UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

"DISEÑO DE UN DISPOSITIVO PARA EL ESTUDIO CUANTITATIVO DE CONDUCTAS EVOCADAS POR LUZ EN LA MOSCA *DROSOPHILA*"

TESIS

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO ELÉCTRICO-ELECTRÓNICO

PRESENTAN:

CLARA SUSANA YÁÑEZ MARTÍNEZ
GIOVANNI FONSECA FONSECA

ASESOR: DR. FRANCISCO FERNÁNDEZ DE MIGUEL

México, D.F., Octubre 2007

Agradecimientos:

- A la Secretaría de Desarrollo Institucional y la Dirección General de CCH, UNAM por su apoyo en la realización de este proyecto universitario.

- Al Dr. Francisco Fernández de Miguel y al M.I. Juan Manuel Gómez González, por la confianza que demostraron siempre hacia nosotros y por su guía en la desarrollo de este proyecto.

- A los miembros del jurado: Ing. Gabriel Alejandro Jaramillo Morales, M.I. Antonio Salvá Calleja e Ing. Rodolfo Peters Lammel por su tiempo y dedicación en la revisión de este trabajo así como sus agudos comentarios y sugerencias.

- Al Dr. Juan Hernández Cordero por sus asesorías en el tema “fuentes de luz” de este proyecto.

- A la Dra. Martha Rossete y el Sr. Valentín López Cabañas por su participación en el diseño óptico y la manufactura de la cámara de registro respectivamente.

- Al Ing. Bruno Méndez Ambrosio por su ayuda con la técnica de serigrafía realizada en la manufactura del circuito impreso del dispositivo.

- A la Universidad Nacional Autónoma de México y su Facultad de Ingeniería por la formación académica y cultural brindada.

"Por mi Raza Hablará el Espíritu"

Ciudad Universitaria, México, D.F., Octubre, 2007

Agradecimientos personales:

A mis padres: **Yolanda Martínez** y **Roberto Yáñez**, gracias a ambos por su cariño, cuidados, apoyo, comprensión, por alentarme a ser mejor cada día y por enseñarme cada una de las cosas que me han hecho llegar hasta este momento. A ti Yolis por tu fuerza, entereza y ejemplo de madre y mujer trabajadora. A ti Roberto por la ayuda y apoyo y esos pequeños pero acertados consejos.

A mis hermanos: **Bibianita** por tus consejos, apoyo, ánimos y ocurrencias. A **Carlos** por los buenos momentos.

A mi familia: **Mi abuelita**, **Mis tíos**: Lupe, Cuco, Chucho, Toño, **Mis primos** Jimenita y Lalo por estar siempre cerca, brindarme su ejemplo y apoyo a cada momento, y a todos los demás miembros, por que en conjunto me enseñaron a valorar y disfrutar lo que significa una familia.

A **Giovanni** por todos los conocimientos que me transmitió a lo largo de este tiempo, por su trabajo y sus ideas.

A mis amigos:

Mon por su amistad incondicional, por compartir tantas cosas, por escucharme, apoyarme, por su complicidad, por su paciencia y por todos sus ánimos.

Xim, Christian, Osve, Compañerito, por la valiosa y gran amistad que nos une y todos esos momentos divertidos que hemos pasados juntos.

Avril y Geral por escucharme y por cada una de las palabras que me han brindado.

León, David, Luis y Alejandro por creer en mí y preocuparse por verme feliz.

A **Daniel** por su cariño, paciencia, confianza, por creer en mí, por apoyarme, por darle un giro a mi vida y llenarla de cariño.

Con mucho cariño para cada uno de ustedes

Susana

Dedicatoria de Giovanni

Este trabajo se lo dedico a mis padres: Miguel Antonio Fonseca Posada✝, a quien le hubiera dado mucho gusto y orgullo haber estado físicamente en mi examen profesional pero sé que está en el corazón de todos los seres que lo quisimos; y Guadalupe Aurora Fonseca Sánchez de Aparicio, quien ha sido la persona de la que más amor puro he recibido y a la que le debo la vida entera. A mis hermanos, Guadalupe, Madeleine y Miguel, que en muchos momentos fueron como padres para mí.

Agradecimientos de Giovanni

En el transcurso de mi vida he conocido a personas muy valiosas y de todas he tratado de aprender siempre lo más importante de su ser. He sido una esponja de buenos y malos ejemplos y el resultado es lo que soy actualmente.

A mi padre le debo agradecer el que me enseñara a pensar, a aprender y a transmitir mi conocimiento. Él me dio todas las herramientas académicas y racionales que hoy tengo para desarrollarme en mi ámbito profesional. Le agradezco todo el esfuerzo que hizo para darme la mejor educación que pude haber recibido y sobre todo, la dedicación que tuvo apoyándome en mi desarrollo académico. Gracias a él puedo decir que soy ingeniero eléctrico electrónico desde tercero de Kinder, cuando me preparó para dar una conferencia acerca de electricidad a mis compañeros de la escuela, donde mi padre y mis hermanos construyeron una maqueta donde se mostraba un rayo, que impresionó a todos mis compañeros; el flujo de electrones, que simulaba una hilera de LED's que se encendían consecutivamente; y dos circuitos para explicar la diferencia entre conectar dispositivos en serie y en paralelo. Jamás olvidaré lo bien que me sentí cuando mis compañeros me aplaudieron…Gracias Papá.

A mi madre le quiero agradecer todo ese amor incondicional, toda la comprensión, toda la fuerza que demostró en los momentos difíciles y toda la devoción que ha tenido hacia sus hijos y nietos. Sin su ejemplo jamás podría haber vivido los momento más alegres de mi vida, porque jamás hubiera sabido como amar a mis seres queridos. Madre, gracias por vivir para nosotros.

A mi hermano Miguel, le debo agradecer el gusto que me inculcó por la física y las matemáticas, hoy en día el disfrute y placer que siento al momento de hacer ingeniería me recuerda su entusiasmo y pasión cuando me explicaba cómo hacer mis tareas de la prepa. Gracias Hermano, has sido como un padre para mí.

A mi hermana Guadalupe le agradezco haber tomado las riendas de la casa cuando tuvo que hacerlo, ha sido un ejemplo de madurez y valor para mí. Gracias por darme a los sobrinos más maravillosos que pude haber tenido.

A mi hermana Madeleine, quiero agradecerle todo el cariño, y amor que me ha demostrado toda su vida. Es una de las personas más inteligentes y nobles que he conocido, estoy seguro que la vida le tendrá una gran recompensa por haber sido la más especial de las hermanas conmigo. Te adoro Made.

Particularmente debo hacer un agradecimiento a un par de personas que marcaron mi vida de dos diferentes formas pero que de ambas aprendí muchísimo: la ética de mi abuelito Paco y el afecto interminable de mi abuelita Paca.

Esta sección podría ser infinita si agradeciera a todos los compañeros, que durante mi vida se han dado a la tarea de regalarme su afecto, su apoyo, y sobre todo su amistad. No tengo con qué

pagarles los momentos tan especiales que hemos pasado juntos. Sólo puedo agradecerle a la vida el que me hiciera toparme con ellos, porque han sido importantísimos para mi desarrollo personal.

Con algunos de estos amigos me ha tocado vivir experiencias de esas que te marcan y que te hacen crecer como persona, convirtiéndolos en mis hermanos adoptivos; a ellos les agradezco infinitamente su fraternidad y cariño.

Buena parte de mi éxito profesional se lo debo a muchos de los profesores que durante mi vida me fueron dejando la semillita del conocimiento y de su propia experiencia, haciendo que ahora pueda tener una cosecha de la que seguro podré alimentarme el resto de mi vida. A todos aquellos profesores de los que aprendí lo que se debía y lo que no se debía hacer, mil gracias.

Por último les agradezco a todas las personas que estuvieron involucradas en la realización de este proyecto. He aprendido mucho realizando esta Tesis y eso es invaluable para mí. Gracias a todos.

Índice

- **Resumen** ……..……1
- **Marco teórico**
  - Capítulo I.- La Mosca *Drosophila*……3
  - Capítulo II.- Microcontroladores……...6
- **Desarrollo**
  - Capítulo III.- Diseño.....11
    - Análisis del Problema.....11
    - Alternativas de Solución.....11
    - Fuentes de Luz Fría (LED).....13
      - Proceso de Caracterización.....14
    - La Cámara de Registro.....19
    - La Electrónica (Hardware).....20
    - La Interfase con el Usuario GUI (Software).....27
    - Cuantificación de la Intensidad Luminosa.....30
- **Resultados**
  - Capítulo IV.- Resultados Experimentales.....33
- **Discusión**
  - Capítulo V.- Docencia e Investigación.....44
- **Conclusiones y Recomendaciones**.....46
- **Apéndices**
    - A.- Arreglo Óptico 48
    - B.- Especificaciones y Requerimientos del dispositivo.....49
    - C.- PCB para Circuito Impreso.....50
    - D.- Programa del PIC……52
    - E.- Datasheets……59
    - F.- Programa de la GUI en MATLAB……..68
    - G.- Cuantificación de Intensidad Luminosa ……..…81
    - H.- Gráficas Resultados……87
    - I.- Cotización del Dispositivo……...92
- **Bibliografía**……93

Resumen

La presente tesis consiste en el diseño, desarrollo, implementación y prueba de un dispositivo para la cuantificación de conductas evocadas por luz en la mosca Drosophila.

La importancia de este proyecto radica en las posibilidades de este dispositivo para cuantificar conductas de un ser vivo que se utiliza con gran éxito en investigación científica en el área de la genética. Con ello es posible estudiar las bases genéticas de ciertas conductas. Creando un aparato robusto y versátil que puede ser utilizado como material didáctico para la enseñanza experimental de la ciencia.

Un reto a superar en el diseño del dispositivo fue minimizar variables ambientales que afectan la conducta de las moscas y hacer que ésta sea consecuencia principalmente de la luz. Analizamos diferentes posibilidades de estimular a la mosca por medio de diversas fuentes de luz y encontramos que la más adecuada fue la producida por los LED's.

El paso siguiente consistió en controlar las variaciones en la intensidad luminosa de cada una de las fuentes de iluminación. La propuesta que hacemos es realizar el control por medio de un "cerebro electrónico": un microcontrolador, el cual gobierna a un potenciómetro digital que es el circuito integrado encargado de aumentar o disminuir la intensidad luminosa de cada uno de los LED's por medio de pulsos.

El dispositivo diseñado tiene una gran versatilidad en cuanto a los parámetros que el usuario puede definir en su experimento: tiempo de adaptación a la obscuridad, tiempo de exposición a la luz, una gama de 20 diferentes intensidades para cada color y cinco colores diferentes para realizar todas las combinaciones posibles en pruebas de atracción, huida o selección.

Una parte fundamental del diseño y desarrollo de este proyecto es la posibilidad de comunicarlo con una computadora, sin necesidad de que ésta cuente con recursos específicos de hardware, dado que la comunicación se realiza por medio del puerto serie. La única condición es que la computadora cuente con Matlab dado que la interfase gráfica con el usuario se programó bajo esta plataforma.

Esta interfase gráfica se desarrolló de manera que fuera sencilla de manipular por el usuario y tiene los elementos mínimos indispensables para ofrecer una gráfica con información necesaria para analizar la conducta de la mosca.

Para monitorear el movimiento de la mosca, en el interior de la cámara de registro se implementó un sistema de cuatro optoacopladores de infrarrojo a lo largo de un tubo que desemboca a cada extremo en las fuentes de iluminación. Cada optoacoplador ofrece información cada 0.25 segundos acerca de la posición de la mosca en la cámara.

En el proceso de investigación y desarrollo tecnológico participaron diferentes profesionistas que nos orientaron para proponer la mejor solución a los diversos problemas que se fueron presentando en la elaboración de este proyecto.

Este trabajo es una muestra más de que en México se pueden hacer desarrollos tecnológicos muy útiles así como específicos que contribuyan al avance científico y educativo de nuestro país.

Marco Teórico

Capítulo I.- La Mosca *Drosophila*

La mosca *Drosophila melanogaster* es la pequeña mosca de la fruta **[Fig. 1.1a].** Ésta pertenece al orden Díptera, que agrupa a los organismos en los que el primer par de alas es funcional y el segundo se ha transformado en órganos de equilibrio llamados salterios o balancines **[3]**.

Los primeros estudios con esta especie fueron realizados por Castle en 1901 y posteriormente en 1909 por Morgan para experimentos de genética, a partir de los cuales se ha convertido en una especie ampliamente utilizada para trabajos experimentales debido a las ventajas que presenta como su fácil manejo, su tamaño que oscila entre 1 mm y 1.5 mm de longitud, su rápida reproducción produciendo una nueva generación cada dos semanas. Sus características anatómicas y estructurales, su transmisión de caracteres hereditarios de generación en generación y la gran diversidad de mutantes que existen la han hecho un animal excepcional para el estudio de la expresión genética y el desarrollo. Gracias a esto, ha aportado una gran cantidad de información al conocimiento de los mecanismos de regulación de la expresión genética y a descubrimientos fundamentales como por ejemplo la existencia de los genes homeóticos, cuya expresión influye en la formación de segmentos o unidades funcionales completas. Además, se cuenta con un número enorme de mutantes de diferentes genes, lo que permite estudiar la importancia de su expresión en el desarrollo o en la función. En ese sentido, no solo es útil para estudios de genética, biología molecular y desarrollo, sino que también es ampliamente usada con fines didácticos en las mismas áreas.

Esta especie se encuentra en todo tipo de clima, altitud y latitud, localizándose especialmente en las frutas en que ha iniciado la fermentación y en algunos alimentos con alto contenido de ácido acético **[3]**.

Además de ofrecer grandes ventajas para estudios sobre genética también permite estudiar respuestas conductuales, como lo hace Edward A. Kravitz[1] con los estudios de agresión en esta especie, lo que abre un campo de interés que es el estudio de las bases genéticas de la conducta. La mosca *Drosophila* responde a estímulos como luz, olor, sabor, humedad, temperatura y gravedad por mencionar algunos.

[1] Kravitz Edwuard. 2002. Fighting fruit flies: A model system for the studyof aggression
Kravitz Edwuard. 2004. Gender-selective patterns of aggressive behavior in Drosophila melanogaster
Kravitz Edwuard. 2006. Learning and memory associated with aggression in Drosophila melanogaster

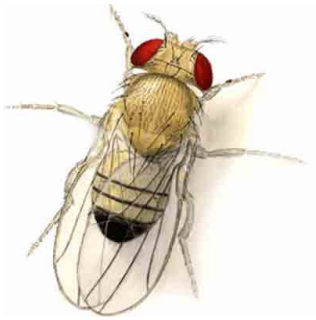
De los estímulos antes mencionados, la mosca presenta una alta sensibilidad a los visuales, respondiendo a éstos en un rango de longitudes de onda entre 350nm y 650nm; es decir, entre el ultravioleta y el rojo **[4]**.

Entre las respuestas que se pueden estudiar al estimular con luz a estos ejemplares, están las optomotoras que son las de coordinación de movimientos en respuesta a un estimulo visual.

Las respuestas del ojo a la luz han sido ampliamente estudiadas en vertebrados y en invertebrados, y de nuevo, las moscas *Drosophila* han contribuido al campo con información acerca de qué genes regulan la expresión de ciertas estructuras o proteínas. Sin embargo, se desconoce aún mucho acerca de la relación entre éstos y la conducta.

