar a les figures 10 i 11. Com ja s'ha explicat en anterioritat, això és degut a que els LED blancs tenen més d'un pic, introduint complexitat a l'ajust.

- Una altra implementació a tenir en compte seria incorporar una variació automàtica de la temperatura de color de manera que la lluminària vagi canviant el color de forma autònoma. Això faria possible una simulació de la llum exterior a una habitació tancada. Aquesta aplicació pot ser interessant per la regulació de la melatonina i altres processos circadians.
- Es pot pensar, a més a més, en la possibilitat d'introduir nous espectres per tal que el sistema els reproduïxi. Una opció interessant per l'usuari seria incorporar un mecanisme que permetés reproduir més colors o fins i tot permetre l'enregistrament de diferents espectres a través d'un dispositiu mòbil. Aquesta millora necessitaria una reprogramació de part del sistema, ja que les interfícies gràfiques que permeten la comunicació amb la lluminària haurien d'incorporar el programa d'ajust dels LED als diferents espectres.
- Per acabar, es pot realitzar un anàlisi de cicle de vida. Aquesta classe d'estudis, claus per determinar la sostenibilitat d'un producte, ofereixen valors quantitius de diferents indicadors de sostenibilitat com podria ser l'empremta de CO<sub>2</sub> o l'energia involucrada durant la vida útil. En aquest anàlisi es té en compte el procés d'extracció, manufactura, transport i consum del producte, pel que fa necessària la interacció i facilitació d'informació per part de les diferents empreses fabricants. D'aquesta manera es podria saber amb força exactitud l'estalvi real d'energia i emissions d'una lluminària com la presentada respecte una bombeta tradicional incandescent o halògena.

## 6. Bibliografia

- [1] Franz Halberg. *Introduction to Chronobiology*. University of Minnesota, 1994. [En línia]. [Consulta: 07/14]. <http://www.msi.umn.edu/~halberg/introd/index.html>
- [2] Altun A, Ugur-Altun B. Melatonin: therapeutic and clinical utilization. *The International Journal of Clinical Practice*. Maig 2007, volum 61, número 5, pàgines 835-845.
- [3] Malacara, D. *Color vision and colorimetry*. SPIE Press, 2011. ISBN-13: 978-0819483973
- [4] Wikimedia Commons. *Espectre en el rang del visible* [En línia]. [Consulta: 07/14]. [http://upload.wikimedia.org/wikipedia/commons/d/d9/Linear\\_visible\\_spectrum.svg](http://upload.wikimedia.org/wikipedia/commons/d/d9/Linear_visible_spectrum.svg)
- [5] Wikimedia Commons. *Diagrama cromàtic CIE 1931* [En línia]. [Consulta: 04/14]. <http://upload.wikimedia.org/wikipedia/commons/b/b0/CIExy1931.png>
- [6] Wikimedia Commons. *Diagrama cromàtic i planckian locus* [En línia]. [Consulta: 07/14]. <http://upload.wikimedia.org/wikipedia/commons/b/ba/PlanckianLocus.png>
- [7] Schubert, E.F. *Light Emitting Diodes*. Cambridge University Press, 2006. ISBN-13: 978-0521865388
- [8] Judd, Vos. *Modified CIE 2-deg photopic luminosity curve (1978)* [En línia]. [Consulta 07/14]. <http://www.cvrl.org/database/text/lum/vljv.htm>
- [9] National Museum of American History. *Energy Efficiency: Light Sources in the 20th Century* [En línia]. [Consulta: 06/14]. <http://americanhistory.si.edu/lighting/tech/chart.htm>
- [10] Lighting Energy Oportunity. *General Light Source Comparison* [En línia]. [Consulta: 06/14]. [http://www.leoindustries.com/why\\_led\\_lighting.php](http://www.leoindustries.com/why_led_lighting.php)
- [11] LEDs Magazine and Illumination in Focus. *Cree launches 200-lm/W, 1226-lm discrete LED for directional SSL* [En línia] 22 de maig del 2014. [Consulta 08/14]. <http://www.ledsmagazine.com/articles/2014/05/cree-launches-200-lm-w-1226-lm-discrete-led-for-directional-ssl.html>
- [12] Philips LumiLED Lighting Company. *LUXEON Rebel Color Porfolio Datasheet* [En línia]. [Consulta 07/14]. <http://www.farnell.com/datasheets/1719136.pdf>
- [13] Philips LumiLED Lighting Company. *LUXEON A Datasheet* [En línia]. [Consulta 07/14]. <http://www.farnell.com/datasheets/1641570.pdf>
- [14] Arduino. *Arduino Nano* [En línia]. [Consulta: 05/14]. <http://arduino.cc/en/Main/arduinoBoardNano>
- [15] EXP Tech. *Arduino Nano* [En línia]. [Consulta 08/14]. <http://www.exp-tech.de/Mainboards/Arduino-Nano.html>
- [16] Core Electronics. *JY-MCU Bluetooth to Uart Wireless Serial Port* [En línia]. [Consulta 05/14]. <https://core-electronics.com.au/attachments/guides/Product-User-Guide-JY-MCU-Bluetooth-UART-R1-0.pdf>

- [17] Wikimedia Commons. *Circuit divisor de tensió* [En línia]. [Consulta: 06/14].  
[http://upload.wikimedia.org/wikipedia/commons/d/db/Resistive\\_divider.png](http://upload.wikimedia.org/wikipedia/commons/d/db/Resistive_divider.png)
- [18] Friends-of-Fritzing Foundation, IXDS. *Fritzing App* [En línia]. [Consulta: 07/14].  
<http://fritzing.org/home/>
- [19] The Gimp Team. *GNU Image Manipulation Program (GIMP)*. [En línia]. [Consulta: 06/14]  
<http://docs.gimp.org/2.8/en/>
- [20] The MathWorks, Inc. *MATLAB* [En línia]. [Consulta: 04/14].  
<http://www.mathworks.es/products/matlab/>
- [21] MathWorks Classroom Resources Team. *MATLAB Support Package for Arduino (ArduinoIO Package)* [En línia] [Consulta 07/14].  
<http://www.mathworks.com/matlabcentral/fileexchange/32374-matlab-support-package-for-arduino--aka-arduinoio-package->
- [22] Kim et al (2002). Design of Advanced Color – Temperature Control System for HDTV Applications. *Journal of the Korean Physical Society*, 41 (6), 865 – 871. [En línia].  
[Consulta: 04/14].  
<http://icpr.snu.ac.kr/resource/wop.pdf/J01/2002/041/R06/J012002041R060865.pdf>
- [23] Mitchell Charity. *Blackbody color datafile* [En línia]. [Consulta: 04/14].  
[http://www.vendian.org/mncharity/dir3/blackbody/UnstableURLs/bbr\\_color.html](http://www.vendian.org/mncharity/dir3/blackbody/UnstableURLs/bbr_color.html)
- [24] Python Software Foundation. Python 2.7. [En línia]. [Consulta: 04/14]  
<https://docs.python.org/2/>
- [25] Massachusetts Institute of Technology. *MIT App Inventor* [En línia]. [Consulta: 05/14].  
<http://appinventor.mit.edu/explore/>
- [26] Massachusetts Institute of Technology. *Scratch* [En línia]. [Consulta: 05/14].  
[http://info.scratch.mit.edu/support/reference\\_guide\\_1.4](http://info.scratch.mit.edu/support/reference_guide_1.4)
- [27] Premier Farnell plc. *Farnell* [En línia]. [Consulta: 07/14] <http://www.farnell.com/>
- [28] Farnell. *Atmega 328* [En línia]. [Consulta: 07/14]. <http://es.farnell.com/atmel/atmega328p-au/mcu-8bit-avr-32-k-flash-32tqfp/dp/1715486>
- [29] Edison Tech Center. *LEDs and OLEDs* [En línia]. [Consulta: 07/14].  
<http://www.edisontechcenter.org/LED.html>
- [30] Edison Tech Center. *The Halogen Lamp* [En línia]. [Consulta: 07/14].  
<http://www.edisontechcenter.org/halogen.html>

## Annex

### Annex A. Material i especificacions

#### A1. Material utilitzat

- 1 IKEA FADO (element de vidre difusor)
- 1 Arduino Nano ATmega328
- 6 LED de diferents colors / temperatures de color
- 1 dispositiu Bluetooth JY-MCU
- 2 plaques de connexions
- Cablejat, connexions i resistències
- Altres (MATLAB, dispositiu Android, transformador)

#### A2. Especificacions dels LED

Les especificacions dels LED utilitzats es mostren a continuació:

Taula A1. Especificacions dels LED utilitzats [12] [13].

Marca	Sèrie	Color	Temperatura de color (K)	Longitud d'ona (nm)	Flux lluminós (lm)	Corrent nominal (mA)	Voltatge directe (V)
Philips LumiLED	Luxeon Rebel Color	Blau	-	470	70	700	3.3
Philips LumiLED	Luxeon Rebel Color	Cian	-	505	122	700	3.25
Philips LumiLED	Luxeon Rebel Color	Verd	-	530	150	700	3.25
Philips LumiLED	Luxeon Rebel Color	Vermell	-	627	106	700	2.9
Philips LumiLED	Luxeon A	Blanc càlid	2700	-	160	700	2.81
Philips LumiLED	Luxeon A	Blanc neutre	4000	-	183	700	3

### A3. Especificacions de l'Arduino Nano ATmega 328

Les principals característiques i les fotografies il·lustratives de l'Arduino Nano ATmega 328 es detallen a continuació [14]:

- Nombre de pins d'entrada i sortida: 14 (6 dels quals poden proporcionar una sortida PWM)
- Voltatge de sortida màxim dels pins: 5V
- Intensitat de sortida màxima dels pins: 40mA
- Nombre d'entrades analògiques: 8
- Velocitat del processador: 16MHz
- Connexió a través de l'ordinador via USB
- Programació a través d'un software propi disponible per descarregar a la web del projecte Arduino.

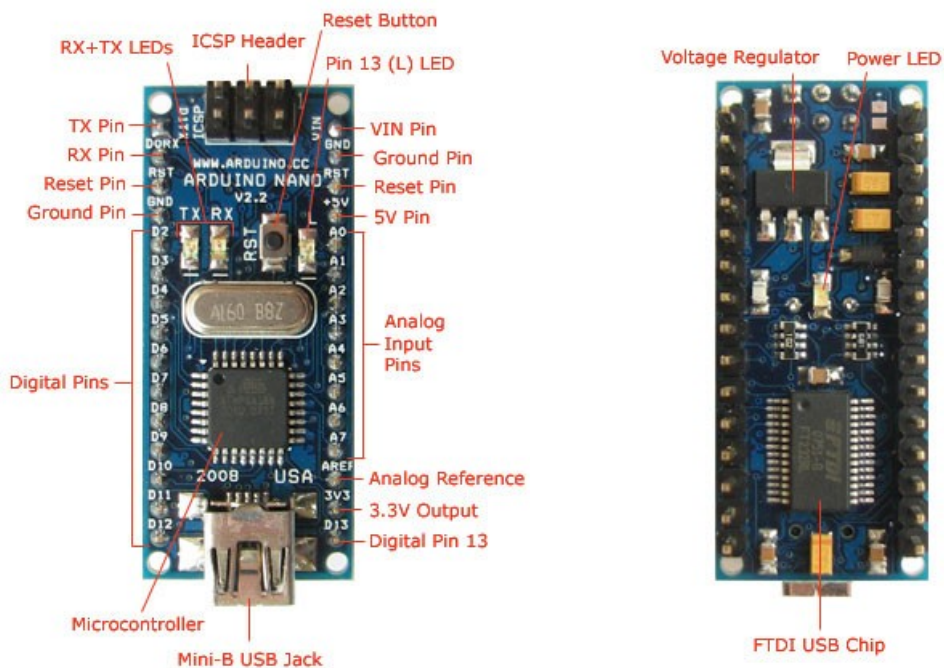


Figura A1. Vista de l'Arduino Nano Atmega328 (a) frontal i (b) posterior [15].

## Annex B. Relacions matemàtiques

### B1. Generació del planckian locus al GUI del MATLAB

El *planckian locus* es pot representar com una corba en el diagrama cromàtic CIE 1931. Per generar els diferents punts que formen el *planckian locus* s'utilitzen dues funcions combinades:

- Una funció polinòmica definida a intervals que relaciona les diferents temperatures de color amb les coordenades  $x$  i  $y$  utilitzada per Kim *et al* [22].
- Una taula de dades utilitzada per Mitchell Charity [23].

La funció `imread(imatge)` del MATLAB llegeix una imatge en color o en escala de grisos retornant una matriu amb les tres coordenades RGB de cada píxel. D'aquesta manera, sabent la localització exacta del *planckian locus* sobre el diagrama cromàtic del model de color CIE 1931, es poden obtenir les tres components RGB per cada punt del *planckian locus*.

Cal notar, però, que s'està treballant amb dos sistemes de coordenades diferents: per una banda es tenen les coordenades  $(x, y)$  que formen el *planckian locus* i per altra una imatge (el diagrama cromàtic) que utilitza un altre sistema de coordenades amb el píxel com a unitat mètrica. Per tant, si es vol representar el *planckian locus* sobre el diagrama cromàtic per saber les components RGB, s'ha de realitzar un canvi de coordenades.