La mosca Drosophila al igual que otros insectos tiene ojos compuestos, formados por aproximadamente 800 omatidios **[5,2] [Fig. 1.1b]. C**ada uno es un arreglo concéntrico con 19 células, 11 de ellas llamadas células adjuntas y 8 células fotorreceptoras **[1]**, divididos los primeros 6 fotorreceptores R1-R6 en un grupo, R7 y R8 en otro grupo respectivamente, siendo el primer grupo quien rodea a las células fotorreceptoras distal (R7) y proximal (R8) al centro del arreglo de la omatidia **[1,5]**. Estas últimas células son fotorreceptores que perciben color, distribuidos aleatoriamente: 70% amarillo y 30% azul, captando el amarillo longitudes de onda largas desde el verde mientras que el caso del azul abarca longitudes de onda cortas en el rango del azul y UV **[2]**.

a) **b)**



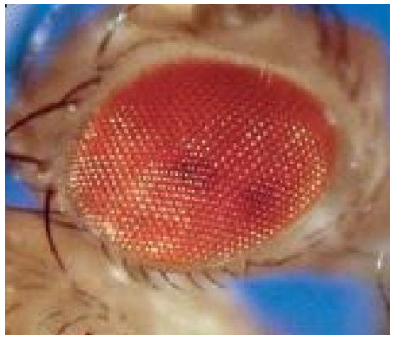


Fig. 1.1: a) Ejemplar de la Mosca *Drosophila melanogaster.* **b)** Ojo compuesto de la Mosca *Drosophila melanogaster*

Dispositivos para el estudio de la conducta en Drosophila

En el estudio del comportamiento de la mosca *Drosophila*, el laberinto en forma de “T” es el dispositivo más utilizado. Este permite estudiar respuestas basadas en la selección de estímulos como luz, sabor, olor o *electrochoques* ya que al introducir a las moscas justo en medio del laberinto se ofrece sólo dos

opciones; es decir, deben elegir entre uno de los dos brazos de la “T”. Cada brazo presenta una distribución del 50% de preferencia en ausencia de señales de atracción o repulsión **[4]**.

En moscas adultas, la respuesta a la luz ha sido medida usando dispositivos con contenedores transparentes en forma de embudo conectados por un pequeño túnel, en donde mediante infrarrojo se detecta el paso entre los depósitos, las moscas se dejan de un lado, alternadamente se iluminan los contenedores y es registrado el paso de las moscas. Otro tipo de medición similar es utilizando el laberinto en forma de “T”, en el cual un brazo es traslucido y el otro es obscurecido, permitiendo que las moscas elijan entre la luz o la obscuridad de los brazos de la “T” **[4]**.

Los dispositivos anteriores, sin embargo, no permiten tener un buen control de variables fundamentales para entender los procesos visuales. Éstas son, las cantidades y las longitudes de onda aplicadas, los tiempos y las condiciones de adaptación a la luz de los animales.

En este sentido, consideramos importante contar con un dispositivo que permita medir diferentes tipos de conductas, incluidas, las de atracción, huída, selección, adaptación a la luz y adaptación a la obscuridad, mediante un dispositivo versátil en el que cada conducta pueda medirse bajo condiciones controladas en el tiempo, como la intensidad de la luz y la longitud de onda. Con base en ello diseñamos y construimos el aparato objeto de la presente tesis.

Capítulo II.- Microcontroladores

Es importante tener antecedentes relacionados con el uso e implementación de microcontroladores en el diseño de aplicaciones digitales avanzadas debido a que las necesidades de este proyecto así lo exigieron, es por esto que a continuación hablaremos de los conceptos básicos relacionados con este tipo de dispositivos electrónicos.

Los microcontroladores son circuitos electrónicos de alta escala de integración que nos permiten controlar puertos de entrada y salida de información. Un microcontrolador cuenta con un "cerebro" llamado microprocesador, que se encarga de realizar las operaciones aritméticas y lógicas necesarias para que el microcontrolador lleve a cabo una cierta función. Este "cerebro" es el mismo microprocesador de las computadoras personales. Comercialmente existen varios fabricantes como Intel, Motorola, AMD, por mencionar algunos, los cuales tienen diversos modelos de microprocesadores como los muy conocidos Pentium (Intel), Dragonball (Motorola) y Athlon (AMD). Con lo antes dicho se puede entender la principal diferencia que existe entre un microprocesador y un microcontrolador. El primero, sólo procesa información; el segundo se aprovecha de este procesamiento para controlar puertos de entrada y salida. Además del "cerebro", el microcontrolador tiene dos tipos de unidades de memoria: volátil y no volátil.

La memoria volátil (RAM[2]) es aquella que utilizada por el microcontrolador para tener un registro temporal de información. La cantidad de memoria RAM nos indica el desempeño, ya que el hecho de facilitarle al microprocesador mayor cantidad de espacio para almacenar información temporal, en muchos de los casos agiliza el procesamiento mismo de la información.

La memoria no volátil (ROM[3]) es aquélla en la cual se graba la información que el microcontrolador necesita saber para poder hacer la función que el programador desea. Es decir, el programa que se diseña para que el microcontrolador efectúe la función deseada, es grabado en este tipo de memoria en el entendido de que no se borrará al momento de interrumpir la alimentación del microcontrolador. Hay varios tipos de memoria no volátil y cada microcontrolador puede tener uno en particular; el más utilizado es el EEPROM[4] (también conocido como E^2PROM) debido a su versatilidad y facilidad para grabar información en ella. Las PC's tienen un sistema de arranque indispensable para que la computadora comience a reconocer todos sus puertos de entrada y salida de información y así los pueda utilizar el sistema operativo posteriormente. Este sistema se conoce por el nombre de BIOS[5] el cual no puede ser alterado por el usuario de la computadora.

[2] Acrónimo de *Random Access Memory*.
[3] Acrónimo de *Read-Only Memory*.
[4] Acronimo de *Electrically-Erasable Programmable Read-Only Memory.*
[5] Acronimo de *Basic Input-Output System*.

Este sistema está grabado en una memoria no volátil de manera que al apagar la máquina, ésta no pierde dicha información.

Por último, un microcontrolador cuenta con varios puertos de entrada y salida de información, la cual es utilizada o procesada para "tomar decisiones" y enviar información a sus puertos de salida. Éstos a su vez son dispositivos periféricos que harán una determinada función para que las variables sean controladas por el microcontrolador. Los puertos de entrada en una computadora, son por ejemplo: el teclado, el micrófono, el *mouse*, un puerto infrarrojo o *bluetooth*, etc. En el caso de los puertos de salida de una computadora, hablamos del monitor, las bocinas, impresora, *scanner*, etc.

En resumen, podemos decir que un microcontrolador es un dispositivo electrónico que tiene la finalidad de controlar determinadas variables mediante el control de dispositivos periféricos. Para ello es necesario que el programador grabe en su memoria no volátil una serie de instrucciones que le indiquen al microcontrolador lo que tiene que hacer dependiendo de la información que recabe en sus puertos de entrada. Habiendo analizado dicha información, el microcontrolador le enviará las órdenes a sus puertos de salida para que éstos modifiquen las variables externas y así podamos estar hablando de un sistema controlado.

Puertos de Entrada y Salida

Los puertos son terminales físicas con las que cuentan los microcontroladores y a las cuales podemos conectar diferentes dispositivos. Los dispositivos son de entrada si le proporcionan información al microcontrolador o de salida si esperan que el microcontrolador les dé información para realizar su función.

Un microcontrolador puede tener dos tipos diferentes de puertos de entrada: analógicos y digitales. El primero consiste en una entrada acoplada a un convertidor analógico digital de *n* bits. Así, las señales analógicas son convertidas a lenguaje binario. Dependiendo del convertidor analógico digital (ADC, por sus siglas en inglés), el número en binario puede variar en su cantidad de bits. Los ADC's más comunes son de 8 bits, lo cual permite representar la señal analógica de entrada como 256 diferentes valores discretos.

Los puertos de entrada digitales, permiten el ingreso exclusivo de valores lógicos de entrada; es decir, "unos" o "ceros", definidos como la presencia o ausencia de tensión eléctrica.

Los puertos de salida de un microcontrolador sólo pueden ser digitales; es decir, la única salida que obtendremos de un microcontrolador será de tipo lógico, "unos" o "ceros". Estos puertos de

salida pueden ser en serie o en paralelo lo que dependerá del microcontrolador mismo. Cabe aclarar que puede haber microcontroladores integrados a un convertidor digital analógico de tal forma que podríamos tener salidas analógicas.

Arquitecturas Internas de los Microcontroladores

Anteriormente hemos dicho que los microcontroladores tienen unidades de memoria volátil y no volátil. Hay microcontroladores con diferentes arquitecturas internas; las cuales cambian dependiendo de cómo están distribuidos estos dos tipos de memoria.

La arquitectura Von Newman **[Fig. 2.1]** consta de un único módulo de memoria dividido de tal forma que una parte sea utilizada para las instrucciones de programa y otra para el almacenamiento temporal de información, de tal forma que el microprocesador tenga acceso a la información por medio de un *bus* de datos.

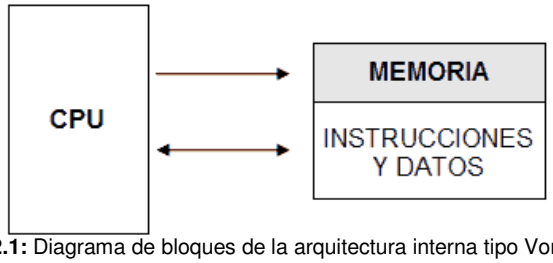


Fig. 2.1: Diagrama de bloques de la arquitectura interna tipo Von Newman

En la arquitectura Harvard **[Fig. 2.2]** la memoria de instrucciones y la de almacenamiento temporal de información (memoria de datos) son independientes y cada una cuenta con su sistema de *bus* para que el microprocesador tenga acceso. Esta dualidad, además de propiciar el paralelismo en el procesamiento de la información, permite una mejor adecuación del tamaño de palabra y los *buses* para los requerimientos específicos de las instrucciones y los datos.

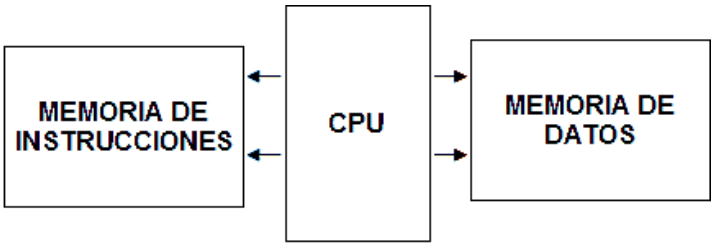


Fig. 2.2: Diagrama de bloques de la arquitectura interna tipo Harvard.

Peripheral Interface Controller (PIC)

Los controladores tipo PIC cuentan con una arquitectura interna de tipo Harvard, además de un sistema RISC[6], que permite correr cada instrucción en un ciclo de reloj, mejorando así la respuesta

[6] Acrónimo de *Reduced Instruction Set Computer*.

del microcontrolador, en comparación con sus antecesores. Actualmente es muy común el uso de los microcontroladores llamados PIC.

Comercialmente se pueden encontrar diferentes y muy variados modelos de microcontroladores PIC. Pueden variar en el número de entradas, el número de salidas y en si las entradas tienen ADC o no, en la frecuencia de oscilación que soportan, el rango de alimentación y el consumo de corriente, entre otras. Todo esto debe ser considerado al planear una aplicación determinada, escogiendo el modelo que más se adecue a las necesidades del proyecto.

En general, es necesario contar con uno o varios *software's* para poder programar al PIC. Normalmente se tiene uno que funciona como editor, en el cual el programador escribe la secuencia de instrucciones que desea que el PIC ejecute. Posteriormente se necesita otro programa que convierta la serie de instrucciones en lenguaje ensamblador y por último, otro más que permita grabar el programa en la memoria interna del PIC mediante un circuito programador.

Existen varias opciones de software que incluyen estas tres fases y además están diseñados para que la programación sea lo más simple posible, utilizando lenguajes de programación como C o *Basic*. El sistema PICAXE es uno de éstos, ya que permite de una forma muy simple, realizar programas para PIC, así como grabar dichos programas sin necesidad de un circuito programador extra.

PICAXE

PICAXE es un sistema extremadamente poderoso y de bajo costo. Este sistema de programación de microcontroladores está diseñado para simplificar, a estudiantes, la utilización de microcontroladores.

Esta simplificación se logra por medio de un pequeño software grabado en la memoria EEPROM de los PIC de propósito general. La empresa *Revolution Education*, creó este *Firmware* y se lo ha grabado a ciertos modelos de PIC de la empresa *Microchip*. El hecho es que al utilizar el sistema PICAXE, se programan microcontroladores PIC de alta calidad, de manera muy sencilla.

Para el diseño del programa que se le grabará al microcontrolador, PICAXE cuenta con un *software* gratuito con el cual se pueden realizar diseños de programas en diagrama de flujo o bien con comandos de programación *Basic*. Esto facilita mucho la programación de los microcontroladores ya que no es necesario hacerlo en lenguaje ensamblador. Cabe aclarar que el hecho de no programar un microcontrolador en lenguaje ensamblador puede dar como resultado un programa que no necesariamente es el más eficiente para resolver nuestra problemática. Es

muy recomendable, para aplicaciones donde se requiera extremo control de variables, que el lenguaje de programación que se utilice sea el de más bajo nivel para que el programa esté hecho justo a la medida de nuestras necesidades. Para aplicaciones generales, el sistema PICAXE ofrece excelentes resultados.

El software de PICAXE, permite hacer simulaciones del programa antes de grabárselo al microcontrolador. De esta forma ahorramos vida de nuestro microcontrolador, debido a que el fabricante nos garantiza un cierto número de veces en las que el programa puede ser regrabado.

Otra ventaja que ofrece el sistema PICAXE es que no requiere de un circuito programador para grabar el microcontrolador; esto se hace por medio de un cable de comunicación serie encontrándose el microcontrolador en el circuito de nuestra aplicación.

Los modelos con los que cuenta PICAXE se muestran en la siguiente tabla **[Fig. 2.3]**:

Modelo PICAXE	Modelo Microchip	Pines	Líneas de programación	Salidas	Entradas	ADC *8bits **8/10bits
PICAXE-08	PIC12F629-I/P	8	40	1-4	1-4	1*
PICAXE-08M	PIC12F683-I/P	8	80	1-4	1-4	3**
PICAXE-18	PIC16F627	18	40	8	5	3*
PICAXE-18A	PIC16F819-I/P	18	80	8	5	3**
PICAXE-18X	PIC16F88-I/P	18	600	9	5	3**
PICAXE-28A	PIC16F872-I/SP	28	80	8	8	4**
PICAXE-28X	PIC16F873A-I/SP	28	600	9-17	0-12	0-4**
PICAXE-40X	PIC16F874A-I/P	40	600	9-17	8-20	3-7**

Fig. 2.3: Tabla donde se muestran las diferentes características que tienen cada uno de los diferentes modelos de PICAXE.

Desarrollo

Capítulo III.- Diseño

Análisis del Problema

Se necesita diseñar y construir un dispositivo que permita cuantificar la conducta evocada por luz en la mosca Drosophila.

Este dispositivo debe permitir el control de la intensidad y de la longitud de onda de la fuente de luz, que podrá variar entre los colores blanco, azul, verde, ámbar y rojo. A su vez debe permitir la activación simultánea de dos fuentes de luz con diferentes intensidades y longitudes de onda, cada una en un extremo del dispositivo, generando una zona uniformemente iluminada para ser vista por la mosca.

El dispositivo debe permitir estudiar los mecanismos de selección, para lo cual se propone que tenga dos fuentes de luz en extremos opuestos de un tubo en el cual la mosca se desplazará. La mosca deberá ingresar a la cámara por un orificio central.

Debido a que la variable principal del dispositivo es la luz, no es posible observar la conducta de la mosca a simple vista, ya que para ello la cámara tendría que ser transparente, permitiendo así la entrada de luz natural y por ende alterando el experimento. Por esta razón el dispositivo debe contar con sensores para cuantificar posición, dirección, latencia y velocidad de la mosca durante los experimentos. Estos sensores deberán operar a una longitud de onda fuera del espectro visual de las moscas.

El control de las variables y el registro conductual se harán desde una computadora y mediante una interfase gráfica. Así el usuario podrá definir la longitud de onda e intensidad de cada fuente de luz así como los tiempos de oscuridad y de iluminación en cada prueba.

Alternativas de Solución

Para variar la longitud de onda de la luz se pensó originalmente en dos alternativas: la primera consistía en tener una fuente de luz blanca y un sistema mecánico de intercambio de filtros para cada longitud de onda. La segunda consistía en fuentes fijas de diferentes longitudes de onda activables individualmente, dependiendo de la longitud de onda requerida.

La diferencia primordial entre estas dos formas de resolver el problema es que, la primera, necesita un sistema electromecánico; mientras que la segunda sólo necesita un sistema electrónico. Independientemente de cual de las soluciones se implemente, es necesario contar con dos "módulos de control luminoso", uno para cada extremo de la cámara. Debido a esta duplicidad es que se decidió implementar un control independiente de intensidades para fuentes de diferente longitud de onda. Ésta al no contar con elementos mecánicos, es una solución más sencilla de construir y de mantener. Cabe aclarar que en términos de costos; la ausencia de elementos mecánicos también aporta un beneficio.

Fuentes de Iluminación

Hay dos alternativas, ya que existen fuentes de luz de diferente longitud de onda, cuyo principio de funcionamiento es distinto, básicamente existen las incandescentes (bombillas) y las fabricadas con semiconductores (LED's). La principal razón de optar por los LED's es su emisión fría, con lo que se evita la posible respuesta conductual al calor.

En estudios previos de *Drosophila* se sabe que las células que conforman sus ojos responden a longitudes de onda entre los 350 y los 650 nm; es por esto que escogimos trabajar con LED's que emiten dentro de este espectro óptico. A estos LED's se les realizó un proceso de caracterización, mostrado en el apartado: "Fuentes de luz fría" de este capítulo.

Control de Intensidades

Con respecto al diseño del controlador de intensidad de los LED's se contemplaron dos opciones: utilizar un convertidor digital analógico (DAC, por sus siglas en inglés) para alimentar a los LED's con voltajes variables, mediante un código binario; o bien utilizar un potenciómetro digital, cuyo funcionamiento permite aumentar gradualmente el valor de una resistencia al ser "excitado" con un pulso en una de sus entradas; de esta forma la terminal del selector del potenciómetro tendrá un valor de voltaje variable con el cual alimentar el LED.

El DAC presenta una ventaja respecto al potenciómetro digital, ésta consiste en que permite variar los valores de voltaje de forma salteada, mientras que en el potenciómetro digital requiere pasar por valores intermedios para alcanzar el valor de resistencia requerida. Esta desventaja del potenciómetro es insignificante al momento de cambiar el valor del potenciómetro digital en un tiempo muy reducido, ya que el microcontrolador, nos permite introducirle pulsos de 0.5 ms al potenciómetro digital. Teniendo en consideración que el potenciómetro tiene 100 valores diferentes, el tiempo máximo que le

llevará ajustar un valor determinado es de 50 ms; siendo este tiempo compatible con el tiempo de respuesta del ojo de la mosca.

En cambio el DAC tiene la desventaja de requerir 8 entradas para controlarlo y teniendo en cuenta que necesitaríamos tener dos DAC, uno para cada módulo de luces; resulta necesario asignar 16 de las salidas del microcontrolador sólo para el control de intensidad. Esto en comparación con las 4 salidas que se necesitan para controlar a los dos potenciómetros digitales, resulta poco eficiente.

La decisión de utilizar el potenciómetro digital supone contar con un sensor que le indique al microcontrolador que se ha llegado al valor de intensidad que se requiere y de esta forma deje de enviar pulsitos al potenciómetro digital. Esto se logra realimentando la salida de tensión del potenciómetro a un ADC del microcontrolador.

Finalmente, es posible desarrollar la interfase con el usuario en diferentes plataformas de programación. Se decidió utilizar Matlab ya que en el "Centro de Experimenta[1]" se usa en otras aplicaciones y los alumnos que asisten al Centro están familiarizados con ellas.

Fuentes de Luz Fría (LED's)

En este apartado se describe el proceso de caracterización al que fueron sometidos los diferentes diodos emisores de luz (LED's, por sus siglas en inglés) y se muestran los resultados obtenidos de cada etapa de la caracterización a partir de los cuales se determinó el tipo de LED's a utilizar en el dispositivo.

LED

El LED es un diodo que es capaz de emitir luz al ser polarizado directamente. Es un dispositivo de unión *p-n,* que debido a la polarización y al "dopaje" de los semiconductores emite luz. Puede existir cierta absorción de los paquetes de energía fotónica en la misma estructura, sin embargo un alto porcentaje de fotones escapa produciendo la emisión de luz. En todas las uniones de semiconductores *p-n,* cierta cantidad de esta energía se desprende en forma de luz, como el caso del fosfuro de arseniuro de galio (GaAsP) o el fosfuro de galio (GaP) en los que el número de fotones de energía luminosa emitida es suficiente como para crear una fuente de luz intensa **[6]**. Los fabricantes utilizan elementos como Galio, Arsénico y Fósforo para poder radiar luz de diferentes longitudes de onda **[7]**.

[1] Centro para la Enseñanza Experimental de la Ciencia en el Bachillerato.

Al requerir el dispositivo una fuente de luz que permita variar entre las longitudes de onda correspondientes al blanco, azul, verde, ámbar y rojo, se realizó la caracterización de los LED's comerciales. Existen dos tipos de éstos: LED's "normales" y "brillantes". Los LED's blanco y azul existen únicamente como brillantes, mientras que el verde, rojo y ámbar se fabrican en ambos tipos.

Proceso de Caracterización

La caracterización de cada LED abarcó tres variables:

- El espectro óptico
- La divergencia angular
- La relación intensidad-corriente

Espectro Óptico

Se determinó qué longitudes de onda forman el espectro óptico de los LED's rojo, ámbar, verde, azul y blanco.

Para obtener el espectro óptico se utilizó un espectrómetro **Ocean Optics HR4000 High-Resolution Fiber Optic Spectrometer [Fig. 3.1a]**, que se comunica con la computadora mediante el puerto USB o bien por el puerto serie. El software con el cual opera es el **OOIBase32,** permite visualizar los resultados en el monitor de la computadora. Las pruebas, para los diferentes LED's, se realizaron armando un arreglo compuesto de una computadora, el espectrómetro y una fibra óptica, de tal forma que el espectrómetro recibe la intensidad luminosa del LED mediante la fibra óptica, como se muestra **[Fig. 3.1b]**.

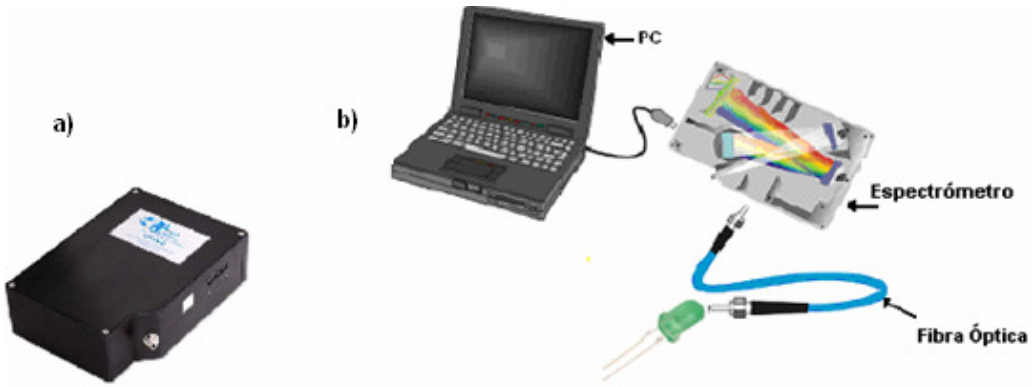


Fig. 3.1: a) Espectrómetro (Ocean Optics HR4000 High-Resolution Fiber Optic Spectrometer) utilizado para la obtención de los espectros ópticos correspondientes a los LED's en el proceso de caracterización. **b)** Arreglo utilizado en la medición de los espectros ópticos mediante el cual la luz emitida por cada LED pasa a través de la fibra óptica para entrar al espectrómetro, el cual envía los datos a la computadora mediante el puerto USB, permitiendo desplegar el espectro en la pantalla.

Cabe hacer la aclaración de que la fibra óptica que utilizamos tiene una respuesta espectral muy cercana a una plana, aunque no exactamente. Por lo que las mediciones que obtuvimos en cuanto a la longitud de onda son correctas, simplemente que para unos colores obtuvimos curvas de menor amplitud que otras; es por esto que se normalizaron para presentarse en este trabajo.

A continuación se muestran los espectros de emisión, de los LED's normales **[Fig. 3.2a]** y los LED's brillantes **[Fig. 3.2b]**.

a) **b)**

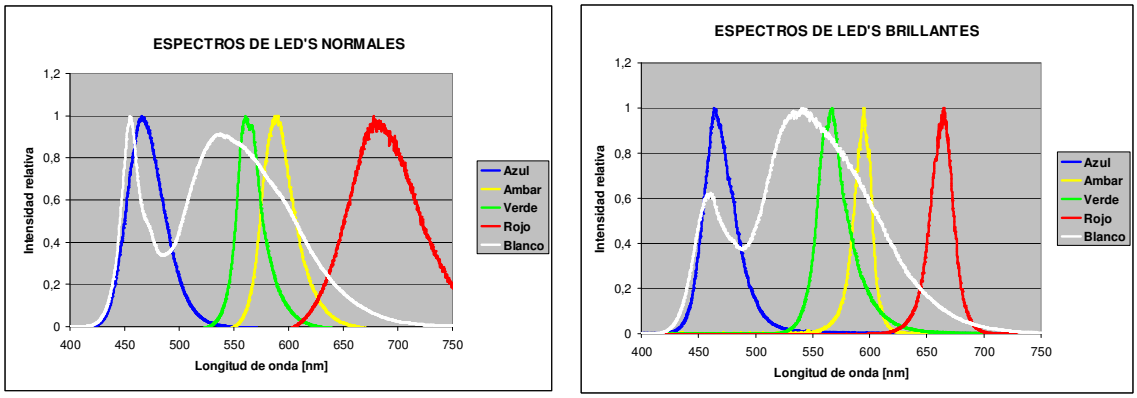


Fig. 3.2: Espectros normalizados para mostrar los intervalos de longitudes de onda para cada LED. Los espectros de LED's normales (**a**)) y brillantes (**b**)) muestran los picos de emisión dentro del rango del espectro visible de la mosca 350 nm a 650 nm. Nótese la presencia de 2 picos en la luz blanca.

Los LED's brillantes muestran espectros mejor definidos, con intervalos de longitudes de onda más estrechos.

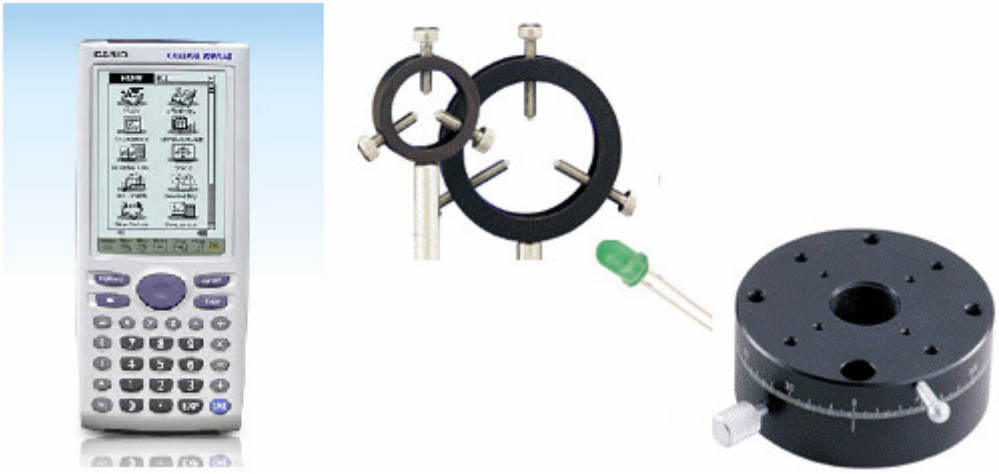
El LED rojo brillante tuvo una banda más estrecha, alrededor del pico (630 nm – 650 nm). Éste también se encuentra dentro del rango teórico para la longitud de onda correspondiente a dicho color y dentro del límite de longitudes de onda al que responde la mosca, 650 nm. En contraste, el LED rojo normal tiene un espectro notablemente desplazado hacia el infrarrojo, acercándose más a los 700 nm.

El espectro de los LED's blancos no abarcó todas las longitudes de onda uniformemente, mostrando en ambos casos un pico en el espectro azul y otro más amplio abarcando incluso el espectro rojo.

Divergencia Angular

En esta etapa se pretende obtener información acerca de la divergencia del haz de los LED's al emitir cada uno en su longitud de onda respectiva y así conocer la zona donde se concentra la mayor cantidad de luz.

El arreglo que se utilizó para esta etapa consta de una platina rotatoria y un sensor de luz del analizador de datos **Casio Data Analyzer EA-200.** Éste permite medir la cantidad de flujo luminoso en lúmenes [Lm]. Dicho sensor se utilizó para medir la cantidad de flujo luminoso que emitía el LED conforme se giraba la platina donde estaba montado. **[Fig. 3.3]**. Para realizar las mediciones se colocó dentro de cada sopote, uno frente al otro, tanto el sensor como el LED, este último montado además sobre la platina rotatoria que permitía mover 90° la posición del LED en sentido de las manecillas del reloj y contrario a éste, mientras que el sensor permaneció fijo para detectar la cantidad de luz que incidía en su superficie.



a) **b)** **c)** **d)**

Fig. 3.3: Componentes del montaje utilizado para medir la divergencia angular de los LED's: **a)** Dispositivo analizador de datos Casio Data Analyzer EA-200 que cuenta con un sensor de luz, **b)** Soportes para posicionar al transductor y al LED, **c)** LED, **d)** Platina rotatoria graduada.

A continuación se muestran las gráficas obtenidas del cono de divergencia angular emitido por LED's, normales **[Fig. 3.4a]** y brillantes **[Fig. 3.4b]**.

a) b)

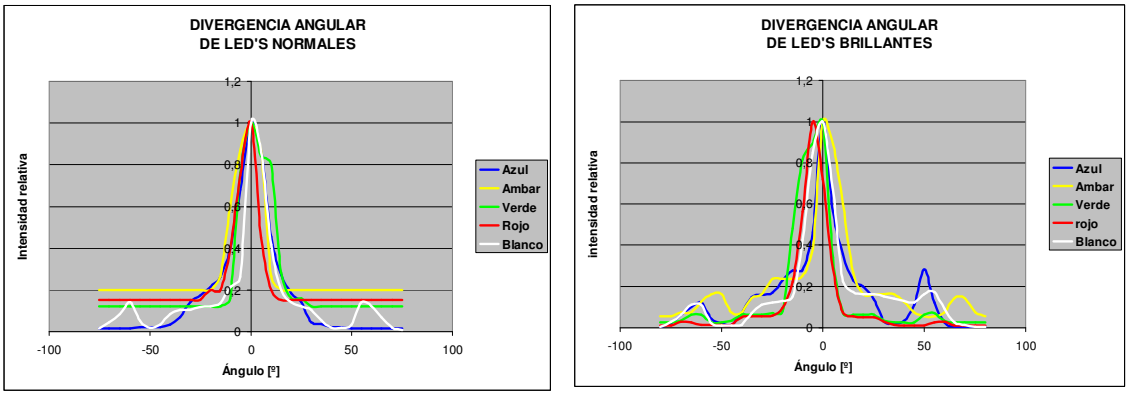


Fig. 3.4: Divergencia angular de: **a)** LED's normales con mayor concentración de luz en un cono de divergencia de alrededor de 40°. **b)** LED's brillantes con un cono de divergencia de alrededor de 40°.

Las gráficas de divergencia angular de ambos tipos de LED's, un cono de divergencia similar de alrededor de 40°, que es la zona donde se concentra la mayor cantidad de luz emitida.

Relación Intensidad-Corriente

Esta etapa tiene como objetivo mostrar la relación entre la cantidad de flujo luminoso y corriente de excitación en cada uno de los LED's.

El arreglo que se utilizó, requirió únicamente del sensor de luz con el que cuenta el dispositivo analizador de datos **Casio Data Analyzer EA-200, [Fig. 3.3],** el cual nos permite obtener mediciones de la cantidad de flujo luminoso en lúmenes [Lm]. Así al variar la cantidad de corriente inyectada, se produjo un cambio en la intensidad de luz emitida por cada LED.