Per trobar la relació entre les coordenades  $(x, y)$  del *planckian locus* i les coordenades de la imatge s'ha d'aplicar una transformació lineal de la forma:

$$x_p = a \cdot x_g + b$$

$$y_p = c \cdot y_g + d$$

on  $(x_p, y_p)$  són les coordenades referents a la imatge (en píxels) i  $(x_g, y_g)$  són les coordenades referents al *planckian locus*. Es necessiten, doncs, dos punts per tal de poder trobar les constants  $a$ ,  $b$ ,  $c$  i  $d$  que definiran la transformació. Amb l'ajuda del software d'edició d'imatges GIMP es pot trobar fàcilment aquesta relació. Situant el cursor sobre punts coneguts dels eixos de coordenades de la figura 3, les relacions trobades són les següents:

**Taula A2.** Relacions entre els píxels de la figura 3 i les coordenades del *planckian locus*.

Gràfic		Píxels	
$X_g$	$Y_g$	$X_p$	$Y_p$
0.8	0	1098	1155
0	0.9	116	50

Així doncs, s'han de resoldre els següents sistemes d'equacions:

$$\begin{cases} 1098 = a \cdot 0.8 + b \\ 116 = a \cdot 0 + b \end{cases} \quad \begin{cases} 1155 = c \cdot 0 + d \\ 50 = c \cdot 0.9 + d \end{cases}$$

Procedint, arribem als següents resultats:

$$a = 1228 \text{ px}$$

$$b = 116 \text{ px}$$

$$c = -1228 \text{ px}$$

$$d = 1155 \text{ px}$$

El fet que el valor de  $c$  sigui negatiu respon al diferent origen de coordenades i al diferent sentit de l'eix de les ordenades als dos sistemes.



## Annex C. Programació

### C1. Càlcul de la intensitat dels diferents LED en funció de la temperatura de color d'un cos negre (MATLAB)

```
Pi = pi; %PI number
hPLANCK = 4.13566751691e-15; % Planck constant in eV·s
cLIGHT = 2.99792458e17; % speed of light in vacuum in nm/s
KBoltzmann = 8.617332478e-5; %Boltzmann constant in eV/K
Temperature = 25; % temperature in Celcius
ExponentialConstant = hPLANCK * cLIGHT /
(KBoltzmann*(273.15+Temperature)); % h*c/(KT) in units of cm

%%%%%% Open paralalization %%%%%%

% We need the eye response in order to normalize the LED intensity
% (and evaluate it in lux)
eye_response = load('eye_response.dat');
integral_eye_response = trapz (eye_response(:,1),eye_response(:,2));
eye_response(:,2) = eye_response(:,2)/integral_eye_response;

% Loading the spectra from commercial LEDs, from Lumileds
led450 = load('led450.dat');
%integral_led450 = trapz (led450(:,1), led450(:,2).*eye_response(:,2))
* 683.002*4*Pi;
led460 = load('led460.dat');
%integral_led460 = trapz (led460(:,1), led460(:,2).*eye_response(:,2))
* 683.002*4*Pi;
led470 = load('led470.dat');
integral_led470 = trapz (led470(:,1), led470(:,2).*eye_response(:,2)) *
683.002*4*Pi;
led470(:,2) = led470(:,2) /integral_led470 * 41 * 10^2 * 2; % 41 lm @
350 mA --- 70 /2; % 70 lm @ 700 mA, we will consider a current of only
350 mA
led505 = load('led505.dat');
integral_led505 = trapz (led505(:,1), led505(:,2).*eye_response(:,2)) *
683.002*4*Pi;
led505(:,2) = led505(:,2) /integral_led505 * 76 * 10^2 * 2; % 76 lm @
350 mA --- 122 /2; % 122 lm @ 700 mA, we will consider a current of
only 350 mA
led525 = load('led525.dat');
integral_led525 = trapz (led525(:,1), led525(:,2).*eye_response(:,2)) *
683.002*4*Pi;
led525(:,2) = led525(:,2) /integral_led525 * 95 * 10^2 * 2; %95 lm @
350 mA --- 125 /2; % 125 lm @ 700 mA, we will consider a current of
only 350 mA
led590 = load('led590.dat');
% integral_led590 = trapz (led590(:,1), led590(:,2).*eye_response(:,2))
* 683.002*4*Pi;
% led590(:,2) = led590(:,2) /integral_led590 * 77 /2; % 77 lm @ 700 mA,
we will consider a current of only 350 mA
led630 = load('led630.dat');
% integral_led630 = trapz (led630(:,1), led630(:,2).*eye_response(:,2))
* 683.002*4*Pi;
% led630(:,2) = led630(:,2) /integral_led630 * 134 /2; % 134 lm @ 700
mA, we will consider a current of only 350 mA
led640 = load('led640.dat');
integral_led640 = trapz (led640(:,1), led640(:,2).*eye_response(:,2)) *
683.002*4*Pi;
```

```

led640(:,2) = led640(:,2) /integral_led640 * 53 * 10^2 * 2; %53 lm @
350 mA --- 102 /2; % 102 lm @ 700 mA, we will consider a current of
only 350 mA
led660 = load('led660.dat');
% integral_led660 = trapz (led660(:,1), led660(:,2).*eye_response(:,2))
* 683.002*4*Pi;
% led660(:,2) = led660(:,2) /integral_led660 * 1 /2; % 122 lm @ 700 mA,
we will consider a current of only 350 mA
ledwht = load('ledwht.dat');
integral_ledwht = trapz (ledwht(:,1), ledwht(:,2).*eye_response(:,2)) *
683.002*4*Pi;
ledwht(:,2) = ledwht(:,2) /integral_ledwht * 89 * 10^1.3 * 2; %
considering 3 LEDs with 89 lm @ 350 mA --- 160 /2; % 160 lm @ 700 mA,
we will consider a current of only 350 mA
ledbwh = load('ledbwh.dat');
integral_ledbwh = trapz (ledbwh(:,1), ledbwh(:,2).*eye_response(:,2)) *
683.002*4*Pi;
ledbwh(:,2) = ledbwh(:,2) /integral_ledbwh * 106 * 10^1 * 2 ; %
considering 3 LEDs with 106 lm @ 350mA --- 190 /2; % 190 lm @ 700 mA,
we will consider a current of only 350 mA

% making available the spectra in different scripts
setappdata(0, 'led450', led450);
setappdata(0, 'led460', led460);
setappdata(0, 'led470', led470);
setappdata(0, 'led505', led505);
setappdata(0, 'led525', led525);
setappdata(0, 'led590', led590);
setappdata(0, 'led630', led630);
setappdata(0, 'led640', led640);
setappdata(0, 'led660', led660);
setappdata(0, 'ledwht', ledwht);
setappdata(0, 'ledbwh', ledbwh);

% Blackbody temperature is changed from 1000 to 8000 K
%for Temp=1500:500:7500;
for ooo=1:71;
    Temp = 900+100*ooo;
    arrayTemp(ooo) = Temp;
% Temp = 3500;
%Temp = input('Enter the temperature for calculating the blackbody
spectrum (in Kelvin): ')
%Temp = str2num(Temp);
fprintf('\n %%%%%%%%%%%\n');

fprintf('\n Temperature = %g K \n', Temp);
jj=0;
setappdata(0, 'n_iterations', jj);

% Spectrum calculation from the blackbody
for i=1:900;
lambda = 350.0 + i*0.5;

Ibb(i,1) = lambda;
Ibb(i,2) = (2*hPLANCK*cLIGHT/lambda^3) / (exp( (hPLANCK*cLIGHT)/
(KBoltzmann*Temp*lambda)) -1) * (hPLANCK*cLIGHT / lambda^2);

end;

```

```

%Ibb(:,2) = Ibb(:,2)/max(Ibb(:,2))/10;
integral_Ibb = trapz (Ibb(:,1), Ibb(:,2));
Ibb(:,2) = Ibb(:,2) /integral_Ibb *20 / 3.35; %This is just a
normalization factor, only for our convinience...

% making available the spectrum in different scripts
setappdata(0, 'lambda_x', Ibb(:,1));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Lower Fitting Limits %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
lower_interval (1) = 0; %I450, Intensity factor for the led emitting at
a central wavelength of 450 nm
lower_interval (2) = 0; %I460, Intensity factor for the led emitting at
a central wavelength of 460 nm
lower_interval (3) = 0; %I470, Intensity factor for the led emitting at
a central wavelength of 470 nm
lower_interval (4) = 0; %I505, Intensity factor for the led emitting at
a central wavelength of 505 nm
lower_interval (5) = 0; %I525, Intensity factor for the led emitting at
a central wavelength of 525 nm
lower_interval (6) = 0; %I590, Intensity factor for the led emitting at
a central wavelength of 590 nm
lower_interval (7) = 0; %I630, Intensity factor for the led emitting at
a central wavelength of 630 nm
lower_interval (8) = 0; %I640, Intensity factor for the led emitting at
a central wavelength of 640 nm
lower_interval (9) = 0; %I660, Intensity factor for the led emitting at
a central wavelength of 660 nm
lower_interval (10) = 0; %Iwhty, Intensity factor for the white led
(yellow)
lower_interval (11) = 0; %Iwhtb, Intensity factor for the white led
(blue)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Upper Fitting Limits %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
upper_interval (1) = 2e-08; %I450, Intensity factor for the led
emitting at a central wavelength of 450 nm
upper_interval (2) = 2e-08; %I460, Intensity factor for the led
emitting at a central wavelength of 460 nm
upper_interval (3) = 1.5; %I470, Intensity factor for the led emitting
at a central wavelength of 470 nm
upper_interval (4) = 1.5; %I505, Intensity factor for the led emitting
at a central wavelength of 505 nm
upper_interval (5) = 1.5; %I525, Intensity factor for the led emitting
at a central wavelength of 525 nm
upper_interval (6) = 2e-08; %I590, Intensity factor for the led
emitting at a central wavelength of 590 nm
upper_interval (7) = 2e-08; %I630, Intensity factor for the led
emitting at a central wavelength of 630 nm
upper_interval (8) = 1.5; %I640, Intensity factor for the led emitting
at a central wavelength of 640 nm
upper_interval (9) = 2e-08; %I660, Intensity factor for the led
emitting at a central wavelength of 660 nm
upper_interval (10) = 1; %Iwhty, Intensity factor for the white led
(yellow)

```

```

upper_interval (11) = 1; %Iwhbtb, Intensity factor for the white led
(blue)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Starting Fitting Points %%%%%%%%%%%%%%%
starting_point (1) = 0; %I450, Intensity factor for the led emitting at
a central wavelength of 450 nm
starting_point (2) = 0.5; %I460, Intensity factor for the led emitting
at a central wavelength of 460 nm
starting_point (3) = 0.5; %I470, Intensity factor for the led emitting
at a central wavelength of 470 nm
starting_point (4) = 0.5; %I505, Intensity factor for the led emitting
at a central wavelength of 505 nm
starting_point (5) = 0.5; %I525, Intensity factor for the led emitting
at a central wavelength of 525 nm
starting_point (6) = 0; %I590, Intensity factor for the led emitting at
a central wavelength of 590 nm
starting_point (7) = 0; %I630, Intensity factor for the led emitting at
a central wavelength of 630 nm
starting_point (8) = 0; %I640, Intensity factor for the led emitting at
a central wavelength of 640 nm
starting_point (9) = 0; %I660, Intensity factor for the led emitting at
a central wavelength of 660 nm
starting_point (10) = 0.5; %Iwhty, Intensity factor for the white led
(yellow)
starting_point (11) = 0.5; %Iwhbtb, Intensity factor for the white led
(blue)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Initial guest is calculated to show it together with the blackbody
% spectrum

I450 = starting_point (1); %I450, Intensity factor for the led emitting
at a central wavelength of 450 nm
I460 = starting_point (2); %I460, Intensity factor for the led emitting
at a central wavelength of 460 nm
I470 = starting_point (3); %I470, Intensity factor for the led emitting
at a central wavelength of 470 nm
I505 = starting_point (4); %I505, Intensity factor for the led emitting
at a central wavelength of 505 nm
I525 = starting_point (5); %I525, Intensity factor for the led emitting
at a central wavelength of 525 nm
I590 = starting_point (6); %I590, Intensity factor for the led emitting
at a central wavelength of 590 nm
I630 = starting_point (7); %I630, Intensity factor for the led emitting
at a central wavelength of 630 nm
I640 = starting_point (8); %I640, Intensity factor for the led emitting
at a central wavelength of 640 nm
I660 = starting_point (9); %I660, Intensity factor for the led emitting
at a central wavelength of 660 nm
Iwhty = starting_point (10); %Iwhty, Intensity factor for the white led
(yellow)
Iwhbtb = starting_point (11); %Iwhbtb, Intensity factor for the white led
(blue)

for i=1:900,
Isimulated(i) = I450*led450(i,2) + I460*led460(i,2) + I470*led470(i,2)
+ I505*led505(i,2) + I525*led525(i,2) + I590*led590(i,2) +

```

```

I630*led630(i,2) + I640*led640(i,2) + I660*led660(i,2) +
Iwhty*ledwht(i,2)+ Iwhtb*ledbwh(i,2);

end;

%%% Repeating the fitting procedure 5 times!!
for kk=1:5;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Starting Fitting Points %%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
starting_point (1) = I450; %I450, Intensity factor for the led emitting
at a central wavelength of 450 nm
starting_point (2) = I460; %I460, Intensity factor for the led emitting
at a central wavelength of 460 nm
starting_point (3) = I470; %I470, Intensity factor for the led emitting
at a central wavelength of 470 nm
starting_point (4) = I505; %I505, Intensity factor for the led emitting
at a central wavelength of 505 nm
starting_point (5) = I525; %I525, Intensity factor for the led emitting
at a central wavelength of 525 nm
starting_point (6) = I590; %I590, Intensity factor for the led emitting
at a central wavelength of 590 nm
starting_point (7) = I630; %I630, Intensity factor for the led emitting
at a central wavelength of 630 nm
starting_point (8) = I640; %I640, Intensity factor for the led emitting
at a central wavelength of 640 nm
starting_point (9) = I660; %I660, Intensity factor for the led emitting
at a central wavelength of 660 nm
starting_point (10) = Iwhty; %Iwhty, Intensity factor for the white
led (yellow)
starting_point (11) = Iwhtb; %Iwhtb, Intensity factor for the white led
(blue)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Start counting time %%%%%%%%%%%
initial_time = clock;

s =
fitoptions('Method','NonlinearLeastSquares','Lower',lower_interval,'Upp
er',upper_interval,'Startpoint',starting_point);
[SLbbdata, SLbbstatistics] =
fit(Ibb(:,1),Ibb(:,2),fittype('BlackBody(I450, I460, I470, I505, I525,
I590, I630, I640, I660, Iwhty, Iwhtb, x)', 'options', s));

final_time = clock;
total_elapsed_time = etime (final_time, initial_time)/60;

I450 = SLbbdata.I450;
I460 = SLbbdata.I460;
I470 = SLbbdata.I470;
I505 = SLbbdata.I505;
I525 = SLbbdata.I525;
I590 = SLbbdata.I590;
I630 = SLbbdata.I630;
I640 = SLbbdata.I640;
I660 = SLbbdata.I660;
Iwhty = SLbbdata.Iwhty;
Iwhtb = SLbbdata.Iwhtb;