Los resultados de esta etapa se muestran en las siguientes dos gráficas: la relación de Flujo luminoso vs. Corriente de LED's normales **[Fig. 3.5a]**.y la relación de Flujo luminoso vs. Corriente de LED's brillantes **[Fig. 3.5b]**. Ambas gráficas muestran el comportamiento arrojado directamente por el sensor. Dicha relación que se obtiene con el sensor no es lineal, condición que se puede observar mejor en el caso de los LED's blanco y azul. Esta aparente no linealidad en gran medida es debida a la resolución del dispositivo **Casio Data Analyzer EA-200,** que es 0.333 Lm, lo que hace que a pesar de notarse a simple vista un ligero incremento al aumentar la corriente inyectada, en ocasiones el sensor no alcanza a detectar un cambio más pequeño.

a) b)

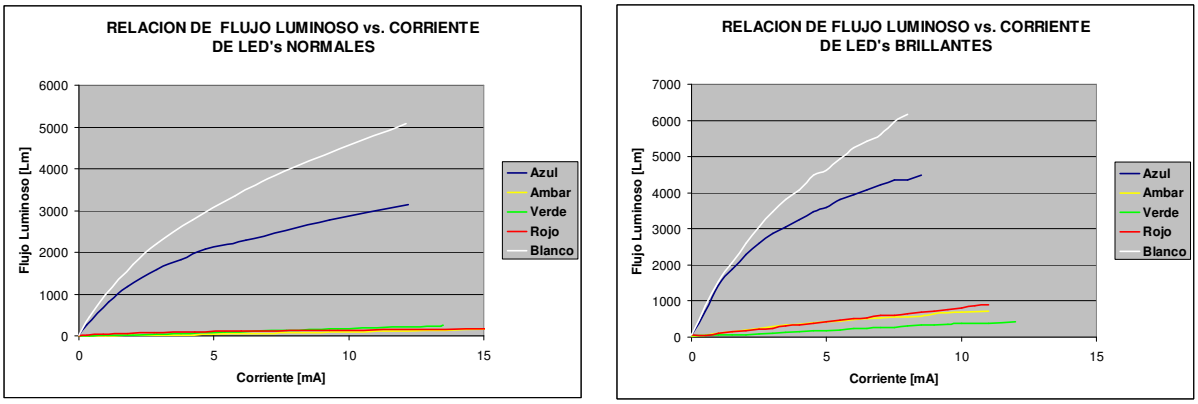


Fig. 3.5: a) Relación de Flujo luminoso vs. Corriente de LED's normales **b)** Relación de Flujo luminoso vs. Corriente de los LED's brillantes. En ambos casos con los valores obtenidos del sensor de flujo luminoso del dispositivo Casio Data Analyzer EA-200. Los valores máximos de flujo luminoso están entre los 171 a los 6162 Lm. Nótese la diferencia de pendientes para los LED's según su longitud de onda.

Debido a la razón expuesta anteriormente, y con base en las gráficas anteriores, se obtuvieron los modelos matemáticos para cada relación con la finalidad de mostrar de una forma aproximada como es el incremento del flujo luminoso al incrementar la corriente inyectada a cada LED de forma lineal.

A continuación se presentan las gráficas obtenidas a partir de los modelos matemáticos: la relación de Flujo luminoso vs. Corriente de LED's normales **[Fig. 3.6a]**.y la relación de Flujo luminoso vs. Corriente de LED's brillantes **[Fig. 3.6b]**.

a) b)

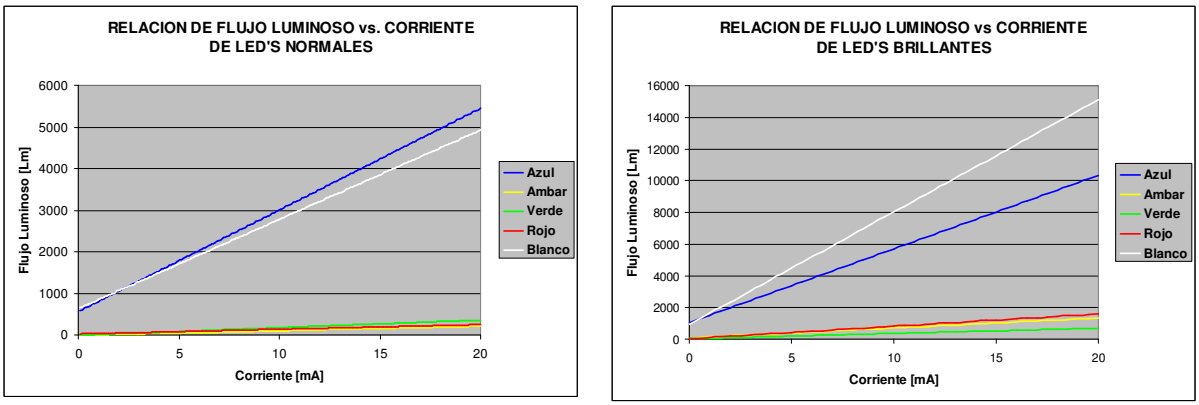


Fig. 3.6: a) La relación de Flujo luminoso vs. Corriente de LED's normales muestra una relación lineal, es decir al aumentar la corriente inyectada aumenta el flujo luminoso emitido por el LED además de que permite observar que el rango de flujo luminoso emitido por las diferentes longitudes de onda se encuentran entre los 200 a 5500 Lm para los valores máximos. **b)** La gráfica que muestra la relación de Flujo luminoso vs. Corriente de los LED's brillantes presenta una relación lineal entre la corriente inyectada y el flujo luminoso emitido. Los valores máximos de flujo luminoso se encuentran entre los 690 a los 15200 Lm. Nótese la diferencia de pendientes para los LED's según su longitud de onda.

La relación entre la corriente y el flujo luminoso que se presentan a partir de los modelos matemáticos en ambos casos se percibe que existe una relación lineal.

Finalmente el flujo luminoso emitido por cada LED brillante es mucho mayor que la de los normales.

Con base en las características de los LED's se tomó la decisión de usar brillantes en el dispositivo.

Diseño de la Cámara de Registro

Se fabricó una cámara en forma tubular en la que la entrada de la mosca, es por el centro, de modo que tenga dos opciones para moverse: a la izquierda o a la derecha de la cámara.

Cada extremo de la cámara tiene una fuente de iluminación consistente de 5 LED's, de los diferentes colores antes mencionados. Para lograr una iluminación lo más homogénea posible, se distribuyeron las 5 fuentes de luz en una superficie semiesférica que hizo incidir la luz de cada LED sobre una superficie de 5 mm. Para ello fue necesario diseñar el arreglo óptico mostrado en la figura **[Fig. 3.6]**.

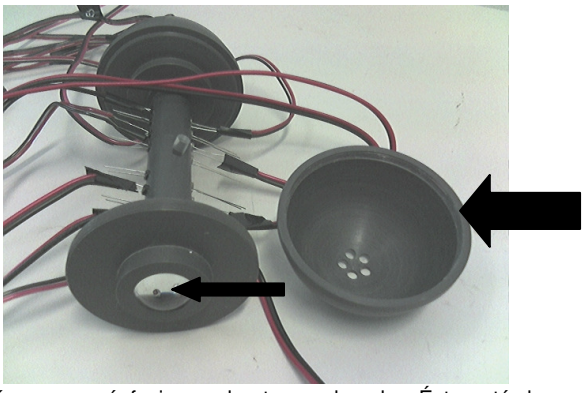


Fig. 3.6: La imagen muestra la cámara con énfasis en el extremo derecho. Éste está desarmado para mostrar la superficie semiesférica (flecha grande) donde se colocan los LED's y la lente que hace converger los diferentes haces de luz en la superficie de 5 mm (flecha delgada).

La parte cilíndrica de la cámara tiene un diámetro interno de 5mm basado en estudios previos de *Drosophila*, que muestran que éste es el espacio mínimo que le permite moverse girando con libertad. Por lo mismo, la superficie de luminosidad en cada extremo del tubo tiene el mismo diámetro.

A lo largo del tubo hay una perforación central para la entrada y salida de las moscas. El tubo tiene 8 pares de orificios destinados a la colocación de 4 pares de optoacopladores, compuestos por un emisor infrarrojo y su receptor. Las perforaciones adicionales permiten variar las posiciones de los optoacopladores en caso de requerirse. El emisor y el receptor de cada par se localizan diametralmente opuestos. La distancia entre cada par de perforaciones es de 1 cm. **[Fig. 3.7]**.

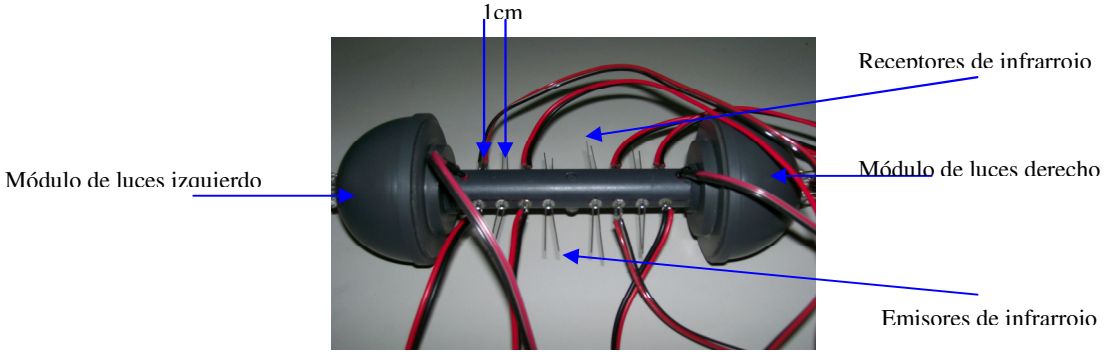


Fig. 3.7: La imagen muestra una vista en planta del dispositivo con los sensores y la perforación en el centro para la entrada y salida de la mosca.

El diseño del arreglo óptico fue realizado por la Dra. Martha Rosete Aguilar del Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET, UNAM) **Apéndice A**, mientras que la manufactura del dispositivo fue realizada por el Sr. Valentín López Cabañas en el taller mecánico del mismo Centro.

La Electrónica

El diseño y construcción de la parte electrónica del dispositivo requirió de varias etapas descritas a continuación.

Control de Intensidad Luminosa

Consiste de tres elementos principales: el microcontrolador, el potenciómetro digital y el LED. Se diseñó un control de lazo cerrado cuyo funcionamiento es bastante simple: el microcontrolador le envía pulsos al potenciómetro para que varíe su resistencia interna de tal forma que cambie el voltaje con el que se alimenta el LED, este cambio es medido por uno de los ADC's del microcontrolador, éste monitorea el valor de voltaje y lo compara con un valor esperado. Hasta que no se alcanza el valor deseado el microcontrolador hace variar el valor del potenciómetro digital el número de veces que sea necesario. A continuación se muestra el diagrama de esta etapa. Ver **[Fig. 3.8]**

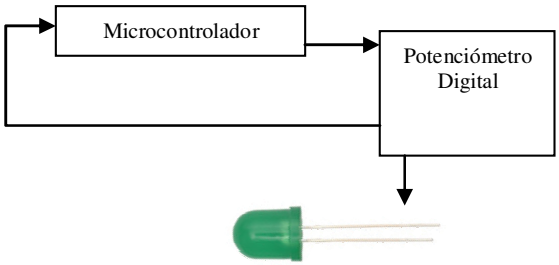


Fig. 3.8: La figura muestra el diagrama del control en lazo cerrado para la intensidad luminosa de un LED. Se observa que el microcontrolador gobierna el valor del potenciómetro digital de tal forma que la tensión a la salida del potenciómetro debe coincidir con el establecido por el usuario.

Selección de color.

Esta etapa tiene la finalidad de encender el LED de color que elija el usuario. Sus elementos son los siguientes: el microcontrolador, el potenciómetro digital, los transistores y los LED's de cada uno de los colores (azul, ámbar, verde, rojo y blanco). Los colectores de los 5 transistores están conectados al selector del potenciómetro digital, produciendo un circuito divisor de voltaje. Las bases de cada transistor están conectadas a una salida diferente del microcontrolador. Así el PIC controla qué salida polarizar, o sea que selecciona el transistor por excitar, para que sólo uno de ellos pase de su estado de corte a un estado de saturación en el cual el voltaje de colector se le aplica al LED respectivo. En la **[Fig. 3.9]** se observa el diagrama de esta etapa.

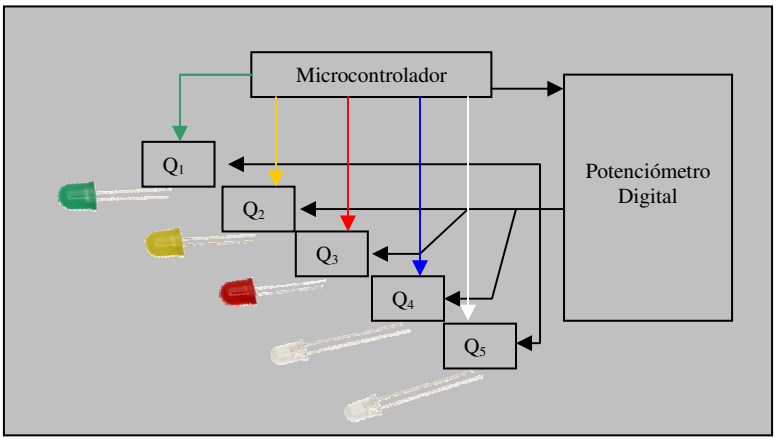


Fig. 3.9: Diagrama de la etapa de selección de color. Se observa como el microcontrolador "elige" qué transistor excitar para que haga encender el LED determinado por el usuario.

Monitoreo de la Posición de la Mosca

El objetivo de esta etapa es determinar automáticamente el movimiento de la mosca dentro de la cámara. Los principales componentes de este módulo son el microcontrolador y los optoacopladores de infrarrojo. Cuando la mosca pasa entre un emisor y su respectivo receptor, disminuye el voltaje en la resistencia conectada entre el emisor del fototransistor y tierra. El colector del fototransistor se encuentra conectado a 5 V. El voltaje en cada una de las resistencias de los cuatro optoacopladores es monitoreado cada 0.25 segundos por el microcontrolador por medio de cuatro diferentes ADC's, uno para cada sensor (cuando hablamos de sensor nos referimos a la pareja emisor-receptor de infrarrojo). A continuación se muestra el diagrama de esta etapa **[Fig. 3.10]**.

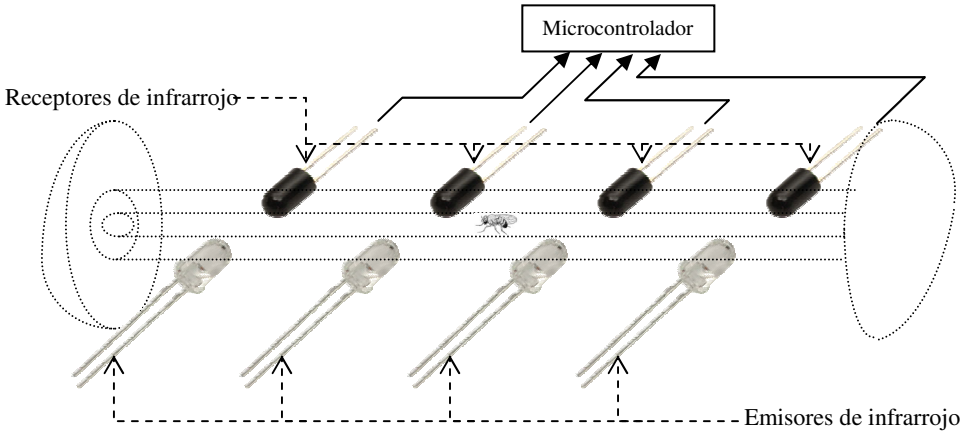


Fig. 3.10: El diagrama de la etapa de monitoreo de la mosca muestra cómo el paso de la mosca interrumpe el haz infrarrojo del emisor de tal forma que el receptor envía una disminución de voltaje al microcontrolador, de esta forma podemos conocer cómo es el movimiento de la mosca durante los experimentos.

Comunicación Serie

Esta etapa permite que el dispositivo reciba y envíe información a la computadora del usuario. La información que recibe son los parámetros que puede definir el usuario en cada experimento: el tiempo que permanecerá obscura la cámara antes y después de comenzar el experimento, como el color de la luz en cada extremo (color izquierdo y color derecho), la intensidad luminosa en cada extremo (intensidad relativa de 0 a 20) y la duración del ensayo. Esto incluye el tiempo de adaptación a la obscuridad, el tiempo de iluminación y el tiempo de readaptación a la obscuridad. A pesar de que los PICAXE se programan mediante comunicación serie con la PC; es necesario establecer una entrada y una salida del microcontrolador que serán exclusivamente utilizadas para la comunicación con la computadora. El protocolo que utilizamos consiste en palabras de 8 bits, sin bit de paridad, un *baudrate* de 4800 y un bit de paro.

Diagrama del Circuito Eléctrico Final

El diagrama final del circuito eléctrico **[Fig. 3.11]** muestra la totalidad de los componentes, a pesar de que no necesariamente están incluidos en el diagrama a bloques correspondiente. En la descripción de la etapa de control de intensidad **[Fig. 3.8]** se muestra la utilización de un solo potenciómetro digital, pero en el diagrama final se muestran dos ya que debe haber un control para cada uno de los módulos de LED's que se encuentran en los extremos de la cámara. Asimismo la descripción de la selección de color **[Fig. 3.9]** sólo muestra el diagrama a bloques para un extremo de la cámara; en el diagrama final se observa la duplicidad de esta etapa.

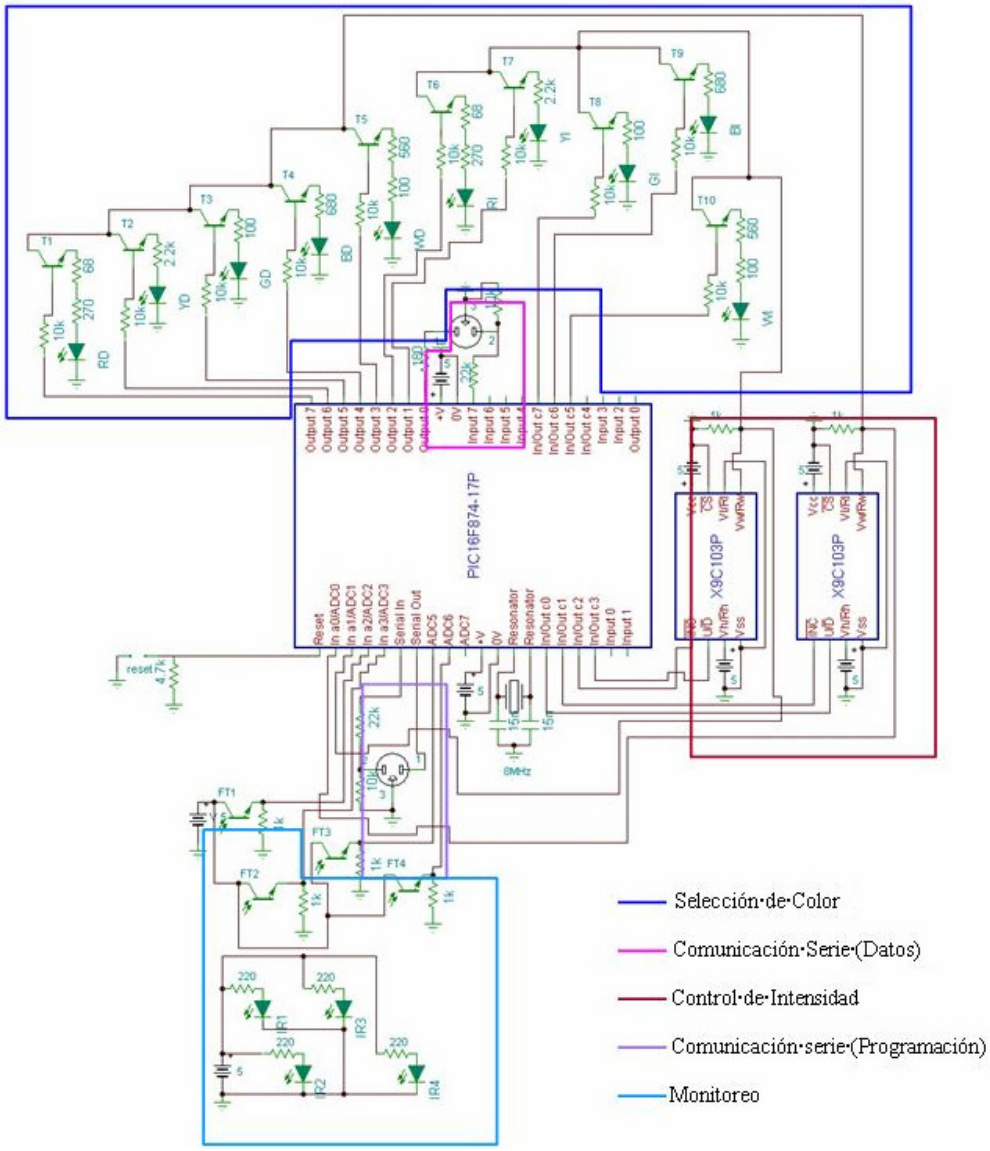


Fig. 3.11: Diagrama del circuito eléctrico con todas las conexiones y componentes electrónicos necesarios para el funcionamiento del dispositivo. Se encuentra dividido en bloques para que se aprecie cada una de las etapas descritas en este capítulo.

Con base en el diagrama del circuito eléctrico que se muestra a continuación se realizó el circuito impreso para la implementación de la electrónica. Ver **Apéndice C** y **Apéndice E.**

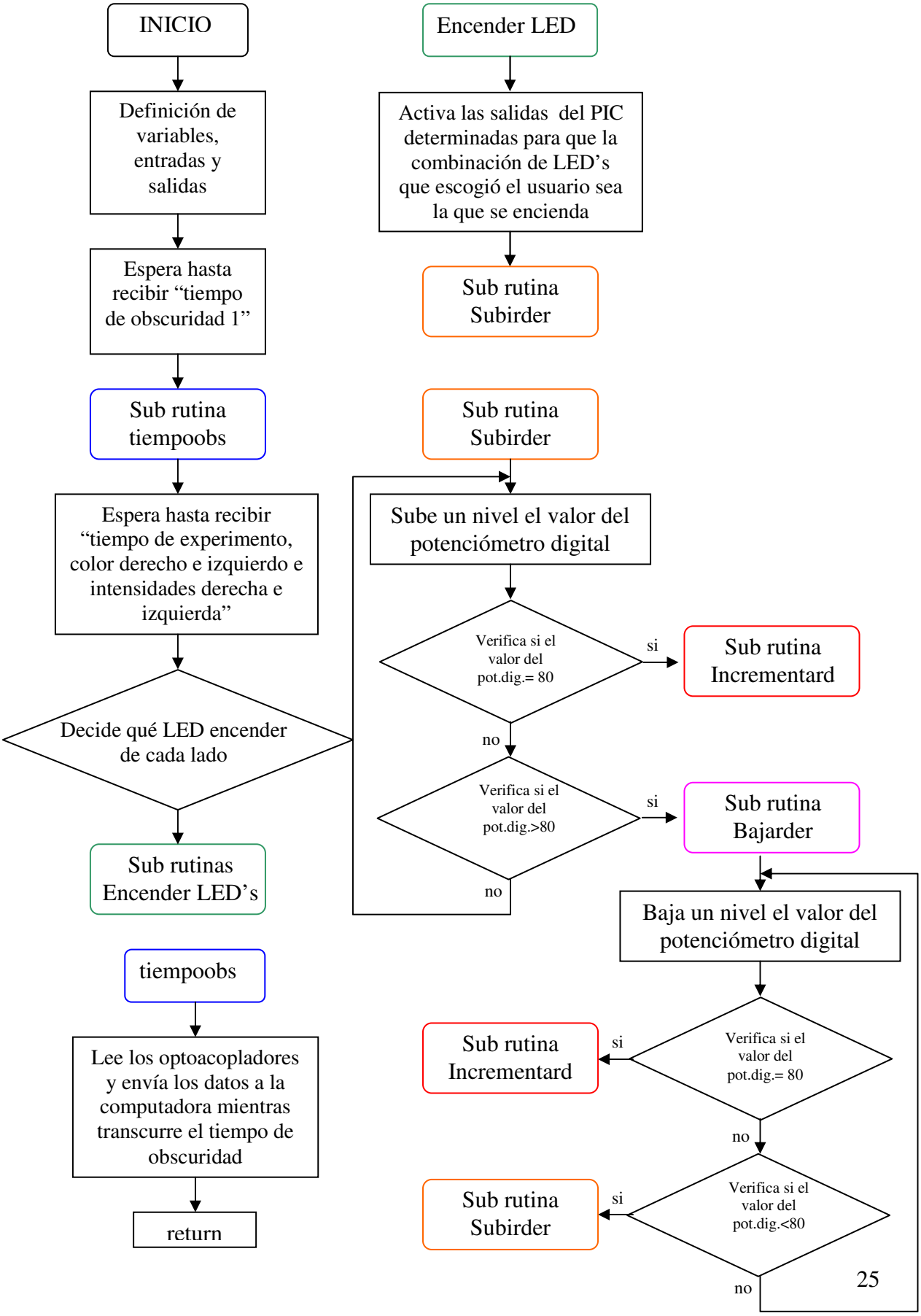
Programa del Microcontrolador

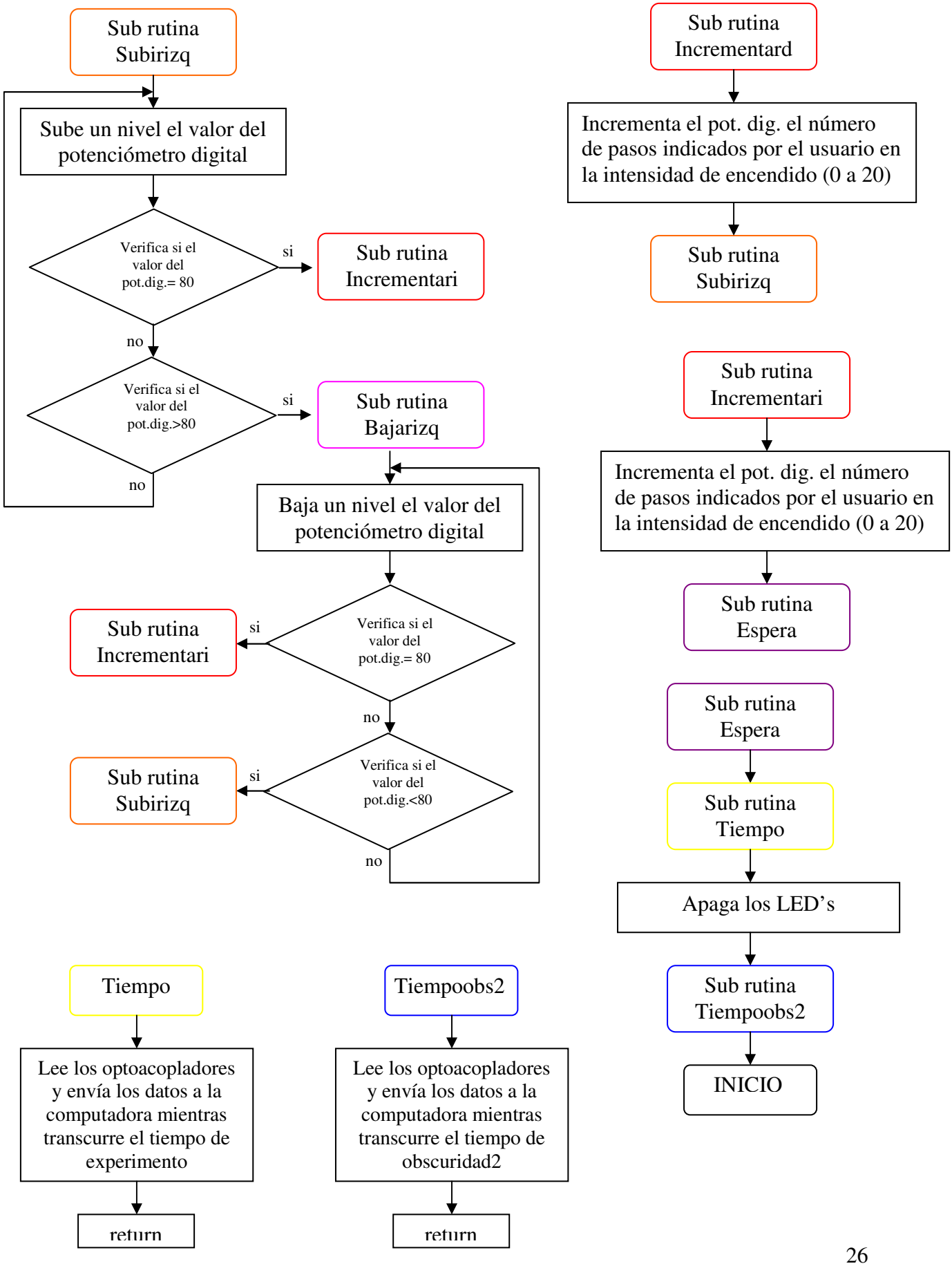
El programa recibe los datos enviados por el usuario, realiza la subrutina “tiempoobs1” (tiempo de obscuridad 1) en la cual no enciende ningun LED pero envía los datos de los optoacopladores a la computadora. A continuación enciende la combinación de luces definida por el usuario y realiza las subrutinas “subirder” (subir intensidad derecha) y “bajarder” (bajar intensidad derecha) de tal forma que el potenciómetro digital quede en un nivel cero de intensidad para después realizar la subrutina “incrementard” (incrementar la intensidad derecha) y así llegar al valor de intensidad deseado.

Posteriormente se hace lo mismo para el lado izquierdo de la cámara. Una vez que las intensidades están listas el programa realiza la subrutina “espera” dentro de la cual se realiza otra subrutina llamada “tiempo”; ésta se encarga de enviar los datos de los optoacopladores a la computadora mientras dura el tiempo de exposición a la luz. Habiendo terminado continúa la subrutina “espera” donde se apagan las luces y se direcciona a la subrutina “tiempoobs2”, la cual se encargará de enviar los datos del monitoreo a la computadora mientras dure el tiempo de readaptación a la obscuridad. Una vez terminada esta subrutina el programa vuelve a condiciones iniciales.

A continuación se describe mediante un diagrama de flujo la lógica del programa para el microcontrolador. El programa completo se encuentra en el **Apéndice D.**

Diagrama de Flujo del Programa del PIC




Sub rutina Subirizq
Sube un nivel el valor del potenciómetro digital
Verifica si el valor del pot.dig.= 80
si
Sub rutina Incrementari
no
Verifica si el valor del pot.dig.>80
si
Sub rutina Bajarizq
no
Baja un nivel el valor del potenciómetro digital
Sub rutina Incrementari
si
Verifica si el valor del pot.dig.= 80
no
Sub rutina Subirizq
si
Verifica si el valor del pot.dig.<80
no
Sub rutina Incrementard
Incrementa el pot. dig. el número de pasos indicados por el usuario en la intensidad de encendido (0 a 20)
Sub rutina Subirizq
Sub rutina Incrementari
Incrementa el pot. dig. el número de pasos indicados por el usuario en la intensidad de encendido (0 a 20)
Sub rutina Espera
Sub rutina Espera
Sub rutina Tiempo
Apaga los LED's
Sub rutina Tiempoobs2
INICIO
Tiempo
Lee los optoacopladores y envía los datos a la computadora mientras transcurre el tiempo de experimento
return
Tiempoobs2
Lee los optoacopladores y envía los datos a la computadora mientras transcurre el tiempo de obscuridad2
return

La Interfase con el Usuario (GUI)

La interfase gráfica permite al usuario comunicarse con el dispositivo para establecer los parámetros de cada prueba y a la vez muestre el comportamiento de la mosca dentro de la cámara. Ver **[Fig. 3.13].**

El ambiente de desarrollo fue Matlab, haciendo uso de MATLAB GUIDE (Graphical User Interface) el cual contiene un conjunto de herramientas gráficas como botones de comando y selección, menús desplegables, barras de control, campos para mostrar y editar texto, etc., que facilitan tanto la creación como la programación de cualquier interfase, permitiendo crear ambientes gráficos parecidos a las ventanas de Windows.

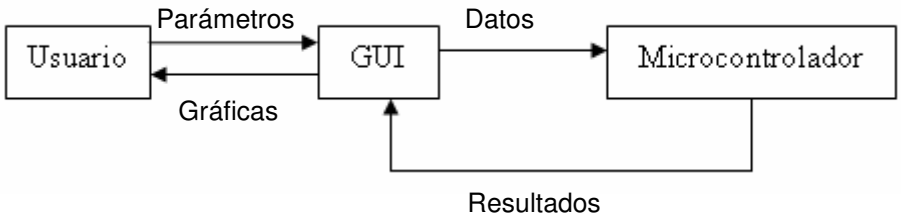


Fig. 3.12: El usuario introduce los parámetros requeridos mediante la interfase gráfica de usuario (GUI). La GUI envía los datos al microcontrolador para activar el ensayo, mientras que el mismo microcontrolador detecta los resultados de los sensores. Estos datos son transferidos a la GUI y mostrados gráficamente.