```

```

Ileds (1) = I450;
Ileds (2) = I460;
Ileds (3) = I470;
Ileds (4) = I505;
Ileds (5) = I525;
Ileds (6) = I590;
Ileds (7) = I630;
Ileds (8) = I640;
Ileds (9) = I450;
Ileds (10) = Iwhty;
Ileds (11) = Iwhtb;

end;

for i=1:900,
Isimulated(i) = (I450*led450(i,2) + I460*led460(i,2) +
I470*led470(i,2) + I505*led505(i,2) + I525*led525(i,2) +
I590*led590(i,2) + I630*led630(i,2) + I640*led640(i,2) +
I660*led660(i,2) + Iwhty*ledwht(i,2)+ Iwhtb*ledbwh(i,2)) / max(Ileds);%
* eye_response(i,2);
%Ibb(i,2) = Ibb(i,2) * eye_response(i,2);
Ibb(i,2) = Ibb(i,2) / max(Ileds);
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Plotting Initial Guest + Experimental spectra
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hold off;
plot(Ibb(:,1),times(Ibb(:,2),eye_response(:,2)), 'r');
hold on
plot(Ibb(:,1),times(Isimulated,eye_response(:,2)),'k');
titol = strcat('Temperatura ', num2str(Temp), 'K');
title(titol);
ylabel('radiació emesa');
xlabel('longitud ona (nm)');
hold off;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pause(2.5);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Displaying out Parameters %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

final_time = clock;
total_elapse_time = etime (final_time, initial_time)/60;

n_iterations = getappdata(0,'n_iterations');
fprintf('\n Total number of iterations: \n %g \n', n_iterations);
%fprintf('\n \n \n ');
fprintf('\n Total time elapse for the calculations: \n %g min \n',
total_elapse_time);
%fprintf('\n \n \n ');
%fprintf('\n Obtained best fitting paremeters: \n \n');
SLbbdata.I450 = SLbbdata.I450;
SLbbdata.I460 = SLbbdata.I460;
SLbbdata.I470 = SLbbdata.I470;
SLbbdata.I505 = SLbbdata.I505;
SLbbdata.I525 = SLbbdata.I525;
SLbbdata.I590 = SLbbdata.I590;
SLbbdata.I630 = SLbbdata.I630;

```

```

SLbbdata.I640 = SLbbdata.I640;
SLbbdata.I660 = SLbbdata.I660;
SLbbdata.Iwhty = SLbbdata.Iwhty;
SLbbdata.Iwhtb = SLbbdata.Iwhtb;

Int_Isimulated = trapz (Ibb(:,1), Isimulated');
wavelength_spd = Ibb(:,1);
spd = Isimulated'/Int_Isimulated;

% First, obtain reference data values
% The CIE 1931 2 degree Standard Colorimetric Observer
% [wavelength(nm)  xbar  ybar  zbar]
CIE31Table = [360 0.000130 0.000004 0.000606;...
361 0.000146 0.000004 0.000681;...
362 0.000164 0.000005 0.000765;...
363 0.000184 0.000006 0.000860;...
364 0.000207 0.000006 0.000967;...
365 0.000232 0.000007 0.001086;...
366 0.000261 0.000008 0.001221;...
367 0.000293 0.000009 0.001373;...
368 0.000329 0.000010 0.001544;...
369 0.000370 0.000011 0.001734;...
370 0.000415 0.000012 0.001946;...
371 0.000464 0.000014 0.002178;...
372 0.000519 0.000016 0.002436;...
373 0.000582 0.000017 0.002732;...
374 0.000655 0.000020 0.003078;...
375 0.000742 0.000022 0.003486;...
376 0.000845 0.000025 0.003975;...
377 0.000965 0.000028 0.004541;...
378 0.001095 0.000032 0.005158;...
379 0.001231 0.000035 0.005803;...
380 0.001368 0.000039 0.006450;...
381 0.001502 0.000043 0.007083;...
382 0.001642 0.000047 0.007745;...
383 0.001802 0.000052 0.008501;...
384 0.001996 0.000057 0.009415;...
385 0.002236 0.000064 0.010550;...
386 0.002535 0.000072 0.011966;...
387 0.002893 0.000082 0.013656;...
388 0.003301 0.000094 0.015588;...
389 0.003753 0.000106 0.017730;...
390 0.004243 0.000120 0.020050;...
391 0.004762 0.000135 0.022511;...
392 0.005330 0.000151 0.025203;...
393 0.005979 0.000170 0.028280;...
394 0.006741 0.000192 0.031897;...
395 0.007650 0.000217 0.036210;...
396 0.008751 0.000247 0.041438;...
397 0.010029 0.000281 0.047504;...
398 0.011422 0.000319 0.054120;...
399 0.012869 0.000357 0.060998;...
400 0.014310 0.000396 0.067850;...
401 0.015704 0.000434 0.074486;...
402 0.017147 0.000473 0.081362;...
403 0.018781 0.000518 0.089154;...
404 0.020748 0.000572 0.098540;...
405 0.023190 0.000640 0.110200;...
406 0.026207 0.000725 0.124613;...

```

407 0.029782 0.000825 0.141702; ...  
408 0.033881 0.000941 0.161303; ...  
409 0.038468 0.001070 0.183257; ...  
410 0.043510 0.001210 0.207400; ...  
411 0.048996 0.001362 0.233692; ...  
412 0.055023 0.001531 0.262611; ...  
413 0.061719 0.001720 0.294775; ...  
414 0.069212 0.001935 0.330799; ...  
415 0.077630 0.002180 0.371300; ...  
416 0.086958 0.002455 0.416209; ...  
417 0.097177 0.002764 0.465464; ...  
418 0.108406 0.003118 0.519695; ...  
419 0.120767 0.003526 0.579530; ...  
420 0.134380 0.004000 0.645600; ...  
421 0.149358 0.004546 0.718484; ...  
422 0.165396 0.005159 0.796713; ...  
423 0.181983 0.005829 0.877846; ...  
424 0.198611 0.006546 0.959439; ...  
425 0.214770 0.007300 1.039050; ...  
426 0.230187 0.008087 1.115367; ...  
427 0.244880 0.008909 1.188497; ...  
428 0.258777 0.009768 1.258123; ...  
429 0.271808 0.010664 1.323930; ...  
430 0.283900 0.011600 1.385600; ...  
431 0.294944 0.012573 1.442635; ...  
432 0.304897 0.013583 1.494804; ...  
433 0.313787 0.014630 1.542190; ...  
434 0.321645 0.015715 1.584881; ...  
435 0.328500 0.016840 1.622960; ...  
436 0.334351 0.018007 1.656405; ...  
437 0.339210 0.019214 1.685296; ...  
438 0.343121 0.020454 1.709875; ...  
439 0.346130 0.021718 1.730382; ...  
440 0.348280 0.023000 1.747060; ...  
441 0.349600 0.024295 1.760045; ...  
442 0.350147 0.025610 1.769623; ...  
443 0.350013 0.026959 1.776264; ...  
444 0.349287 0.028351 1.780433; ...  
445 0.348060 0.029800 1.782600; ...  
446 0.346373 0.031311 1.782968; ...  
447 0.344262 0.032884 1.781700; ...  
448 0.341809 0.034521 1.779198; ...  
449 0.339094 0.036226 1.775867; ...  
450 0.336200 0.038000 1.772110; ...  
451 0.333198 0.039847 1.768259; ...  
452 0.330041 0.041768 1.764039; ...  
453 0.326636 0.043766 1.758944; ...  
454 0.322887 0.045843 1.752466; ...  
455 0.318700 0.048000 1.744100; ...  
456 0.314025 0.050244 1.733559; ...  
457 0.308884 0.052573 1.720858; ...  
458 0.303290 0.054981 1.705937; ...  
459 0.297258 0.057459 1.688737; ...  
460 0.290800 0.060000 1.669200; ...  
461 0.283970 0.062602 1.647529; ...  
462 0.276721 0.065278 1.623413; ...  
463 0.268918 0.068042 1.596022; ...  
464 0.260423 0.070911 1.564528; ...  
465 0.251100 0.073900 1.528100; ...  
466 0.240847 0.077016 1.486111; ...



467 0.229851 0.080266 1.439521; ...  
468 0.218407 0.083667 1.389880; ...  
469 0.206812 0.087233 1.338736; ...  
470 0.195360 0.090980 1.287640; ...  
471 0.184214 0.094918 1.237422; ...  
472 0.173327 0.099046 1.187824; ...  
473 0.162688 0.103367 1.138761; ...  
474 0.152283 0.107885 1.090148; ...  
475 0.142100 0.112600 1.041900; ...  
476 0.132179 0.117532 0.994198; ...  
477 0.122570 0.122674 0.947347; ...  
478 0.113275 0.127993 0.901453; ...  
479 0.104298 0.133453 0.856619; ...  
480 0.095640 0.139020 0.812950; ...  
481 0.087300 0.144676 0.770517; ...  
482 0.079308 0.150469 0.729445; ...  
483 0.071718 0.156462 0.689914; ...  
484 0.064581 0.162718 0.652105; ...  
485 0.057950 0.169300 0.616200; ...  
486 0.051862 0.176243 0.582329; ...  
487 0.046282 0.183558 0.550416; ...  
488 0.041151 0.191274 0.520338; ...  
489 0.036413 0.199418 0.491967; ...  
490 0.032010 0.208020 0.465180; ...  
491 0.027917 0.217120 0.439925; ...  
492 0.024144 0.226735 0.416184; ...  
493 0.020687 0.236857 0.393882; ...  
494 0.017540 0.247481 0.372946; ...  
495 0.014700 0.258600 0.353300; ...  
496 0.012162 0.270185 0.334858; ...  
497 0.009920 0.282294 0.317552; ...  
498 0.007967 0.295050 0.301337; ...  
499 0.006296 0.308578 0.286169; ...  
500 0.004900 0.323000 0.272000; ...  
501 0.003777 0.338402 0.258817; ...  
502 0.002945 0.354686 0.246484; ...  
503 0.002425 0.371699 0.234772; ...  
504 0.002236 0.389288 0.223453; ...  
505 0.002400 0.407300 0.212300; ...  
506 0.002926 0.425630 0.201169; ...  
507 0.003837 0.444310 0.190120; ...  
508 0.005175 0.463394 0.179225; ...  
509 0.006982 0.482940 0.168561; ...  
510 0.009300 0.503000 0.158200; ...  
511 0.012149 0.523569 0.148138; ...  
512 0.015536 0.544512 0.138376; ...  
513 0.019478 0.565690 0.128994; ...  
514 0.023993 0.586965 0.120075; ...  
515 0.029100 0.608200 0.111700; ...  
516 0.034815 0.629346 0.103905; ...  
517 0.041120 0.650307 0.096667; ...  
518 0.047985 0.670875 0.089983; ...  
519 0.055379 0.690842 0.083845; ...  
520 0.063270 0.710000 0.078250; ...  
521 0.071635 0.728185 0.073209; ...  
522 0.080462 0.745464 0.068678; ...  
523 0.089740 0.761969 0.064568; ...  
524 0.099456 0.777837 0.060788; ...  
525 0.109600 0.793200 0.057250; ...  
526 0.120167 0.808110 0.053904; ...

527 0.131114 0.822496 0.050747; ...  
528 0.142368 0.836307 0.047753; ...  
529 0.153854 0.849492 0.044899; ...  
530 0.165500 0.862000 0.042160; ...  
531 0.177257 0.873811 0.039507; ...  
532 0.189140 0.884962 0.036936; ...  
533 0.201169 0.895494 0.034458; ...  
534 0.213366 0.905443 0.032089; ...  
535 0.225750 0.914850 0.029840; ...  
536 0.238321 0.923735 0.027712; ...  
537 0.251067 0.932092 0.025694; ...  
538 0.263992 0.939923 0.023787; ...  
539 0.277102 0.947225 0.021989; ...  
540 0.290400 0.954000 0.020300; ...  
541 0.303891 0.960256 0.018718; ...  
542 0.317573 0.966007 0.017240; ...  
543 0.331438 0.971261 0.015864; ...  
544 0.345483 0.976023 0.014585; ...  
545 0.359700 0.980300 0.013400; ...  
546 0.374084 0.984092 0.012307; ...  
547 0.388640 0.987418 0.011302; ...  
548 0.403378 0.990313 0.010378; ...  
549 0.418312 0.992812 0.009529; ...  
550 0.433450 0.994950 0.008750; ...  
551 0.448795 0.996711 0.008035; ...  
552 0.464336 0.998098 0.007382; ...  
553 0.480064 0.999112 0.006785; ...  
554 0.495971 0.999748 0.006243; ...  
555 0.512050 1.000000 0.005750; ...  
556 0.528296 0.999857 0.005304; ...  
557 0.544692 0.999305 0.004900; ...  
558 0.561209 0.998325 0.004534; ...  
559 0.577821 0.996899 0.004202; ...  
560 0.594500 0.995000 0.003900; ...  
561 0.611221 0.992601 0.003623; ...  
562 0.627976 0.989743 0.003371; ...  
563 0.644760 0.986444 0.003141; ...  
564 0.661570 0.982724 0.002935; ...  
565 0.678400 0.978600 0.002750; ...  
566 0.695239 0.974084 0.002585; ...  
567 0.712059 0.969171 0.002439; ...  
568 0.728828 0.963857 0.002309; ...  
569 0.745519 0.958135 0.002197; ...  
570 0.762100 0.952000 0.002100; ...  
571 0.778543 0.945450 0.002018; ...  
572 0.794826 0.938499 0.001948; ...  
573 0.810926 0.931163 0.001890; ...  
574 0.826825 0.923458 0.001841; ...  
575 0.842500 0.915400 0.001800; ...  
576 0.857932 0.907006 0.001766; ...  
577 0.873082 0.898277 0.001738; ...  
578 0.887894 0.889205 0.001711; ...  
579 0.902318 0.879782 0.001683; ...  
580 0.916300 0.870000 0.001650; ...  
581 0.929800 0.859861 0.001610; ...  
582 0.942798 0.849392 0.001564; ...  
583 0.955278 0.838622 0.001514; ...  
584 0.967218 0.827581 0.001459; ...  
585 0.978600 0.816300 0.001400; ...  
586 0.989386 0.804795 0.001337; ...