La interfase gráfica consta de dos ventanas: en la primera se introducen los parámetros y en la segunda, se muestran los resultados de forma gráfica.

En la ventana inicial el usuario tiene campos de texto (*edit text*) para introducir la duración del tiempo de obscuridad antes y después del ensayo; la duración del ensayo (tiempo en que se mantiene encendida la fuente de luz) y el valor de intensidad en cada lado. Hay también menús desplegables (*pop-up menus*) para elegir el color de la luz y un botón de comando (*push button*) para iniciar comenzar el ensayo. Ver **[Fig. 3.13]**.

Los campos de texto para la duración de la obscuridad y el ensayo son tres debido a que se define en horas, minutos y segundos, y están delimitados a un máximo de 4 hrs en cada caso, teniendo como duración máxima total 12 hrs entre los tiempos de obscuridad y de experimento.

Las intensidades luminosas están limitadas a valores de 0-20 en una escala de intensidad luminosa relativa. La relación entre esta escala y el valor real en lúmenes dentro de la cámara se

analiza en el apartado correspondiente a la “cuantificación de intensidad luminosa”, en este mismo capítulo.

Dentro del cuerpo del programa de la GUI, se hace el manejo de datos para enviar y recibir la información del microcontrolador mediante comunicación a través del puerto serie de la computadora. Ver **[Apéndice F]**. Una vez que se tienen todos los parámetros introducidos por el usuario, el programa hace una asignación de un caracter según el color elegido. El primer dato que se envía es el tiempo de obscuridad. Se espera a recibir los datos de los sensores de infrarrojo por el microcontrolador que son guardados en una matriz para graficarlos de forma ordenada. Al terminar de recibir dichos valores se envían los datos de las intensidades en cada lado, los caracteres de los colores elegidos y el tiempo de ensayo. A través del puerto serie se reciben nuevamente los datos de los sensores que son guardados en una segunda matriz y al término del ensayo se reciben y guardan en una nueva matriz los datos correspondientes al tiempo de obscuridad final. De tal forma que se tiene tres matrices, cada una de 4 columnas, correspondientes a cada sensor y “n” renglones (según el tiempo). De ellas se hacen las gráficas para mostrar la posición de la mosca en la cámara, a lo largo de la prueba.

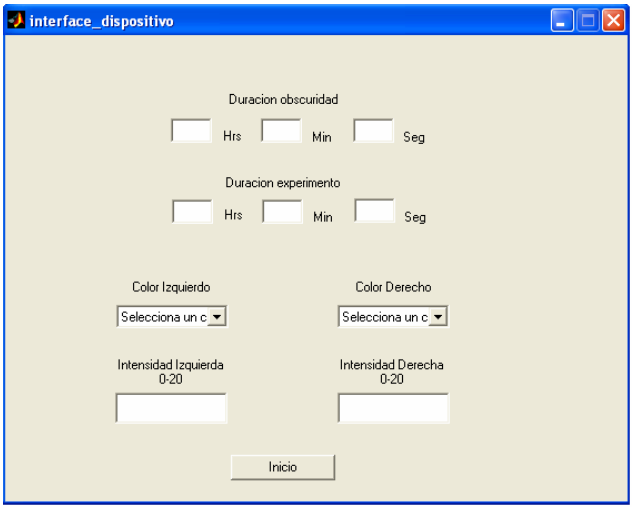


Fig. 3.13: Diseño de la ventana inicial de la interfase con el usuario, en la cual se introducen los parámetros para cada ensayo. Éstos son: duración de la obscuridad, color izquierdo y/o derecho, intensidad izquierda y derecha, duración del ensayo y finalmente un botón de inicio. Cabe aclarar que la duración de la oscuridad inicial y final son la misma por lo que sólo existe un campo en la interfase.

La segunda ventana de la interfase de usuario muestra el paso de la mosca a través de los optoacopladores a lo largo del tiempo. Está compuesta por tres gráficas. En la primera se muestra el comportamiento durante el periodo de adaptación a la obscuridad, la segunda muestra el comportamiento durante la iluminación y la tercera durante el periodo de readaptación a la obscuridad. Las gráficas en todos los casos muestran en el eje de las abscisas el tiempo en segundos y en el eje de las ordenas los sensores. En orden descendente se muestran las

posiciones 4 a 1 de la cámara de izquierda a derecha. En la gráfica correspondiente al tiempo de ensayo se indica el color y la intensidad para cada lado de la prueba en la parte superior e inferior. **[Fig. 3.14]**

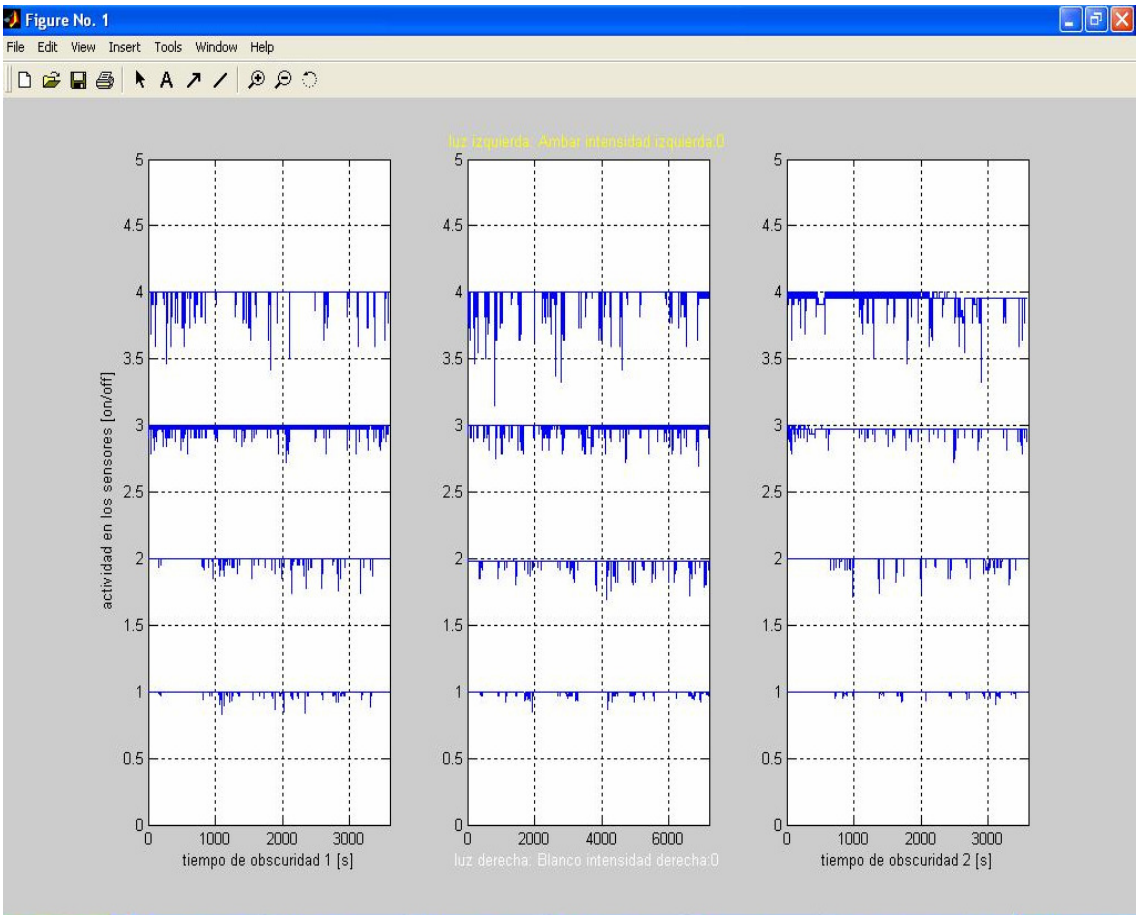


Fig. 3.14: Ventana de la interfase gráfica que muestra el comportamiento de la mosca durante el tiempo de obscuridad y el tiempo que se encuentran encendidas las fuentes de luz.

Cuantificación de la Intensidad Luminosa

Se cuantificó la intensidad luminosa de cada uno de los LED's, para establecer una equivalencia entre la intensidad relativa (de 0 a 20) que especifica el usuario y la intensidad en lúmenes.

Para ello se utilizó el analizador de datos **Casio Data Analyzer EA-200**; el cual cuenta con una fotorresistencia como sensor de intensidad luminosa, cuya respuesta es dependiente de la longitud de onda, **[Fig. 3.15].**

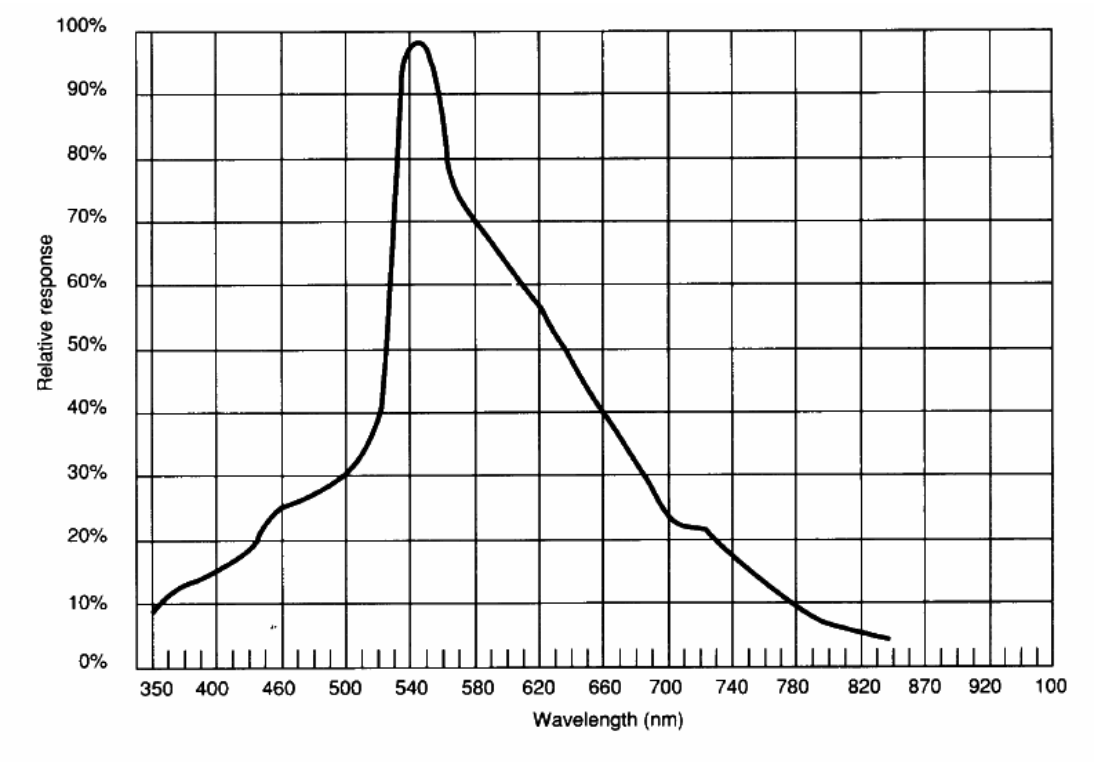


Fig. 3.15: La gráfica muestra la respuesta relativa de la fotorresistencia dependiente de la longitud de onda de la señal a cuantificar.

Las mediciones se realizaron colocando el sensor en las diferentes posiciones que permite el diseño de la cámara; es decir, a 1, 2, 3 y 4 centímetros del origen de la iluminación. Las siguientes tablas muestran las mediciones a 1cm.

Intensidad relativa	rojo	ámbar	verde	azul	blanco
1	0.66	0.66	0.66	0.66	0.66
2	0.66	0.66	0.66	0.66	0.66
3	1	0.66	0.66	0.66	0.66
4	1	0.66	0.66	0.66	0.66
5	1	0.66	0.66	0.66	0.66
6	1.33	0.66	0.66	0.66	0.66
7	1.33	0.66	0.66	0.66	0.66
8	2.33	1	0.66	0.66	0.66
9	2.33	1	0.66	0.66	0.66
10	2.66	1	1	0.66	0.66
11	4	1	1	1	1
12	4.66	1.33	1	2.33	3
13	6.33	1.33	1.33	3	8.33
14	7	2	1.33	6.33	13.66
15	8.66	2.33	2.33	8.66	22
16	11.33	2.66	2.66	13	31
17	15.66	3	3	16.33	41.33
18	21.33	4	4.66	22	53
19	24.33	4.33	6.66	30.33	70
20	28	4.66	9	34.33	85.6

Tabla 3.1: Caracterización para los LED's del extremo izquierdo, se muestra la intensidad relativa con su respectiva intensidad en lúmenes a 1 cm de la zona donde incide la luz homogéneamente.

Intensidad relativa	rojo	ámbar	verde	azul	blanco
1	0.66	0.66	0.66	0.66	0.66
2	0.66	0.66	0.66	0.66	0.66
3	0.66	0.66	0.66	0.66	0.66
4	1	1	0.66	0.66	0.66
5	1	1	0.66	0.66	0.66
6	1.33	1.33	0.66	0.66	0.66
7	1.33	1.33	0.66	0.66	0.66
8	2.33	2.33	0.66	0.66	0.66
9	2.66	2.66	1	0.66	0.66
10	3	3	1	0.66	0.66
11	4	4.33	1	1	0.66
12	4.66	5	1.33	1.33	1.33
13	5	6.33	1.33	2.66	5
14	6.66	7	2.33	4.66	11.33
15	8.33	8.66	2.66	6.66	19
16	10.66	11	3	8.66	30.66
17	13	13.33	4.66	11.33	44.66
18	16	16	6.66	13.66	57.33
19	22	19	10.66	18.66	75
20	30.33	22	16	22	96.33

Tabla 3.2: Caracterización para los LED's del extremo derecho, se muestra la intensidad relativa con su respectiva intensidad en lúmenes a 1 cm de la zona donde incide la luz homogéneamente.

Las mediciones para las demás posiciones de cada extremo de la cámara se pueden revisar en su totalidad en el **Apéndice G.**

Se puede observar de las lecturas que la resolución del analizador de datos utilizado, no es la adecuada para medir las variaciones de intensidad del dispositivo, es por esto que en sucesivas mediciones obtenemos valores repetidos de 1 Lm. Algo más a destacar es la desigualdad de los LED's de un lado respecto del otro. Esto se puede deber a la calidad de fabricación de los LED's pero puede ser corregido ajustando su posición en la cámara, de tal forma que se igualen la mayoría de las mediciones. Se sugiere que este proceso de cuantificación, que es más complicado se realice con un aparato que cuente con una mejor resolución.

A continuación se muestran las gráficas correspondientes a la caída de la intensidad luminosa a lo largo de la cámara para los LED's de color rojo. El detalle de los demás colores se aprecia en el **Apéndice G**.

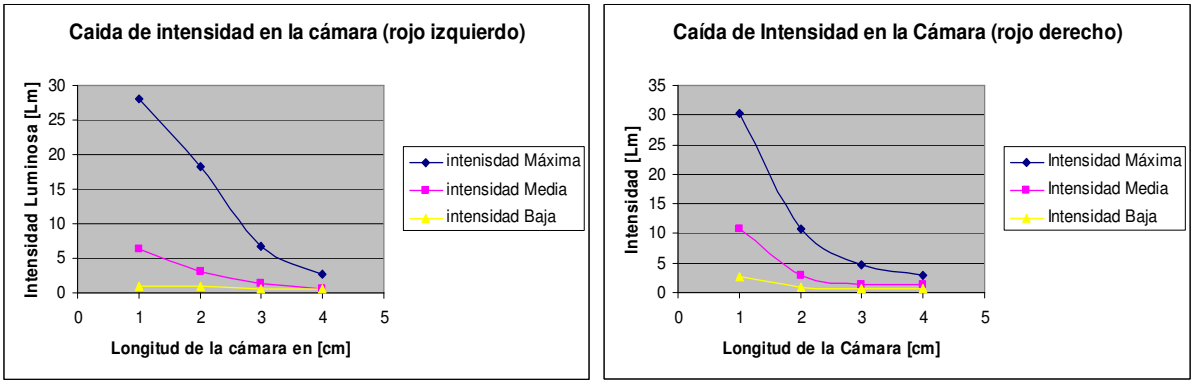


Fig. 3.16: La gráfica muestra cómo es la caída de la intensidad luminosa a lo largo de la cámara, en este caso para la luz roja.