587 0.999549 0.793082 0.001270; ...  
588 1.009089 0.781192 0.001205; ...  
589 1.018006 0.769155 0.001147; ...  
590 1.026300 0.757000 0.001100; ...  
591 1.033983 0.744754 0.001069; ...  
592 1.040986 0.732422 0.001049; ...  
593 1.047188 0.720004 0.001036; ...  
594 1.052467 0.707496 0.001021; ...  
595 1.056700 0.694900 0.001000; ...  
596 1.059794 0.682219 0.000969; ...  
597 1.061799 0.669472 0.000930; ...  
598 1.062807 0.656674 0.000887; ...  
599 1.062910 0.643845 0.000843; ...  
600 1.062200 0.631000 0.000800; ...  
601 1.060735 0.618155 0.000761; ...  
602 1.058444 0.605314 0.000724; ...  
603 1.055224 0.592476 0.000686; ...  
604 1.050977 0.579638 0.000645; ...  
605 1.045600 0.566800 0.000600; ...  
606 1.039037 0.553961 0.000548; ...  
607 1.031361 0.541137 0.000492; ...  
608 1.022666 0.528353 0.000435; ...  
609 1.013048 0.515632 0.000383; ...  
610 1.002600 0.503000 0.000340; ...  
611 0.991367 0.490469 0.000307; ...  
612 0.979331 0.478030 0.000283; ...  
613 0.966492 0.465678 0.000265; ...  
614 0.952848 0.453403 0.000252; ...  
615 0.938400 0.441200 0.000240; ...  
616 0.923194 0.429080 0.000230; ...  
617 0.907244 0.417036 0.000221; ...  
618 0.890502 0.405032 0.000212; ...  
619 0.872920 0.393032 0.000202; ...  
620 0.854450 0.381000 0.000190; ...  
621 0.835084 0.368918 0.000174; ...  
622 0.814946 0.356827 0.000156; ...  
623 0.794186 0.344777 0.000136; ...  
624 0.772954 0.332818 0.000117; ...  
625 0.751400 0.321000 0.000100; ...  
626 0.729584 0.309338 0.000086; ...  
627 0.707589 0.297850 0.000075; ...  
628 0.685602 0.286594 0.000065; ...  
629 0.663810 0.275624 0.000057; ...  
630 0.642400 0.265000 0.000050; ...  
631 0.621515 0.254763 0.000044; ...  
632 0.601114 0.244890 0.000039; ...  
633 0.581105 0.235334 0.000036; ...  
634 0.561398 0.226053 0.000033; ...  
635 0.541900 0.217000 0.000030; ...  
636 0.522599 0.208162 0.000028; ...  
637 0.503546 0.199549 0.000026; ...  
638 0.484744 0.191155 0.000024; ...  
639 0.466194 0.182974 0.000022; ...  
640 0.447900 0.175000 0.000020; ...  
641 0.429861 0.167224 0.000018; ...  
642 0.412098 0.159646 0.000016; ...  
643 0.394644 0.152278 0.000014; ...  
644 0.377533 0.145126 0.000012; ...  
645 0.360800 0.138200 0.000010; ...  
646 0.344456 0.131500 0.000008; ...

647 0.328517 0.125025 0.000005; ...  
648 0.313019 0.118779 0.000003; ...  
649 0.298001 0.112769 0.000001; ...  
650 0.283500 0.107000 0.000000; ...  
651 0.269545 0.101476 0.000000; ...  
652 0.256118 0.096189 0.000000; ...  
653 0.243190 0.091123 0.000000; ...  
654 0.230727 0.086265 0.000000; ...  
655 0.218700 0.081600 0.000000; ...  
656 0.207097 0.077121 0.000000; ...  
657 0.195923 0.072826 0.000000; ...  
658 0.185171 0.068710 0.000000; ...  
659 0.174832 0.064770 0.000000; ...  
660 0.164900 0.061000 0.000000; ...  
661 0.155367 0.057396 0.000000; ...  
662 0.146230 0.053955 0.000000; ...  
663 0.137490 0.050674 0.000000; ...  
664 0.129147 0.047550 0.000000; ...  
665 0.121200 0.044580 0.000000; ...  
666 0.113640 0.041759 0.000000; ...  
667 0.106465 0.039085 0.000000; ...  
668 0.099690 0.036564 0.000000; ...  
669 0.093331 0.034200 0.000000; ...  
670 0.087400 0.032000 0.000000; ...  
671 0.081901 0.029963 0.000000; ...  
672 0.076804 0.028077 0.000000; ...  
673 0.072077 0.026329 0.000000; ...  
674 0.067687 0.024708 0.000000; ...  
675 0.063600 0.023200 0.000000; ...  
676 0.059807 0.021801 0.000000; ...  
677 0.056282 0.020501 0.000000; ...  
678 0.052971 0.019281 0.000000; ...  
679 0.049819 0.018121 0.000000; ...  
680 0.046770 0.017000 0.000000; ...  
681 0.043784 0.015904 0.000000; ...  
682 0.040875 0.014837 0.000000; ...  
683 0.038073 0.013811 0.000000; ...  
684 0.035405 0.012835 0.000000; ...  
685 0.032900 0.011920 0.000000; ...  
686 0.030564 0.011068 0.000000; ...  
687 0.028381 0.010273 0.000000; ...  
688 0.026345 0.009533 0.000000; ...  
689 0.024453 0.008846 0.000000; ...  
690 0.022700 0.008210 0.000000; ...  
691 0.021084 0.007624 0.000000; ...  
692 0.019600 0.007085 0.000000; ...  
693 0.018237 0.006591 0.000000; ...  
694 0.016987 0.006138 0.000000; ...  
695 0.015840 0.005723 0.000000; ...  
696 0.014791 0.005343 0.000000; ...  
697 0.013831 0.004996 0.000000; ...  
698 0.012949 0.004676 0.000000; ...  
699 0.012129 0.004380 0.000000; ...  
700 0.011359 0.004102 0.000000; ...  
701 0.010629 0.003838 0.000000; ...  
702 0.009939 0.003589 0.000000; ...  
703 0.009288 0.003354 0.000000; ...  
704 0.008679 0.003134 0.000000; ...  
705 0.008111 0.002929 0.000000; ...  
706 0.007582 0.002738 0.000000; ...

707 0.007089 0.002560 0.000000; ...  
708 0.006627 0.002393 0.000000; ...  
709 0.006195 0.002237 0.000000; ...  
710 0.005790 0.002091 0.000000; ...  
711 0.005410 0.001954 0.000000; ...  
712 0.005053 0.001825 0.000000; ...  
713 0.004718 0.001704 0.000000; ...  
714 0.004404 0.001590 0.000000; ...  
715 0.004109 0.001484 0.000000; ...  
716 0.003834 0.001384 0.000000; ...  
717 0.003576 0.001291 0.000000; ...  
718 0.003334 0.001204 0.000000; ...  
719 0.003109 0.001123 0.000000; ...  
720 0.002899 0.001047 0.000000; ...  
721 0.002704 0.000977 0.000000; ...  
722 0.002523 0.000911 0.000000; ...  
723 0.002354 0.000850 0.000000; ...  
724 0.002197 0.000793 0.000000; ...  
725 0.002049 0.000740 0.000000; ...  
726 0.001911 0.000690 0.000000; ...  
727 0.001781 0.000643 0.000000; ...  
728 0.001660 0.000599 0.000000; ...  
729 0.001546 0.000558 0.000000; ...  
730 0.001440 0.000520 0.000000; ...  
731 0.001340 0.000484 0.000000; ...  
732 0.001246 0.000450 0.000000; ...  
733 0.001158 0.000418 0.000000; ...  
734 0.001076 0.000389 0.000000; ...  
735 0.001000 0.000361 0.000000; ...  
736 0.000929 0.000335 0.000000; ...  
737 0.000862 0.000311 0.000000; ...  
738 0.000801 0.000289 0.000000; ...  
739 0.000743 0.000268 0.000000; ...  
740 0.000690 0.000249 0.000000; ...  
741 0.000641 0.000231 0.000000; ...  
742 0.000595 0.000215 0.000000; ...  
743 0.000552 0.000199 0.000000; ...  
744 0.000512 0.000185 0.000000; ...  
745 0.000476 0.000172 0.000000; ...  
746 0.000442 0.000160 0.000000; ...  
747 0.000412 0.000149 0.000000; ...  
748 0.000383 0.000138 0.000000; ...  
749 0.000357 0.000129 0.000000; ...  
750 0.000332 0.000120 0.000000; ...  
751 0.000310 0.000112 0.000000; ...  
752 0.000289 0.000104 0.000000; ...  
753 0.000270 0.000097 0.000000; ...  
754 0.000252 0.000091 0.000000; ...  
755 0.000235 0.000085 0.000000; ...  
756 0.000219 0.000079 0.000000; ...  
757 0.000205 0.000074 0.000000; ...  
758 0.000191 0.000069 0.000000; ...  
759 0.000178 0.000064 0.000000; ...  
760 0.000166 0.000060 0.000000; ...  
761 0.000155 0.000056 0.000000; ...  
762 0.000145 0.000052 0.000000; ...  
763 0.000135 0.000049 0.000000; ...  
764 0.000126 0.000045 0.000000; ...  
765 0.000117 0.000042 0.000000; ...  
766 0.000110 0.000040 0.000000; ...