Resultados

Capítulo IV.- Resultados Experimentales

En este capítulo se presentan los resultados de los ensayos realizadas del dispositivo. Los experimentos fueron divididos en dos partes: en obscuridad y en presencia de alguna de las fuentes de luz con las que cuenta el dispositivo.

En obscuridad se realizaron dos tipos de ensayos, para obtener una basal sin y con mosca **[Fig. 4.1]** y **[Fig. 4.2]** respectivamente.

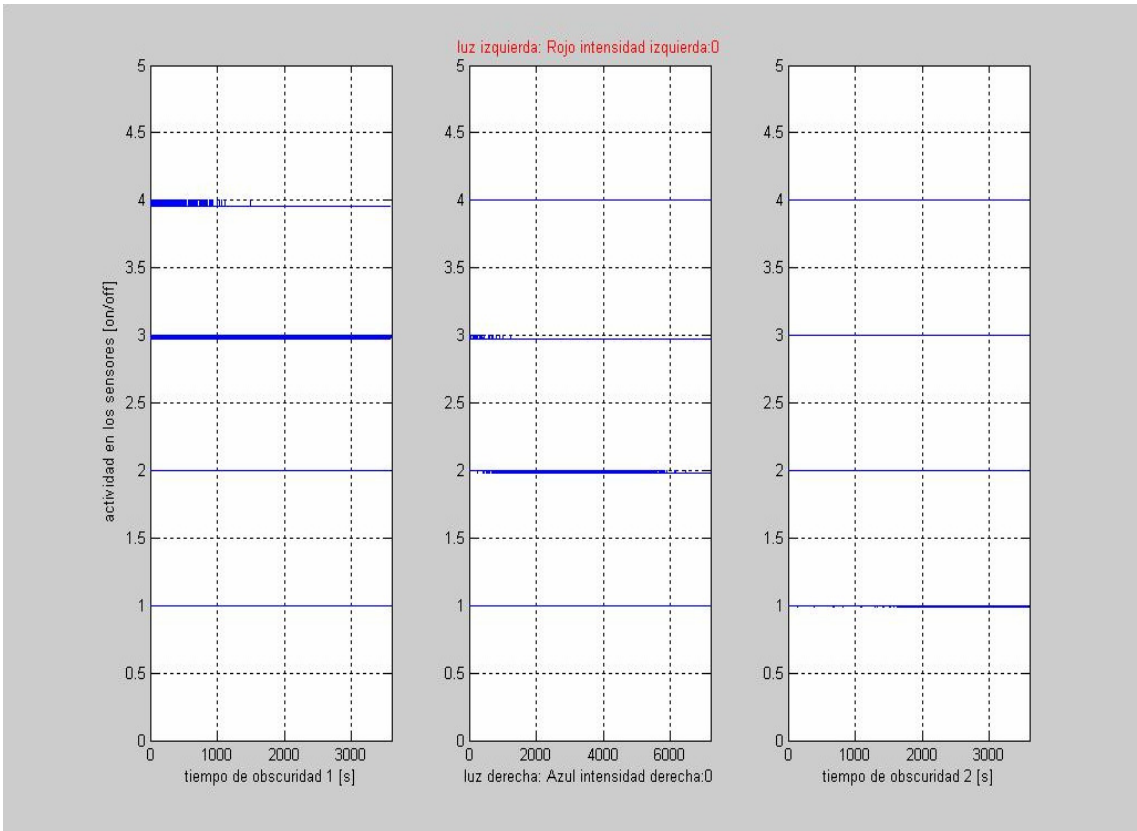


Fig. 4.1: Actividad basal de la cámara de registro en 4 hrs de ensayo (1hr de obscuridad inicial, 2 hrs de experimento y 1hr de obscuridad final) sin mosca en el interior. Las gráficas representan la actividad de los sensores a lo largo del tiempo determinado. Los números 1, 2, 3 y 4 en el eje *y* representan los sensores ordenados de derecha a izquierda respectivamente.

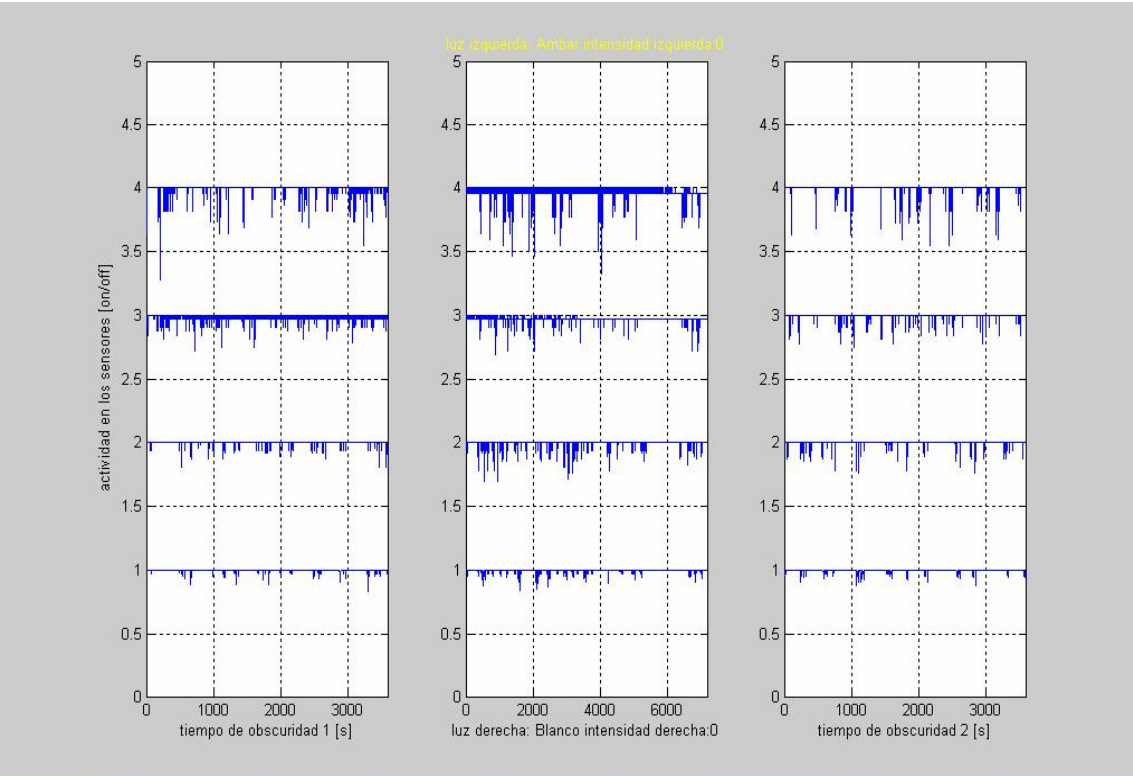


Fig. 4.2: Actividad de una mosca en completa obscuridad a lo largo de 4hrs de ensayo (1hr de obscuridad inicial, 2 hrs de experimento y 1hr de obscuridad final). Las gráficas representan la actividad de los sensores a lo largo del tiempo determinado, mostrando una espiga al momento de pasar la mosca por cada sensor. Los números 1, 2, 3 y 4 en el eje *y* representan los sensores ordenados de derecha a izquierda respectivamente.

Ambos ensayos tuvieron una duración de 4 horas. Estas gráficas están divididas en 1 hora de obscuridad inicial, 2 horas de exposición y 1 hora de obscuridad final y son representativas de una muestra de 2 repeticiones de cada una de ellas, ver **[Apéndice H].**

Para expandir la resolución temporal se realizaron experimentos similares con tiempos de 5 min, utilizando 5 moscas diferentes para obtener información estadística del movimiento de las moscas en ausencia de una fuente de luz. Estos experimentos se repitieron 5 veces por cada mosca. A continuación se muestra una gráfica representativa de cada mosca en obscuridad total durante el tiempo de ensayo **[Fig. 4.3]**. El comportamiento de cada mosca es diferente como se puede observar en cada gráfica. Las espigas registradas al momento de pasar por cada sensor permiten trazar manualmente la trayectoria en cada caso. Para discernir entre si una espiga pequeña corresponde al paso de la mosca o bien es solo una variación del dispositivo, es suficiente con seguir el movimiento de la mosca y observar el tamaño de las demás espigas para tomar una decisión **[Fig. 4.4]**.

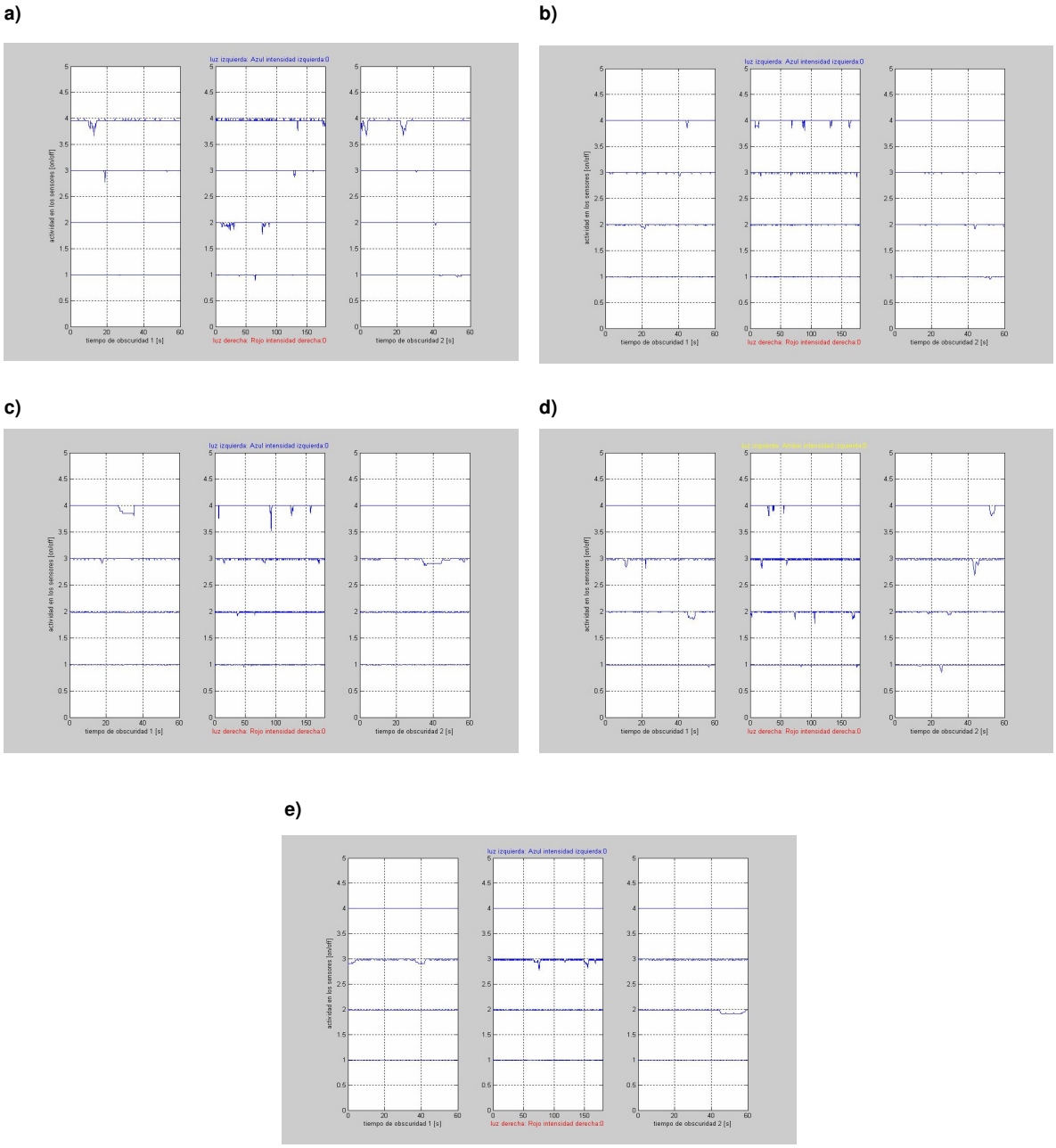


Fig. 4.3: Ensayos de 5 min en obscuridad total. **a)** Gráfica representativa de la mosca 1. **b)** Gráfica representativa de la mosca 2. **c)** Gráfica representativa de la mosca 3. **d)** Gráfica representativa de la mosca 4. **e)** Gráfica representativa de la mosca 5. Cada gráfica presenta un comportamiento distinto de la actividad de cada mosca dentro de la cámara de registro.

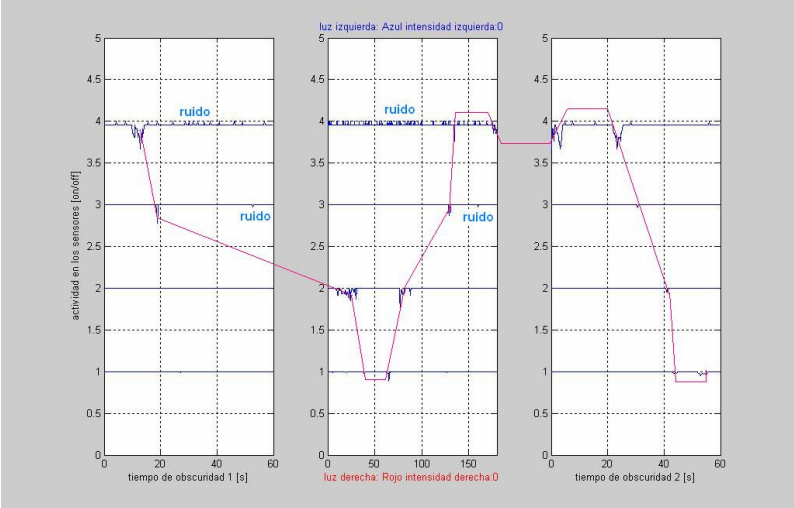


Fig. 4.4: Gráfica representativa de la mosca 1 de 5 min en obscuridad total. Se muestra la trayectoria de la mosca dentro de la cámara de registro y las zonas donde existen variaciones en el dispositivo.

Del primer análisis de estos ensayos se obtuvieron las frecuencias relativas correspondientes a la preferencia por zonas específicas de la cámara. Para ello se midió la frecuencia de paso por cada sensor en ambas direcciones, como se muestra de forma individual por mosca **[Fig. 4.5]**. Como se puede observar en los resultados del total de la muestra, los eventos que presentan mayor frecuencia es el paso de la mosca sobre el sensor izquierdo externo principalmente y derecho interno, mostrando también una tendencia muy baja sobre el sensor derecho externo **[Fig. 4.6]**.