767 0.000102 0.000037 0.000000; ...  
768 0.000095 0.000034 0.000000; ...  
769 0.000089 0.000032 0.000000; ...  
770 0.000083 0.000030 0.000000; ...  
771 0.000078 0.000028 0.000000; ...  
772 0.000072 0.000026 0.000000; ...  
773 0.000067 0.000024 0.000000; ...  
774 0.000063 0.000023 0.000000; ...  
775 0.000059 0.000021 0.000000; ...  
776 0.000055 0.000020 0.000000; ...  
777 0.000051 0.000018 0.000000; ...  
778 0.000048 0.000017 0.000000; ...  
779 0.000044 0.000016 0.000000; ...  
780 0.000042 0.000015 0.000000; ...  
781 0.000039 0.000014 0.000000; ...  
782 0.000036 0.000013 0.000000; ...  
783 0.000034 0.000012 0.000000; ...  
784 0.000031 0.000011 0.000000; ...  
785 0.000029 0.000011 0.000000; ...  
786 0.000027 0.000010 0.000000; ...  
787 0.000026 0.000009 0.000000; ...  
788 0.000024 0.000009 0.000000; ...  
789 0.000022 0.000008 0.000000; ...  
790 0.000021 0.000007 0.000000; ...  
791 0.000019 0.000007 0.000000; ...  
792 0.000018 0.000006 0.000000; ...  
793 0.000017 0.000006 0.000000; ...  
794 0.000016 0.000006 0.000000; ...  
795 0.000015 0.000005 0.000000; ...  
796 0.000014 0.000005 0.000000; ...  
797 0.000013 0.000005 0.000000; ...  
798 0.000012 0.000004 0.000000; ...  
799 0.000011 0.000004 0.000000; ...  
800 0.000010 0.000004 0.000000; ...  
801 0.000010 0.000003 0.000000; ...  
802 0.000009 0.000003 0.000000; ...  
803 0.000008 0.000003 0.000000; ...  
804 0.000008 0.000003 0.000000; ...  
805 0.000007 0.000003 0.000000; ...  
806 0.000007 0.000002 0.000000; ...  
807 0.000006 0.000002 0.000000; ...  
808 0.000006 0.000002 0.000000; ...  
809 0.000005 0.000002 0.000000; ...  
810 0.000005 0.000002 0.000000; ...  
811 0.000005 0.000002 0.000000; ...  
812 0.000004 0.000002 0.000000; ...  
813 0.000004 0.000001 0.000000; ...  
814 0.000004 0.000001 0.000000; ...  
815 0.000004 0.000001 0.000000; ...  
816 0.000003 0.000001 0.000000; ...  
817 0.000003 0.000001 0.000000; ...  
818 0.000003 0.000001 0.000000; ...  
819 0.000003 0.000001 0.000000; ...  
820 0.000003 0.000001 0.000000; ...  
821 0.000002 0.000001 0.000000; ...  
822 0.000002 0.000001 0.000000; ...  
823 0.000002 0.000001 0.000000; ...  
824 0.000002 0.000001 0.000000; ...  
825 0.000002 0.000001 0.000000; ...  
826 0.000002 0.000001 0.000000; ...

```

827 0.000002 0.000001 0.000000;...
828 0.000001 0.000001 0.000000;...
829 0.000001 0.000000 0.000000;...
830 0.000001 0.000000 0.000000];
wavelength = CIE31Table(:,1);
xbar = CIE31Table(:,2);
ybar = CIE31Table(:,3);
zbar = CIE31Table(:,4);

% The spectral reflectance data of 14 color test samples for CRI
% [wavelength (nm) TCS1 TCS2 TCS3 ... TCS14]
TCS = [360 116 53 58 57 143 79 150 75 69 42 74 189 71 36;...
365 136 55 59 59 187 81 177 78 72 43 79 175 76 36;...
370 159 59 61 62 233 89 218 84 73 45 86 158 82 36;...
375 190 64 63 67 269 113 293 90 70 47 98 139 90 36;...
380 219 70 65 74 295 151 378 104 66 50 111 120 104 36;...
385 239 79 68 83 306 203 459 129 62 54 121 103 127 36;...
390 252 89 70 93 310 265 524 170 58 59 127 90 161 37;...
395 256 101 72 105 312 339 546 240 55 63 129 82 211 38;...
400 256 111 73 116 313 410 551 319 52 66 127 76 264 39;...
405 254 116 73 121 315 464 555 416 52 67 121 68 313 39;...
410 252 118 74 124 319 492 559 462 51 68 116 64 341 40;...
415 248 120 74 126 322 508 560 482 50 69 112 65 352 41;...
420 244 121 74 128 326 517 561 490 50 69 108 75 359 42;...
425 240 122 73 131 330 524 558 488 49 70 105 93 361 42;...
430 237 122 73 135 334 531 556 482 48 72 104 123 364 43;...
435 232 122 73 139 339 538 551 473 47 73 104 160 365 44;...
440 230 123 73 144 346 544 544 462 46 76 105 207 367 44;...
445 226 124 73 151 352 551 535 450 44 78 106 256 369 45;...
450 225 127 74 161 360 556 522 439 42 83 110 300 372 45;...
455 222 128 75 172 369 556 506 426 41 88 115 331 374 46;...
460 220 131 77 186 381 554 488 413 38 95 123 346 376 47;...
465 218 134 80 205 394 549 469 397 35 103 134 347 379 48;...
470 216 138 85 229 403 541 448 382 33 113 148 341 384 50;...
475 214 143 94 254 410 531 429 366 31 125 167 328 389 52;...
480 214 150 109 281 415 519 408 352 30 142 192 307 397 55;...
485 214 159 126 308 418 504 385 337 29 162 219 282 405 57;...
490 216 174 148 332 419 488 363 325 28 189 252 257 416 62;...
495 218 190 172 352 417 469 341 310 28 219 291 230 429 67;...
500 223 207 198 370 413 450 324 299 28 262 325 204 443 75;...
505 225 225 221 383 409 431 311 289 29 305 347 178 454 83;...
510 226 242 241 390 403 414 301 283 30 365 356 154 461 92;...
515 226 253 260 394 396 395 291 276 30 416 353 129 466 100;...
520 225 260 278 395 389 377 283 270 31 465 346 109 469 108;...
525 225 264 302 392 381 358 273 262 31 509 333 90 471 121;...
530 227 267 339 385 372 341 265 256 32 546 314 75 474 133;...
535 230 269 370 377 363 325 260 251 32 581 294 62 476 142;...
540 236 272 392 367 353 309 257 250 33 610 271 51 483 150;...
545 245 276 399 354 342 293 257 251 34 634 248 41 490 154;...
550 253 282 400 341 331 279 259 254 35 653 227 35 506 155;...
555 262 289 393 327 320 265 260 258 37 666 206 29 526 152;...
560 272 299 380 312 308 253 260 264 41 678 188 25 553 147;...
565 283 309 365 296 296 241 258 269 44 687 170 22 582 140;...
570 298 322 349 280 284 234 256 272 48 693 153 19 618 133;...
575 318 329 332 263 271 227 254 274 52 698 138 17 651 125;...
580 341 335 315 247 260 225 254 278 60 701 125 17 680 118;...
585 367 339 299 229 247 222 259 284 76 704 114 17 701 112;...
590 390 341 285 214 232 221 270 295 102 705 106 16 717 106;...
595 409 341 272 198 220 220 284 316 136 705 100 16 729 101;...

```

```

600 424 342 264 185 210 220 302 348 190 706 96 16 736 98;...
605 435 342 257 175 200 220 324 384 256 707 92 16 742 95;...
610 442 342 252 169 194 220 344 434 336 707 90 16 745 93;...
615 448 341 247 164 189 220 362 482 418 707 87 16 747 90;...
620 450 341 241 160 185 223 377 528 505 708 85 16 748 89;...
625 451 339 235 156 183 227 389 568 581 708 82 16 748 87;...
630 451 339 229 154 180 233 400 604 641 710 80 18 748 86;...
635 451 338 224 152 177 239 410 629 682 711 79 18 748 85;...
640 451 338 220 151 176 244 420 648 717 712 78 18 748 84;...
645 451 337 217 149 175 251 429 663 740 714 78 18 748 84;...
650 450 336 216 148 175 258 438 676 758 716 78 19 748 84;...
655 450 335 216 148 175 263 445 685 770 718 78 20 748 84;...
660 451 334 219 148 175 268 452 693 781 720 81 23 747 85;...
665 451 332 224 149 177 273 457 700 790 722 83 24 747 87;...
670 453 332 230 151 180 278 462 705 797 725 88 26 747 92;...
675 454 331 238 154 183 281 466 709 803 729 93 30 747 96;...
680 455 331 251 158 186 283 468 712 809 731 102 35 747 102;...
685 457 330 269 162 189 286 470 715 814 735 112 43 747 110;...
690 458 329 288 165 192 291 473 717 819 739 125 56 747 123;...
695 460 328 312 168 195 296 477 719 824 742 141 74 746 137;...
700 462 328 340 170 199 302 483 721 828 746 161 97 746 152;...
705 463 327 366 171 200 313 489 720 830 748 182 128 746 169;...
710 464 326 390 170 199 325 496 719 831 749 203 166 745 188;...
715 465 325 412 168 198 338 503 722 833 751 223 210 744 207;...
720 466 324 431 166 196 351 511 725 835 753 242 257 743 226;...
725 466 324 447 164 195 364 518 727 836 754 257 305 744 243;...
730 466 324 460 164 195 376 525 729 836 755 270 354 745 260;...
735 466 323 472 165 196 389 532 730 837 755 282 401 748 277;...
740 467 322 481 168 197 401 539 730 838 755 292 446 750 294;...
745 467 321 488 172 200 413 546 730 839 755 302 485 750 310;...
750 467 320 493 177 203 425 553 730 839 756 310 520 749 325;...
755 467 318 497 181 205 436 559 730 839 757 314 551 748 339;...
760 467 316 500 185 208 447 565 730 839 758 317 577 748 353;...
765 467 315 502 189 212 458 570 730 839 759 323 599 747 366;...
770 467 315 505 192 215 469 575 730 839 759 330 618 747 379;...
775 467 314 510 194 217 477 578 730 839 759 334 633 747 390;...
780 467 314 516 197 219 485 581 730 839 759 338 645 747 399;...
785 467 313 520 200 222 493 583 730 839 759 343 656 746 408;...
790 467 313 524 204 226 500 585 731 839 759 348 666 746 416;...
795 466 312 527 210 231 506 587 731 839 759 353 674 746 422;...
800 466 312 531 218 237 512 588 731 839 759 359 680 746 428;...
805 466 311 535 225 243 517 589 731 839 759 365 686 745 434;...
810 466 311 539 233 249 521 590 731 838 758 372 691 745 439;...
815 466 311 544 243 257 525 590 731 837 757 380 694 745 444;...
820 465 311 548 254 265 529 590 731 837 757 388 697 745 448;...
825 464 311 552 264 273 532 591 731 836 756 396 700 745 451;...
830 464 310 555 274 280 535 592 731 836 756 403 702 745 454];

```

```
TCS(:,2:end) = TCS(:,2:end)/1000;
```

```
% Data for isothermperature lines needed for calculating correlated color
temperature
```

```
% The following provides a table of isothermperature lines for use with
the Robertson Method
```

```
% (Robertson, 1968) to interpolate isothermperature lines from the CIE
1960 UCS.
```

```
% The spacing of the isothermp lines is very small (1 1/MK) so very
little
```

```
% interpolation is actually needed for determining CCT. The latest
(2002)
```



```

% recommended values for the physical constants determining blackbody
% radiation spectra are used

dwave = wavelength(2)-wavelength(1); % wavelength increment = 1 nm

ubar = (2/3)*xbar;
vbar = ybar;
wbar = -0.5*xbar + (3/2)*ybar + 0.5*zbar;

% 2002 CODATA recommended values
h = 6.6260693e-34;
c = 299792458;
k = 1.3806505e-23;
c1 = 2*pi*h*c^2;
c2 = h*c/k;

MrecpK = [0.01 1:600]; % mega reciprocal Kelvin values of
isotemperature lines
T = 1./(MrecpK*1e-6);
for i = 1:length(T)
    spdref = c1 * (1e-9*wavelength).^5 ./ (exp(c2./(T(i).* 1e-
9*wavelength)) - 1);
    spdref = spdref/max(spdref);
    wave = wavelength*1e-9;

    % Equations from Wyszecki and Stiles, Color Science, 2nd ed. 1982,
page
% 226 and 227
    U = sum(spdref.*ubar);
    V = sum(spdref.*vbar);
    W = sum(spdref.*wbar);
    R = U+V+W;
    u(i) = U/R;
    v(i) = V/R;

    Uprime = c1*c2*(T(i))^2*sum(wave.^6.*ubar.*exp(c2./
(wave.*T(i))).*(exp(c2./(wave.*(T(i))))-1).^2)*dwave;
    Vprime = sum(c1*c2*T(i)^2*wave.^6.*vbar.*exp(c2./
(wave.*T(i))).*(exp(c2./(wave.*(T(i))))-1).^2)*dwave;
    Wprime = sum(c1*c2*T(i)^2*wave.^6.*wbar.*exp(c2./
(wave.*T(i))).*(exp(c2./(wave.*(T(i))))-1).^2)*dwave;
    Rprime = Uprime+Vprime+Wprime;

    sl(i) = (Vprime*R-V*Rprime)/(Uprime*R-U*Rprime);
    m(i) = -1/sl(i);
end
ut = u;
vt = v;
tt = m;
isoTempLinesTable = [T' u' v' m'];
%save isoTempLinesNewestFine.txt isoTempLinesTable -ascii; % Optionally
save file

% Second, calculate Correlated Color Temperature (CCT), Tc

%load ('isoTempLinesNewestFine.mat', 'T', 'ut', 'vt', 'tt'); % If read
from previously saved file

% Interpolate CIE functions to spd increments
xbar = interp1(wavelength,xbar,wavelength_spd);

```

```

xbar(isnan(xbar)) = 0.0;
ybar = interp1(wavelength,ybar,wavelength_spd);
ybar(isnan(ybar)) = 0.0;
zbar = interp1(wavelength,zbar,wavelength_spd);
zbar(isnan(zbar)) = 0.0;
% Calculate Chromaticity Coordinates
X = trapz(wavelength_spd,spd.*xbar);
Y = trapz(wavelength_spd,spd.*ybar);
Z = trapz(wavelength_spd,spd.*zbar);
x = X/(X+Y+Z);
y = Y/(X+Y+Z);
u = 4*x/(-2*x+12*y+3);
v = 6*y/(-2*x+12*y+3);
fprintf(1, 'x = %.4f\ty = %.4f\n',x,y);

% Find adjacent lines to (us, vs)
n = length (T);
index = 0;
d1 = ((v-vt(1)) - tt(1)*(u-ut(1)))/sqrt(1+tt(1)*tt(1));
for i=2:n
    d2 = ((v-vt(i)) - tt(i)*(u-ut(i)))/sqrt(1+tt(i)*tt(i));
    if (d1/d2 < 0)
        index = i;
        break;
    else
        d1 = d2;
    end
end
if index == 0
    Tc = -1; % Not able to calculate CCT, u, v coordinates outside
range.
    fprintf(1, 'Not able to calculate CCT, u, v coordinates outside
range.\n');
    %return
else
    % Calculate CCT by interpolation between isothermperature lines
    Tc = 1/(1/T(index-1)+d1/(d1-d2)*(1/T(index)-1/T(index-1)));
    fprintf(1, 'CCT = %.1f\n',Tc);
end

% Third, calculate the Color Rendering Indices (CRI and its 14 indices)
% Calculate Reference Source Spectrum, spdref.
if (Tc < 5000)
    c1 = 3.7418e-16;
    c2 = 1.4388e-2;

    spdref = c1 * (1e-9*wavelength_spd).^5 ./ (exp(c2./(Tc.* 1e-
9*wavelength_spd)) - 1);
else
    if (Tc <= 25000)
        %load('CIEDaySn.txt', 'wavelength', 'S0', 'S1', 'S2');
        load('CIEDaySn.txt');
        wavelength = CIEDaySn(:,1);
        S0 = CIEDaySn(:,2);
        S1 = CIEDaySn(:,3);
        S2 = CIEDaySn(:,4);

        if (Tc <= 7000)
            xd = -4.6070e9 / Tc.^3 + 2.9678e6 / Tc.^2 + 0.09911e3 / Tc
+ 0.244063;

```

```

else
    xd = -2.0064e9 / Tc.^3 + 1.9018e6 / Tc.^2 + 0.24748e3 / Tc
+ 0.237040;
end
yd = -3.000*xd*xd + 2.870*xd - 0.275;
M1 = (-1.3515 - 1.7703*xd + 5.9114*yd) / (0.0241 + 0.2562*xd -
0.7341*yd);
M2 = (0.0300 - 31.4424*xd + 30.0717*yd) / (0.0241 + 0.2562*xd -
0.7341*yd);
spdref = S0 + M1*S1 + M2*S2;
spdref = interp1(wavelength, spdref, wavelength_spd);
spdref(isnan(spdref)) = 0.0;
else
    R = -1;
    return
end
end
end

% Interpolate TCS values from 5 nm to spd nm increments
TCS_1 = zeros(length(wavelength_spd),14);
for i = 1:14
    TCS_1(:,i) =
interp1(TCS(:,1), TCS(:,i+1), wavelength_spd, 'linear', 0);
end

% Calculate u, v chromaticity coordinates of samples under test
illuminant, uk, vk and
% reference illuminant, ur, vr.
uki = zeros(1,14);
vki = zeros(1,14);
uri = zeros(1,14);
vri = zeros(1,14);
X = trapz(wavelength_spd, spd .* xbar);
Y = trapz(wavelength_spd, spd .* ybar);
Z = trapz(wavelength_spd, spd .* zbar);
Yknormal = 100 / Y;
Yk = Y*Yknormal;
uk = 4*X/(X+15*Y+3*Z);
vk = 6*Y/(X+15*Y+3*Z);
X = trapz(wavelength_spd, spdref .* xbar);
Y = trapz(wavelength_spd, spdref .* ybar);
Z = trapz(wavelength_spd, spdref .* zbar);
Yrnormal = 100 / Y;
Yr = Y*Yrnormal;
ur = 4*X/(X+15*Y+3*Z);
vr = 6*Y/(X+15*Y+3*Z);
for i = 1:14
    X = trapz(wavelength_spd, spd .* TCS_1(:,i) .* xbar);
    Y = trapz(wavelength_spd, spd .* TCS_1(:,i) .* ybar);
    Z = trapz(wavelength_spd, spd .* TCS_1(:,i) .* zbar);
    Yki(i) = Y*Yknormal;
    uki(i) = 4*X/(X+15*Y+3*Z);
    vki(i) = 6*Y/(X+15*Y+3*Z);
    X = trapz(wavelength_spd, spdref .* TCS_1(:,i) .* xbar);
    Y = trapz(wavelength_spd, spdref .* TCS_1(:,i) .* ybar);
    Z = trapz(wavelength_spd, spdref .* TCS_1(:,i) .* zbar);
    Yri(i) = Y*Yrnormal;
    uri(i) = 4*X/(X+15*Y+3*Z);
    vri(i) = 6*Y/(X+15*Y+3*Z);
end
end

```

```

% Check tolerance for reference illuminant
DC = sqrt((uk-ur).^2 + (vk-vr).^2);

% Apply adaptive (perceived) color shift.
ck = (4 - uk - 10*vk) / vk;
dk = (1.708*vk + 0.404 - 1.481*uk) / vk;
cr = (4 - ur - 10*vr) / vr;
dr = (1.708*vr + 0.404 - 1.481*ur) / vr;

for i = 1:14
    cki = (4 - uki(i) - 10*vki(i)) / vki(i);
    dki = (1.708*vki(i) + 0.404 - 1.481*uki(i)) / vki(i);
    ukip(i) = (10.872 + 0.404*cr/ck*cki - 4*dr/dk*dki) / (16.518 +
1.481*cr/ck*cki - dr/dk*dki);
    vkip(i) = 5.520 / (16.518 + 1.481*cr/ck*cki - dr/dk*dki);
end

% Transformation into 1964 Uniform space coordinates.
for i = 1:14
    Wstarr(i) = 25*Yri(i).^0.333333 - 17;
    Ustarr(i) = 13*Wstarr(i)*(uri(i) - ur);
    Vstarr(i) = 13*Wstarr(i)*(vri(i) - vr);

    Wstark(i) = 25*Yki(i).^0.333333 - 17;
    Ustark(i) = 13*Wstark(i)*(ukip(i) - ur); % after applying the
adaptive color shift, u'k = ur
    Vstark(i) = 13*Wstark(i)*(vkip(i) - vr); % after applying the
adaptive color shift, v'k = vr
end

% Determination of resultant color shift, delta E.
deltaE = zeros(1,14);
R = zeros(1,14);
for i = 1:14
    deltaE(i) = sqrt((Ustarr(i) - Ustark(i)).^2 + (Vstarr(i) -
Vstark(i)).^2 + (Wstarr(i) - Wstark(i)).^2);
    R(i) = 100 - 4.6*deltaE(i);
end
Ra = sum(R(1:8))/8;
fprintf(1, 'CRIra = %.1f\n', Ra);
arrayCRI(ooo) = Ra;

% fourth, calculate the gamut area formed by the 8 CIE standard color
samples
ukii=[uki(:,1:8),uki(1)];
vkii=1.5*[vki(:,1:8),vki(1)];
Ga=polyarea(ukii,vkii);
% Normalize gamut area to equal energy source
Ga=Ga/0.00728468*100;
fprintf(1, 'Gamut Area Index = %.1f\n', Ga);

% Fifth, calculate the FSI (full spectrum index)
% Calculates the Full-spectrum Index

% Interpolate to wavelength interval of 1nm from 380nm to 730nm
numWave = 351;
t=(380:1:730)';
spd=interp1(wavelength_spd,spd,t,'spline');
spd(isnan(spd)) = 0.0;

```

```

spd = spd/sum(spd); % Normalize the relative spd so that the total
power equals 1
%Equal energy cumulative spd
EEcum=(1/numWave:1/numWave:1)';
%Calculate FSI

for j=1:numWave
cum = cumsum(spd); % A MatLab function for cumulative sums
sqrDiff = (cum-EEcum).^2;
sumSqrDiff(j)=sum(sqrDiff);
spd=circshift(spd,1);
end
FSI=mean(sumSqrDiff);
FSCI=100-5.1*FSI;
fprintf(1,'FSCI = %.3f\n',FSCI);

integral_Simulation = trapz (Ibb(:,1), Isimulated.*eye_response(:,2)')
* 683.002*4*Pi;

fprintf(1,'Illuminance = %.3f lm\n',integral_Simulation);

Lumenes_per_watt = integral_Simulation/
((I450*2.95*1e2+I460*2.95*1e2+I470*2.95*1e2+I505*2.9*1e2+I525*2.9*1e2+I
590*2.9*1e2+I630*2.9*1e2+I640*2.1*1e2+I660*2.1*1e2+3*Iwhty*2.85*10^(1.3
)+3*Iwhtb*2.85*1e1)/max(Ileds)*0.350);

fprintf(1,'Illuminance/Elec.Power = %.3f lm/W\n',Lumenes_per_watt);
arrayLumW(ooo) = Lumenes_per_watt;

Temperature_x(ooo) = Tc;
% IntLED450(ooo)=I450/max(Ileds);
% IntLED460(ooo)=I460/max(Ileds);
% IntLED470(ooo)=I470/max(Ileds);
% IntLED505(ooo)=I505/max(Ileds);
% IntLED525(ooo)=I525/max(Ileds);
% IntLED590(ooo)=I590/max(Ileds);
% IntLED630(ooo)=I630/max(Ileds);
% IntLED640(ooo)=I640/max(Ileds);
% IntLED660(ooo)=I660/max(Ileds);
% IntLEDwhty(ooo)=Iwhty/max(Ileds);
% IntLEDwhtb(ooo)=Iwhtb/max(Ileds);

IntLED450(ooo)=I450;
IntLED460(ooo)=I460;
IntLED470(ooo)=I470;
IntLED505(ooo)=I505;
IntLED525(ooo)=I525;
IntLED590(ooo)=I590;
IntLED630(ooo)=I630;
IntLED640(ooo)=I640;
IntLED660(ooo)=I660;
IntLEDwhty(ooo)=Iwhty;
IntLEDwhtb(ooo)=Iwhtb;

fprintf('\n %%%%%%%%%%%\n');
fprintf('\n \n');

%saved_matrix(:,1) = Ibb(:,1);
%saved_matrix(:,2) = Isimulated';

```

```

%matlabpool close;
end;

% printing the results in a new file called parametersTemp.txt
fileID = fopen('parametersTemp2.txt', 'w');

for i=1:size(arrayTemp, 2);
    arrayDataRow = [arrayTemp(i) IntLED450(i) IntLED460(i) IntLED470(i)
IntLED505(i) IntLED525(i) IntLED590(i) IntLED630(i) IntLED640(i)
IntLED660(i) IntLEDwhty(i) IntLEDwhtb(i) arrayCRI(i) arrayLumW(i)];
    fprintf(fileID, '%6.1f\t %6.4f\t %6.4f\t %6.4f\t %6.4f\t %6.4f\t
%6.4f\t %6.4f\t %6.4f\t %6.4f\t %6.4f\t %6.4f\t %4.1f\t %6.3f\n',
arrayDataRow);
end
fclose(fileID);

```

## C2. Generació del planckian locus (Python 2.7.)

```
# programa que genera un planckian locus a partir de les equacions de
  Kim et al.
# s'obre el fitxer que sera escrit
fileWrite = open('locus.dat', 'w')

# per temperatures de 1700K a 2200K en intervals de 100K
for T in range(1600, 2200, 100):
    # coordenada X del planckian locus
    x_c = round(-0.2661239E9/T**3 - 0.2343580E6/T**2 + 0.8776956E3/T
                + 0.179910, 4)
    # coordenada Y del planckian locus
    y_c = round(-1.1063814*x_c**3 - 1.34811020*x_c**2 +
                2.18555832*x_c - 0.20219683, 4)
    # es genera una linia on hi consten les dues coordenades
    fileWrite.write(str(x_c) + '\t' + str(y_c) + '\n')

# per temperatures de 2300K a 4000K en intervals de 100K
for T in range(2200, 4000, 100):
    x_c = round(-0.2661239E9/T**3 - 0.2343580E6/T**2 + 0.8776956E3/T
                + 0.179910, 4)
    y_c = round(-0.9549476*x_c**3 - 1.37418593*x_c**2 +
                2.09137015*x_c - 0.16748867, 4)
    fileWrite.write(str(x_c) + '\t' + str(y_c) + '\n')

# per temperatures de 4100K a 25100 en intervals de 100K
for T in range(4000, 25100, 100):
    x_c = round(-3.0258469E9/T**3 + 2.1070379E6/T**2 + 0.2226347E3/T
                + 0.240390, 4)
    y_c = round(3.0817580*x_c**3 - 5.87338670*x_c**2 + 3.75112997*x_c
                - 0.37001483, 4)
    fileWrite.write(str(x_c) + '\t' + str(y_c) + '\n')

# es tanca el fitxer escrit
fileWrite.close()
```

### C3. Interfície gràfica desenvolupada en MATLAB (MATLAB)

```
function varargout = provesSlider(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @provesSlider_OpeningFcn, ...
                  'gui_OutputFcn',  @provesSlider_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before provesSlider is made visible.
function provesSlider_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% initial plot
CIE = 'CIE1931.png';
CIE_img = imread(CIE);
set(handles.planckian_graph, 'Color' , 'None');

% reading the file locus.dat -> real coordinates of the planckian locus
[x, y] = textread('locus.dat');
% coordinate transformation between the real and the image planckian
locus
[x_PL] = round(666.01*[x] + 55.16);
[y_PL] = round(-670.05*[y] + 602.78);

% loading the image, setting the grid and showing the locus
imshow(CIE, 'Parent', handles.axes1)
hold on
grid on
plot(x, y, 'k-');
axis([0 .8 0 0.9]);
hold off

% setting up Arduino
clear a;
global a;
a = arduino('COM7');
pinMode(a, 3, 'output');
pinMode(a, 5, 'output');
pinMode(a, 6, 'output');
pinMode(a, 9, 'output');
pinMode(a, 10, 'output');
pinMode(a, 11, 'output');

% opening text file for creating a table with all the parameters
global fileID;
```



```

fileID = fopen('parametersTempCSV.txt', 'w');

% --- Outputs from this function are returned to the command line.
function varargout = provesSlider_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on intensity movement.
function intensity_Callback(hObject, eventdata, handles)
% hObject     handle to intensity (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'Value') returns position of intensity
%         get(hObject, 'Min') and get(hObject, 'Max') to determine range
of intensity

% getting the intensity from the intensity slider with the tag
"intensity"

% getting the temperature from the temp slider with the tag
"temperature"
temperature_slider = (get(handles.temperature, 'Value') + 1) * 1000;
temperature_text = num2str(round(temperature_slider / 100) * 100);
set(handles.temperature_label, 'String', temperature_text);

% reading the file locus.dat -> real coordinates of the planckian locus
[x, y] = textread('locus.dat');
% coordinate transformation between the real and the image planckian
locus
[x_PL] = round(1228*[x] + 115);
[y_PL] = round(-1228*[y] + 1155);

% changing the color depending on the temperature
% slider_position gives the position of the slider
slider_position = int16(temperature_slider/100 - 9);

% cla clears the current axis
cla;
% refreshing the whole graphic when the slider changes its value
CIE = 'CIE1931.png';
imshow(CIE, 'Parent', handles.axes1);
hold on;
grid on;
plot(x, y, 'k-');
% this line controls the position of a given T on the graph
plot(x(slider_position), y(slider_position), 'k+');
axis([0 .8 0 0.9]);
%hold off;

% loading and reading the image
CIE_img = imread(CIE);
% getting the 3 RGB components depending on the temperature
RGB_locus = (CIE_img(y_PL(slider_position), x_PL(slider_position), :));

```

```

R_locus = double(CIE_img(y_PL(slider_position), x_PL(slider_position),
1));
G_locus = double(CIE_img(y_PL(slider_position), x_PL(slider_position),
2));
B_locus = double(CIE_img(y_PL(slider_position), x_PL(slider_position),
3));

% scaling the RGB components of the selected temperature
R_locus = (R_locus/255.);
G_locus = (G_locus/255.);
B_locus = (B_locus/255.);

% changing the color of the color_locus handle
set(handles.RGB_locus_label, 'String', RGB_locus);
set(handles.color_locus, 'Color', [R_locus G_locus B_locus])

% dataMatrix is a matrix that contains the intensities of the different
LEDs,
% CRI and Lm/W for each temperature
dataMatrix = dlmread('parametersTemp5.txt', '\t');

% reading and setting the variables related with the intensity slider
intensity_light = round(get(handles.intensity, 'Value') * 10) / 10;
intensity_text = num2str(round(intensity_light * 10) / 10);
set(handles.intensity_label, 'String', intensity_text);

% setting the different intensities with the parameters of the
dataMatrix
% the intensity of each LED is multiplied by the value of the slider
intensity
% intensity_total = dataMatrix(slider_position, 4) +
dataMatrix(slider_position, 5) + dataMatrix(slider_position, 6) +
dataMatrix(slider_position, 9) + dataMatrix(slider_position, 11) +
dataMatrix(slider_position, 12);

intensity_LED470 = dataMatrix(slider_position, 4);
intensity_LED505 = dataMatrix(slider_position, 5);
intensity_LED525 = dataMatrix(slider_position, 6);
intensity_LED640 = dataMatrix(slider_position, 9);
intensity_LEDwhty = dataMatrix(slider_position, 11);
intensity_LEDwhtb = dataMatrix(slider_position, 12);

intensity_array = [intensity_LED470, intensity_LED505,
intensity_LED525, intensity_LED640, intensity_LEDwhty,
intensity_LEDwhtb];
intensity_max = max(intensity_array);

intensity_LED470 = round((intensity_LED470 / intensity_max) *
intensity_light * 1000) / 1000;
intensity_LED505 = round((intensity_LED505 / intensity_max) *
intensity_light * 1000) / 1000;
intensity_LED525 = round((intensity_LED525 / intensity_max) *
intensity_light * 1000) / 1000;
intensity_LED640 = round((intensity_LED640 / intensity_max) *
intensity_light * 1000) / 1000;
intensity_LEDwhty = round((intensity_LEDwhty / intensity_max) *
intensity_light * 1000) / 1000;
intensity_LEDwhtb = round((intensity_LEDwhtb / intensity_max) *
intensity_light * 1000) / 1000;

```

```

CRI_T = dataMatrix(slider_position, 13);
LmW_T = dataMatrix(slider_position, 14);

% setting the label handles with the intensities of each LED, CRI and
lm/W
set(handles.ILED470, 'String', intensity_LED470);
set(handles.ILED505, 'String', intensity_LED505);
set(handles.ILED525, 'String', intensity_LED525);
set(handles.ILED640, 'String', intensity_LED640);
set(handles.ILEDwhty, 'String', intensity_LEDwhty);
set(handles.ILEDwhtb, 'String', intensity_LEDwhtb);
set(handles.CRI, 'String', dataMatrix(slider_position, 13));
set(handles.LmW, 'String', dataMatrix(slider_position, 14));

% ARDUINO pin controller
global a;
a.analogWrite(3, round(intensity_LED640 * 255));
a.analogWrite(5, round(intensity_LED505 * 255));
a.analogWrite(6, round(intensity_LEDwhtb * 255));
a.analogWrite(9, round(intensity_LED470 * 255));
a.analogWrite(10, round(intensity_LED525 * 255));
a.analogWrite(11, round(intensity_LEDwhty * 255));

% --- Executes during object creation, after setting all properties.
function intensity_CreateFcn(hObject, eventdata, handles)
% hObject    handle to intensity (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: intensity controls usually have a light gray background.
if isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', [.9 .9 .9]);
end

% --- Executes on slider movement.
function temperature_Callback(hObject, eventdata, handles)
% hObject    handle to temperature (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'Value') returns position of slider
%        get(hObject, 'Min') and get(hObject, 'Max') to determine range
of slider

% getting the temperature from the temp slider with the tag
"temperature"
temperature_slider = (get(handles.temperature, 'Value') + 1) * 1000;
temperature_text = num2str(round(temperature_slider / 100) * 100);
set(handles.temperature_label, 'String', temperature_text);

% reading the file locus.dat -> real coordinates of the planckian locus
[x, y] = textread('locus.dat');
% coordinate transformation between the real and the image planckian
locus
[x_PL] = round(1228*[x] + 115);
[y_PL] = round(-1228*[y] + 1155);

```

```

% changing the color depending on the temperature
% slider_position gives the position of the slider
slider_position = int16(temperature_slider/100 - 9);

% cla clears the current axis
cla;
% refreshing the whole graphic when the slider changes its value
CIE = 'CIE1931.png';
imshow(CIE, 'Parent', handles.axes1);
hold on;
grid on;
plot(x, y, 'k-');
% this line controls the position of a given T on the graph
plot(x(slider_position), y(slider_position), 'k+');
axis([0 .8 0 0.9]);
%hold off;

% loading and reading the image
CIE_img = imread(CIE);
% getting the 3 RGB components depending on the temperature
RGB_locus = (CIE_img(y_PL(slider_position), x_PL(slider_position), :));

R_locus = double(CIE_img(y_PL(slider_position), x_PL(slider_position),
1));
G_locus = double(CIE_img(y_PL(slider_position), x_PL(slider_position),
2));
B_locus = double(CIE_img(y_PL(slider_position), x_PL(slider_position),
3));

% scaling the RGB components of the selected temperature
R_locus = (R_locus/255.);
G_locus = (G_locus/255.);
B_locus = (B_locus/255.);

% changing the color of the color_locus handle
set(handles.RGB_locus_label, 'String', RGB_locus);
set(handles.color_locus, 'Color', [R_locus G_locus B_locus])

% dataMatrix is a matrix that contains the intensities of the different
LEDs,
% CRI and Lm/W for each temperature
dataMatrix = dlmread('parametersTemp5.txt', '\t');

% reading and setting the variables related with the intensity slider
intensity_light = round(get(handles.intensity, 'Value') * 10) / 10;
intensity_text = num2str(round(intensity_light * 10) / 10);
set(handles.intensity_label, 'String', intensity_text);

% setting the different intensities with the parameters of the
dataMatrix
% the intensity of each LED is multiplied by the value of the slider
intensity
% intensity_total = dataMatrix(slider_position, 4) +
dataMatrix(slider_position, 5) + dataMatrix(slider_position, 6) +
dataMatrix(slider_position, 9) + dataMatrix(slider_position, 11) +
dataMatrix(slider_position, 12);

intensity_LED470 = dataMatrix(slider_position, 4);
intensity_LED505 = dataMatrix(slider_position, 5);
intensity_LED525 = dataMatrix(slider_position, 6);

```

```

intensity_LED640 = dataMatrix(slider_position, 9);
intensity_LEDwhty = dataMatrix(slider_position, 11);
intensity_LEDwhtb = dataMatrix(slider_position, 12);

intensity_array = [intensity_LED470, intensity_LED505,
intensity_LED525, intensity_LED640, intensity_LEDwhty,
intensity_LEDwhtb];
intensity_max = max(intensity_array);

intensity_LED470 = round((intensity_LED470 / intensity_max) *
intensity_light * 1000) / 1000;
intensity_LED505 = round((intensity_LED505 / intensity_max) *
intensity_light * 1000) / 1000;
intensity_LED525 = round((intensity_LED525 / intensity_max) *
intensity_light * 1000) / 1000;
intensity_LED640 = round((intensity_LED640 / intensity_max) *
intensity_light * 1000) / 1000;
intensity_LEDwhty = round((intensity_LEDwhty / intensity_max) *
intensity_light * 1000) / 1000;
intensity_LEDwhtb = round((intensity_LEDwhtb / intensity_max) *
intensity_light * 1000) / 1000;
CRI_T = dataMatrix(slider_position, 13);
LmW_T = dataMatrix(slider_position, 14);

% setting the label handles with the intensities of each LED, CRI and
lm/W
set(handles.ILED470, 'String', intensity_LED470);
set(handles.ILED505, 'String', intensity_LED505);
set(handles.ILED525, 'String', intensity_LED525);
set(handles.ILED640, 'String', intensity_LED640);
set(handles.ILEDwhty, 'String', intensity_LEDwhty);
set(handles.ILEDwhtb, 'String', intensity_LEDwhtb);
set(handles.CRI, 'String', dataMatrix(slider_position, 13));
set(handles.LmW, 'String', dataMatrix(slider_position, 14));

% ARDUINO pin controller
global a;
a.analogWrite(3, round(intensity_LED640 * 255));
a.analogWrite(5, round(intensity_LED505 * 255));
a.analogWrite(6, round(intensity_LEDwhtb * 255));
a.analogWrite(9, round(intensity_LED470 * 255));
a.analogWrite(10, round(intensity_LED525 * 255));
a.analogWrite(11, round(intensity_LEDwhty * 255));

% writting the file
global fileID;
arrayDataRowNorm = [str2num(temperature_text) intensity_LED470
intensity_LED505 intensity_LED525 intensity_LED640 intensity_LEDwhty
intensity_LEDwhtb CRI_T LmW_T R_locus*255. G_locus*255. B_locus*255.]
fprintf(fileID, '%6.1f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%4.1f,%6.3f,
%5.1f,%5.1f,%5.1f\n', arrayDataRowNorm);

% --- Executes during object creation, after setting all properties.
function temperature_CreateFcn(hObject, eventdata, handles)
% hObject    handle to temperature (see GCBO)
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

```

#### **C4. Instruccions per descarregar l'aplicació per a dispositius Android**

Degut a que l'*App Inventor* no permet exportar el codi font escrit a un format diferent a l'utilitzat pel propi programa, s'ha optat per donar accés lliure al codi a través d'un fitxer que pot ser carregat a l'*App Inventor*. Es pot descarregar a l'adreça:

<http://adriahf.comuv.com/LED.aia>

Un cop descarregat, s'ha d'anar a l'adreça <http://ai2.appinventor.mit.edu/> i efectuar el procés d'entrada. Seguidament, a la pestanya *Project* dirigir-se a l'apartat *Import project (.aia) from my computer* i carregar l'arxiu descarregat. D'aquesta manera, visualitzant la pestanya *Designer* es pot veure l'entorn gràfic i a la pestanya *Blocks* s'hi pot trobar el codi font utilitzat.

Per exportar el codi per fer-lo utilitzable per un dispositiu mòbil amb sistema operatiu Android es poden utilitzar les diferents opcions que proporciona la pestanya *Build*.

## C5. Comunicació de l'Arduino amb sistemes Android (C)

```
// relacionem els pins de l'Arduino amb els diferents LEDs
#define pinLED640 3
#define pinLED525 5
#define pinLED505 6
#define pinLED470 9
#define pinLEDwhty 10
#define pinLEDwhtb 11

// item son els caracters individuals que rebem del bluetooth
char item;
// cadena es la concatenacio dels diferents items formant un string
String cadena;
// controlAdd es una variable que controla l'addicio d'items a la
cadena
int controlAdd = 0;

// nombre total de LEDs
#define numLEDs 6
// individualLED es un string relacionat amb com agrupem les dades
String individualLED;
// definim l'array de floats que controlara els pins normalitzat a 1
float LEDs[numLEDs + 1];
// k es una variable relacionada amb l'agrupacio de dades
int k = 0;

// funcio setup per definir el comportament dels diferents components
void setup() {
  pinMode(pinLED640, OUTPUT);
  pinMode(pinLED525, OUTPUT);
  pinMode(pinLED505, OUTPUT);
  pinMode(pinLED470, OUTPUT);
  pinMode(pinLEDwhty, OUTPUT);
  pinMode(pinLEDwhtb, OUTPUT);
  Serial.begin(9600);
}

// loop es el bucle principal
void loop() {
  bluetoothToFloat();
  analogWrite(pinLED640, LEDs[0]*255);
  analogWrite(pinLED525, LEDs[1]*255);
  analogWrite(pinLED505, LEDs[2]*255);
  analogWrite(pinLED470, LEDs[3]*255);
  analogWrite(pinLEDwhty, LEDs[4]*255);
  analogWrite(pinLEDwhtb, LEDs[5]*255);
}

float strToFloat(String str){
  char carray[str.length() + 1]; //determine size of the array
  str.toCharArray(carray, sizeof(carray)); //put str into an array
  return atof(carray);
}

void bluetoothToFloat() {
```

```

// cada cop que arribi una dada pel bluetooth
if (Serial.available()) {
  // llegeix la dada (es un char)
  item = Serial.read();
  // comencem a parsejar les dades entrants
  if (item == '(' || controlAdd == 1) {
    // agrupem les dades a la cadena
    cadena = cadena + item;
    // controlAdd = 1 -> parsegem
    controlAdd = 1;
  }
  // ) es la dada que tanca l'string
  if (item == ')') {
    // k es la variable relacionada amb el num total de LEDs. La iniciem
    per fer el loop
    k = 0;
    // per cada element de la cadena generada
    for (int i = 0; i < cadena.length(); i++) {
      // mirem els espais
      if (cadena[i] == ' ') {
        for (int j = 1; j <= 6; j++) {
          // agafem les 6 xifres de despres dels diferents espais i les
          agrupem
          // a la variable individualLED
          individualLED = individualLED + cadena[i+j];
        }
        if (k <= numLEDs) {
          // passem a float les 6 primeres (corresponents als LEDs)
          LEDs[k] = strToFloat(individualLED);
        }
        // reiniciem la variable individualLED
        individualLED = "";
        // augmentem el comptador k
        k = k++;
      }
    }
    Serial.println(cadena);
    // reiniciem la variable cadena
    cadena = "";
    // reiniciem la variable controlAdd
    controlAdd = 0;
    // imprimim els elements de l'array per pantalla (ja com a floats)
    for (int l = 0; l < 6; l++) {
      Serial.println(LEDs[l]);
    }
  }
}
}
}
}

```



## C6. Càlculs teòrics d'eficiència (Python 2.7.)

```
# llegir l'arxiu generat
fileRead = open('parametersTempCSV.txt', 'r')
# gravar les dades a un nou fitxer
fileWrite = open('dadesGraficEficiencia.dat', 'w')
# s'utilitza la funcio readline per llegir el fitxer
linia = fileRead.readline()

# eficiencies dels diferents LED (lm/W)
eficienciaLED1 = 38
eficienciaLED2 = 75
eficienciaLED3 = 93
eficienciaLED4 = 75
eficienciaLED5 = 93
eficienciaLED6 = 107
# caigudes de potencial dels diferents LED (V)
caigudaVLED1 = 3.3
caigudaVLED2 = 3.25
caigudaVLED3 = 3.25
caigudaVLED4 = 2.9
caigudaVLED5 = 2.81
caigudaVLED6 = 3.

# loop principal: llegeix les línies una per una
while linia != '':
    # converteix la línia en una llista
    linia = linia.split(',')
    # assignacio de variables
    # temperatura (K)
    temperatura = int(float(linia[0]))
    # intensitat LED (unitats normalitzades)
    intensitatLED1 = float(linia[1])
    intensitatLED2 = float(linia[2])
    intensitatLED3 = float(linia[3])
    intensitatLED4 = float(linia[4])
    intensitatLED5 = float(linia[5])
    intensitatLED6 = float(linia[6])
    CRI = linia[7]
    # caiguda de potencial (V)
    caigudaV = 5
    # factor de normalitzacio
    intensitat = 40
    # potencia consumida per cada LED (W)
    potenciaLED1 = caigudaV * intensitatLED1 * intensitat / 1000.
    potenciaLED2 = caigudaV * intensitatLED2 * intensitat / 1000.
    potenciaLED3 = caigudaV * intensitatLED3 * intensitat / 1000.
    potenciaLED4 = caigudaV * intensitatLED4 * intensitat / 1000.
    potenciaLED5 = caigudaV * intensitatLED5 * intensitat / 1000.
    potenciaLED6 = caigudaV * intensitatLED6 * intensitat / 1000.
    # flux lluminos emes per cada LED (lm)
    fluxLED1 = eficienciaLED1 * caigudaVLED1 * intensitatLED1 *
        intensitat / 1000.
    fluxLED2 = eficienciaLED2 * caigudaVLED2 * intensitatLED2 *
        intensitat / 1000.
```

```

fluxLED3 = eficienciaLED3 * caigudaVLED3 * intensitatLED3 * intensitat /
1000.
fluxLED4 = eficienciaLED4 * caigudaVLED4 * intensitatLED4 * intensitat /
1000.
fluxLED5 = eficienciaLED5 * caigudaVLED5 * intensitatLED5 * intensitat /
1000.
fluxLED6 = eficienciaLED6 * caigudaVLED6 * intensitatLED6 * intensitat /
1000.
# potencia total consumida pels LED
potenciaTotalLED = round(potenciaLED1 + potenciaLED2 + potenciaLED3
+ potenciaLED4 + potenciaLED5 + potenciaLED6, 2)
# flux lluminos total emes (lm)
fluxLluminosTotal = round(fluxLED1 + fluxLED2 + fluxLED3 + fluxLED4 +
fluxLED5 + fluxLED6, 1)
# potencia consumida Arduino (W)
potenciaArduino = 0.125
# potencia total
potenciaTotal = potenciaTotalLED + potenciaArduino
# eficiencia de la llum
eficienciaLlum = round(fluxLluminosTotal / potenciaTotalLED, 1)
# eficiencia total
eficienciaTotal = round(fluxLluminosTotal / (potenciaTotalLED +
potenciaArduino), 1)
# afegim xifres
if len(str(potenciaTotalLED)) == 3:
    potenciaTotalLED = str(potenciaTotalLED) + '0'
if len(str(fluxLluminosTotal)) == 3:
    fluxLluminosTotal = str(fluxLluminosTotal) + '0'
# escriue el resultat al fitxer
fileWrite.write(str(temperatura) + '\t' + str(potenciaTotalLED)
+ '\t' + str(potenciaTotal) + '\t' + str(fluxLluminosTotal)
+ '\t' + str(eficienciaLlum) + '\t' + CRI + '\t' +
str(eficienciaTotal) + '\n')
# llegeix la linia següent
linia = fileRead.readline()

fileRead.close()
fileWrite.close()

```

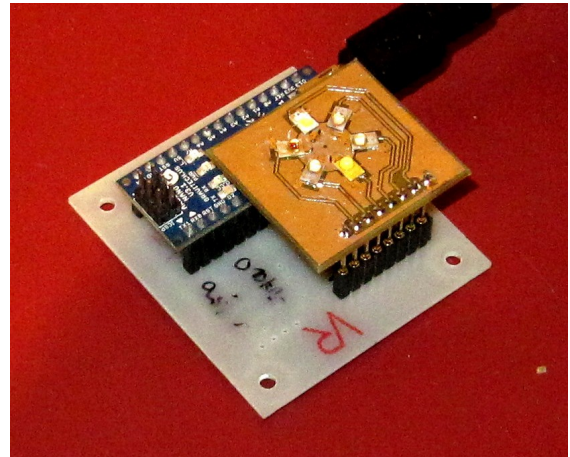
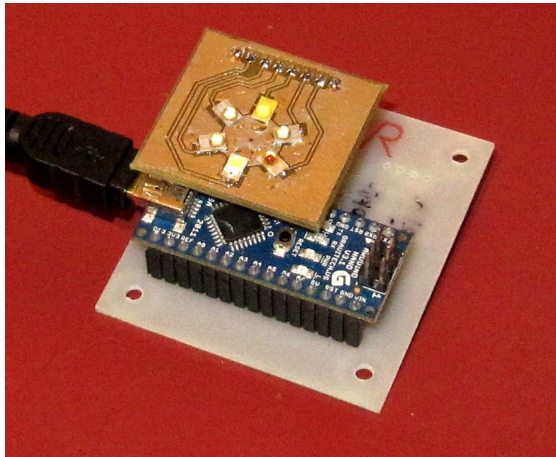


## D2. Simulació realitzada per obtenir l'eficiència teòrica del sistema

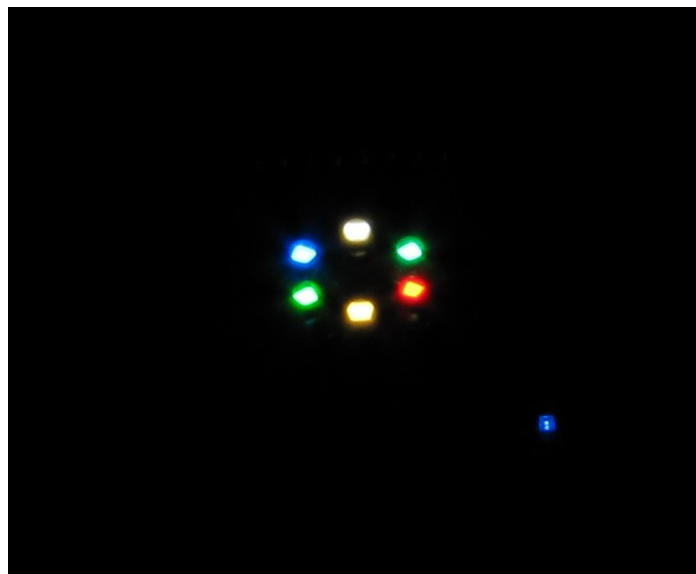
T(K)	PtL(W)	Pt(W)	Ft(lm)	E11	CRI	Et(lm/W)
1000	0.33	0.455	13.9	42.1	79.0	30.5
1100	0.30	0.425	13.0	43.3	82.6	30.6
1200	0.29	0.415	12.4	42.8	85.8	29.9
1300	0.27	0.395	11.8	43.7	88.5	29.9
1400	0.26	0.385	11.4	43.8	90.8	29.6
1500	0.25	0.375	11.1	44.4	92.8	29.6
1600	0.25	0.375	10.8	43.2	94.4	28.8
1700	0.24	0.365	10.6	44.2	95.5	29.0
1800	0.24	0.365	10.4	43.3	96.0	28.5
1900	0.23	0.355	10.3	44.8	95.9	29.0
2000	0.23	0.355	10.1	43.9	95.3	28.5
2100	0.22	0.345	10.0	45.5	94.6	29.0
2200	0.22	0.345	9.90	45.0	93.8	28.7
2300	0.36	0.485	17.1	47.5	93.5	35.3
2400	0.22	0.345	9.90	45.0	94.5	28.7
2500	0.22	0.345	10.0	45.5	95.1	29.0
2600	0.23	0.355	10.0	43.5	95.6	28.2
2700	0.23	0.355	10.1	43.9	96.0	28.5
2800	0.23	0.355	10.2	44.3	96.4	28.7
2900	0.24	0.365	10.3	42.9	96.6	28.2
3000	0.24	0.365	10.5	43.8	96.7	28.8
3100	0.25	0.375	10.9	43.6	96.8	29.1
3200	0.26	0.385	11.3	43.5	96.8	29.4
3300	0.30	0.425	13.5	45.0	97.5	31.8
3400	0.28	0.405	12.4	44.3	96.8	30.6
3500	0.29	0.415	13.0	44.8	96.8	31.3
3600	0.31	0.435	13.7	44.2	96.8	31.5
3700	0.32	0.445	14.4	45.0	96.7	32.4
3800	0.34	0.465	15.2	44.7	96.6	32.7
3900	0.36	0.485	16.0	44.4	96.5	33.0
4000	0.38	0.505	16.9	44.5	96.3	33.5
4100	0.40	0.525	17.9	44.7	96.2	34.1
4200	0.51	0.635	23.2	45.5	96.3	36.5
4300	0.53	0.655	23.9	45.1	96.0	36.5
4400	0.47	0.595	21.4	45.5	95.6	36.0
4500	0.50	0.625	22.8	45.6	95.4	36.5
4600	0.52	0.645	23.7	45.6	95.2	36.7
4700	0.50	0.625	22.5	45.0	94.9	36.0
4800	0.47	0.595	21.5	45.7	94.7	36.1
4900	0.45	0.575	20.6	45.8	94.4	35.8
5000	0.43	0.555	19.8	46.0	94.2	35.7
5100	0.42	0.545	19.1	45.5	93.9	35.0
5200	0.42	0.545	19.0	45.2	94.5	34.9
5300	0.39	0.515	18.0	46.2	94.3	35.0
5400	0.37	0.495	17.2	46.5	94.1	34.7
5500	0.36	0.485	16.7	46.4	93.9	34.4
5600	0.35	0.475	16.2	46.3	93.7	34.1
5700	0.34	0.465	15.8	46.5	93.5	34.0
5800	0.33	0.455	15.4	46.7	93.4	33.8
5900	0.33	0.455	15.3	46.4	93.2	33.6
6000	0.33	0.455	15.3	46.4	93.1	33.6
6100	0.33	0.455	15.2	46.1	93.0	33.4
6200	0.35	0.475	15.9	45.4	92.9	33.5
6300	0.30	0.425	14.0	46.7	92.1	32.9
6400	0.29	0.415	13.7	47.2	91.9	33.0
6500	0.29	0.415	13.5	46.6	91.6	32.5
6600	0.32	0.445	14.9	46.6	92.2	33.5
6700	0.29	0.415	13.5	46.6	91.3	32.5
6800	0.29	0.415	13.4	46.2	91.2	32.3
6900	0.29	0.415	13.4	46.2	91.0	32.3
7000	0.29	0.415	13.4	46.2	90.9	32.3
7100	0.29	0.415	13.4	46.2	90.8	32.3
7200	0.29	0.415	13.4	46.2	90.6	32.3
7300	0.29	0.415	13.4	46.2	90.5	32.3
7400	0.29	0.415	13.4	46.2	90.4	32.3
7500	0.29	0.415	13.4	46.2	90.3	32.3
7600	0.29	0.415	13.4	46.2	90.2	32.3
7700	0.29	0.415	13.4	46.2	90.0	32.3
7800	0.29	0.415	13.4	46.2	89.9	32.3
7900	0.29	0.415	13.4	46.2	89.8	32.3
8000	0.29	0.415	13.4	46.2	89.7	32.3

## Annex E. Fotografies del prototip

### Annex E1. Microcontrolador, plaques de connexions i LED



**Figura A2.** Fotografia del microcontrolador, plaques de connexions i LED utilitzats.



**Figura A3.** Disposició dels diferents LED utilitzats.

## Annex E2. Fotografies del sistema en funcionament



(a)



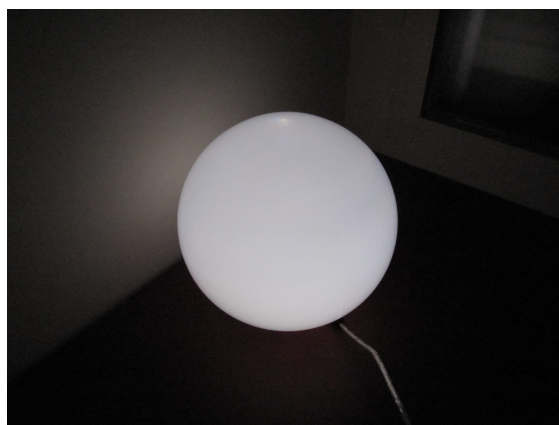
(b)



(c)



(d)



(e)

**Figura A4.** Fotografies del sistema en funcionament a (a) 3400K, (b) 4400K, (c) 5400K, (d) 6400K i (e) 7400K.