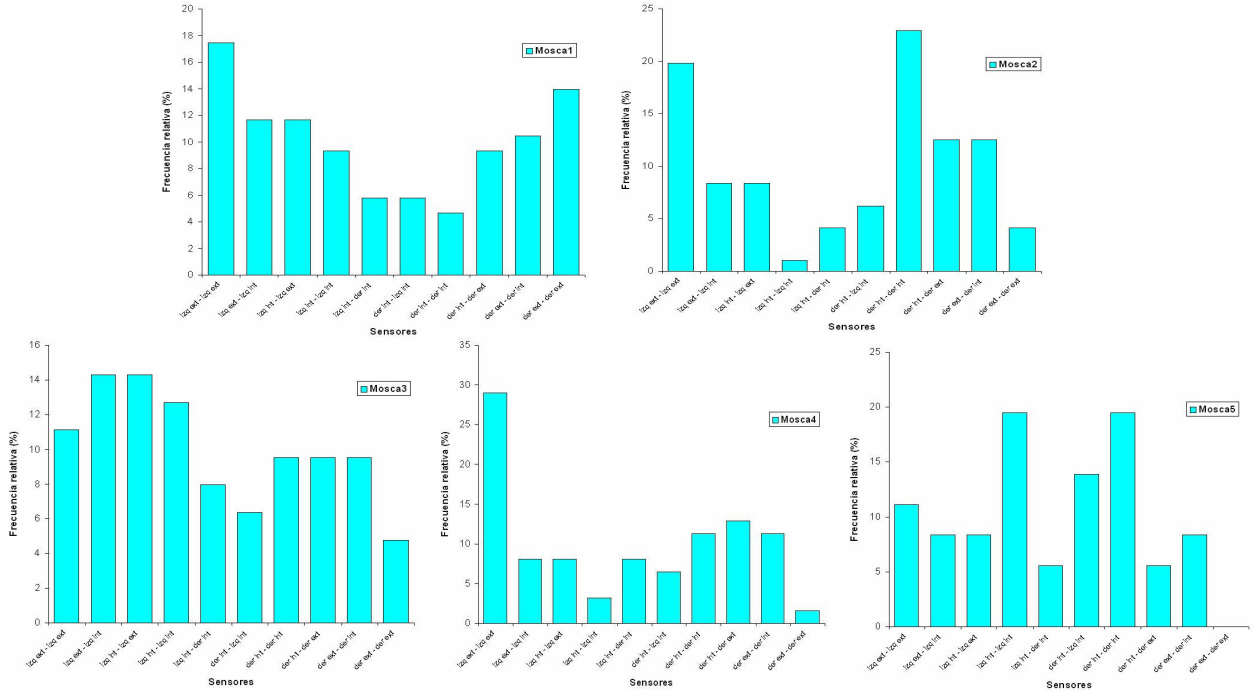


Fig. 4.5: Distribuciones representativas de las frecuencias relativas correspondientes a la preferencia de las moscas por zonas específicas de la cámara de registro.

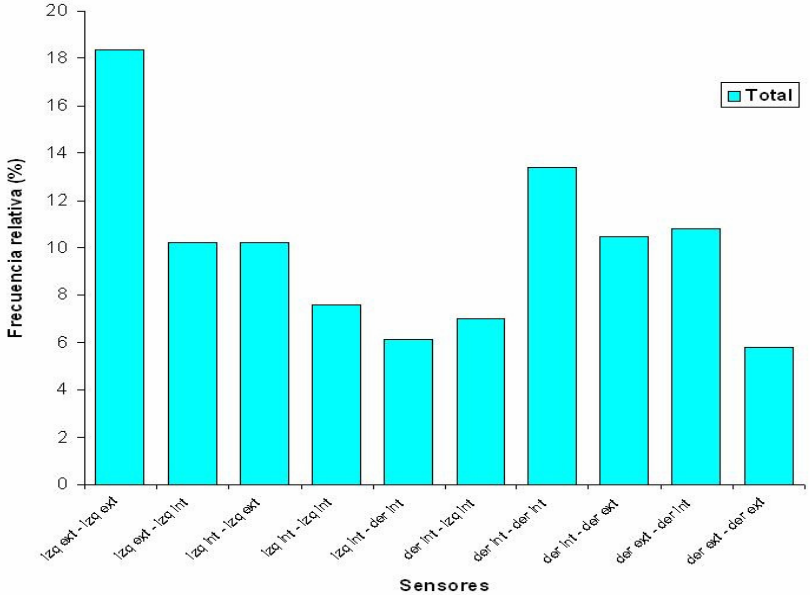


Fig. 4.6: Distribuciones representativas de las frecuencias relativas correspondientes a la preferencia de las moscas por zonas específicas de la cámara de registro. Se muestra una tendencia a permanecer del lado izquierdo sobre el sensor izquierdo externo.

La otra parte del análisis estadístico corresponde a las frecuencias relativas de la latencia del tiempo en que tardan los sujetos en pasar espontáneamente de un sensor a otro. Cabe mencionar que para este análisis se tuvo una muestra de 5 repeticiones para cada mosca. De igual forma se realizó el análisis estadístico para cada mosca **[Fig. 4.7]**.

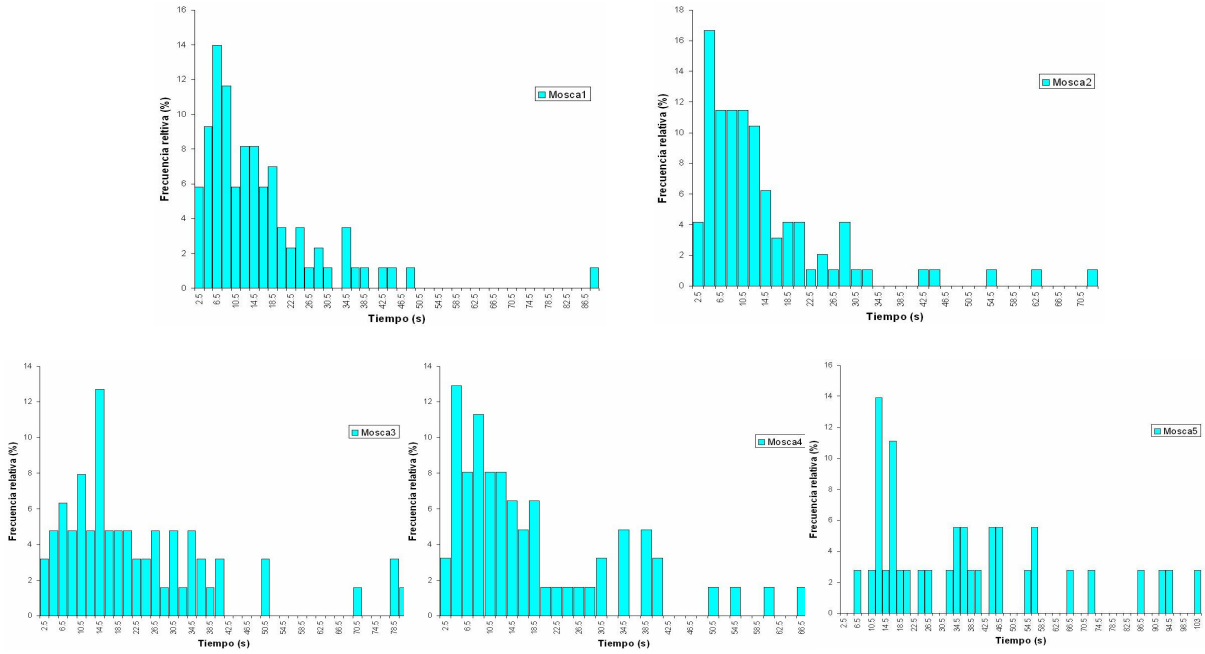


Fig. 4.7: Distribuciones de frecuencias de latencia de cada mosca para pasar de un sensor a otro dentro de la cámara de registro en obscuridad. Cada gráfica contiene los datos de 5 repeticiones. Las distribuciones son diferentes para cada mosca. Las frecuencias mayores se encuentran entre segundos.

A continuación se presentan los resultados estadísticos de la frecuencia relativa de la latencia que tardaron las moscas en cruzar de un sensor a otro. Ver **[Fig. 4.8].** Como se puede observar existe una frecuencia mayor para tiempos menores a 18 segundos aunque hay transiciones mayores a 100 segundos. La distribución de frecuencias tuvo una media de 8.256 ± 1.34segundos. Este dato indica que los cambios inducidos de cada luz deben ser diferentes de este valor.

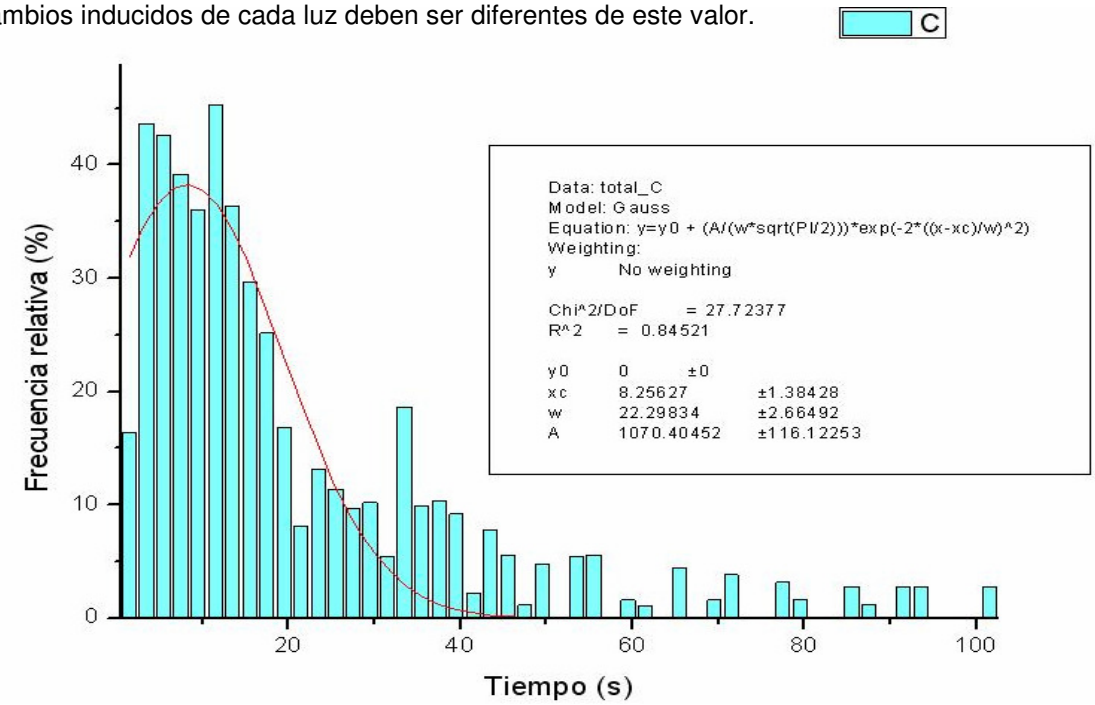


Fig. 4.8: Distribución de frecuencias de latencia presentada por la muestra de moscas para pasar de un sensor a otro dentro de la cámara de registro en obscuridad. El promedio de movimiento de la mosca entre sensores es de 8.256 segundos.

La segunda parte de los ensayos se realizó variando la intensidad de iluminación de cada color por separado para estudiar el efecto de la intensidad luminosa en la actividad locomotora. Los ensayos consistieron en obtener una prueba de 5 min de obscuridad como basal, para después hacer 8, cada una de 4 min (1 min obscuridad inicial, 2 min experimento y 1 min de obscuridad final) y al final se hizo otro ensayo de 5 min en obscuridad. Esto se repitió con tres moscas en cada color: azul **[Fig. 4.9]**, verde **[Fig. 4.10]**, ámbar **[Fig. 4.11]**, rojo **[Fig. 4.12]** y blanco **[Fig. 4.13]**.

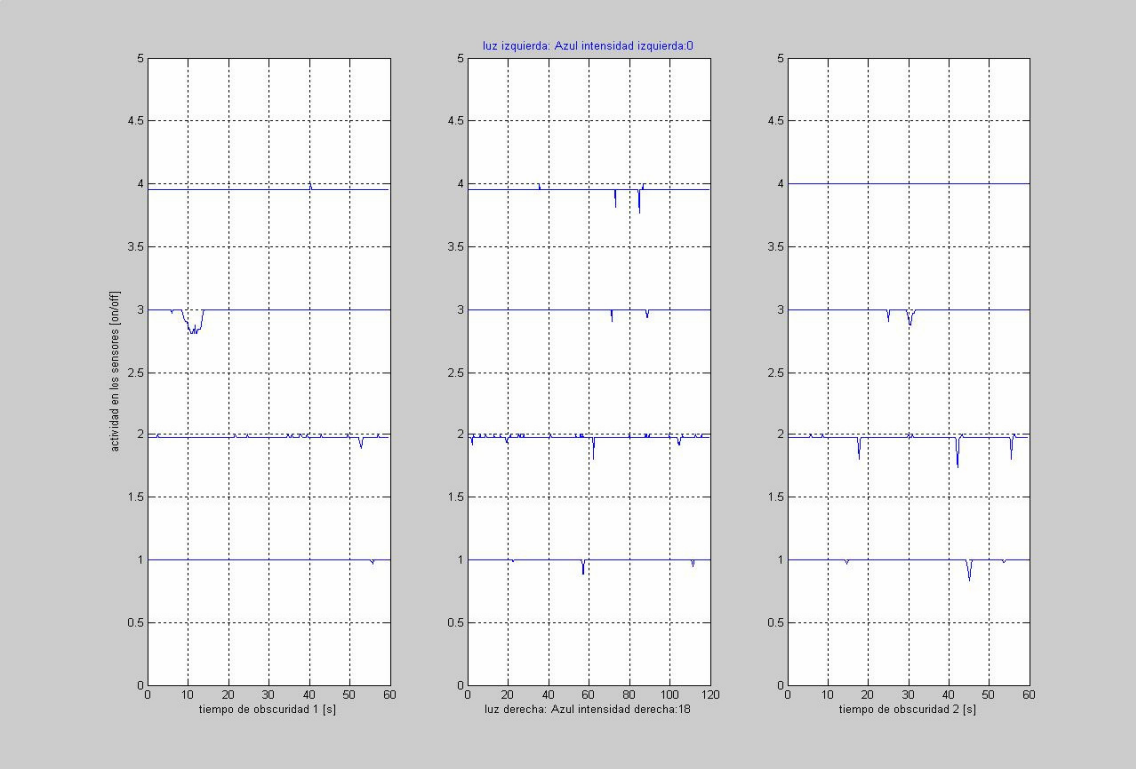


Fig. 4.9: Respuesta fotomotora a la luz azul. Extremo encendido: derecho. Flujo luminoso: 18.66 Lm. Se presenta una respuesta de huída a la luz.

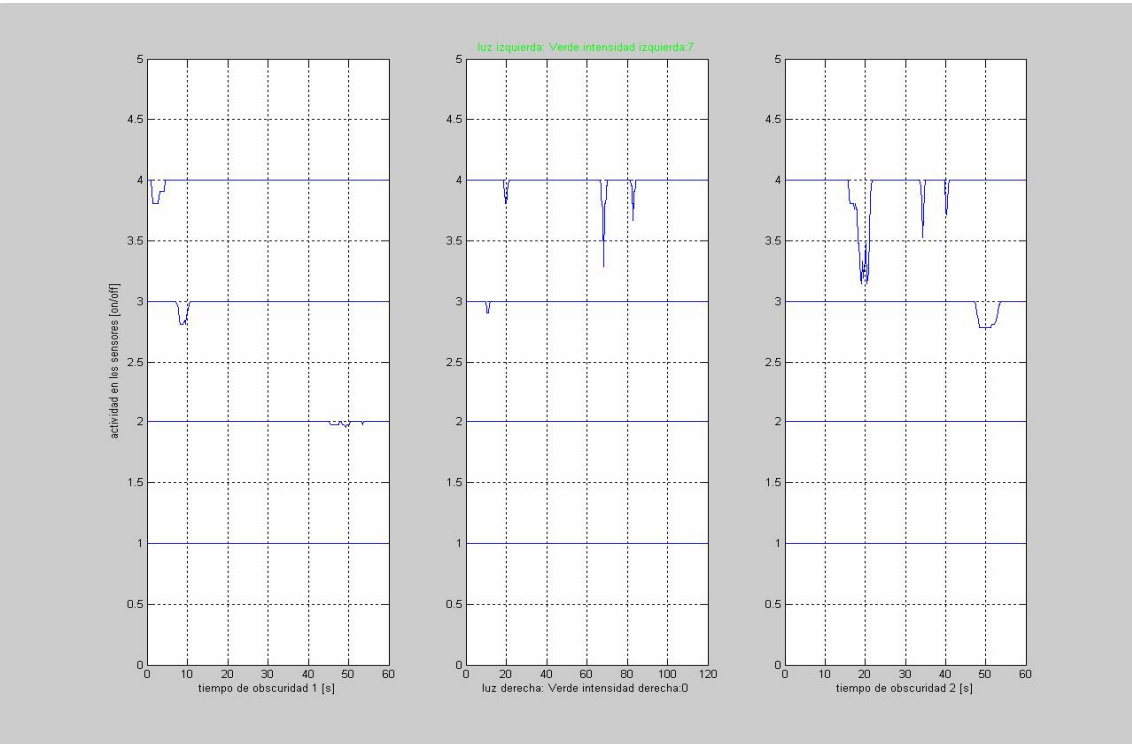


Fig. 4.10: Respuesta fotomotora a la luz verde. Extremo encendido: izquierdo. Flujo luminoso: 0.66 Lm. Se presenta una respuesta de atracción a la luz.

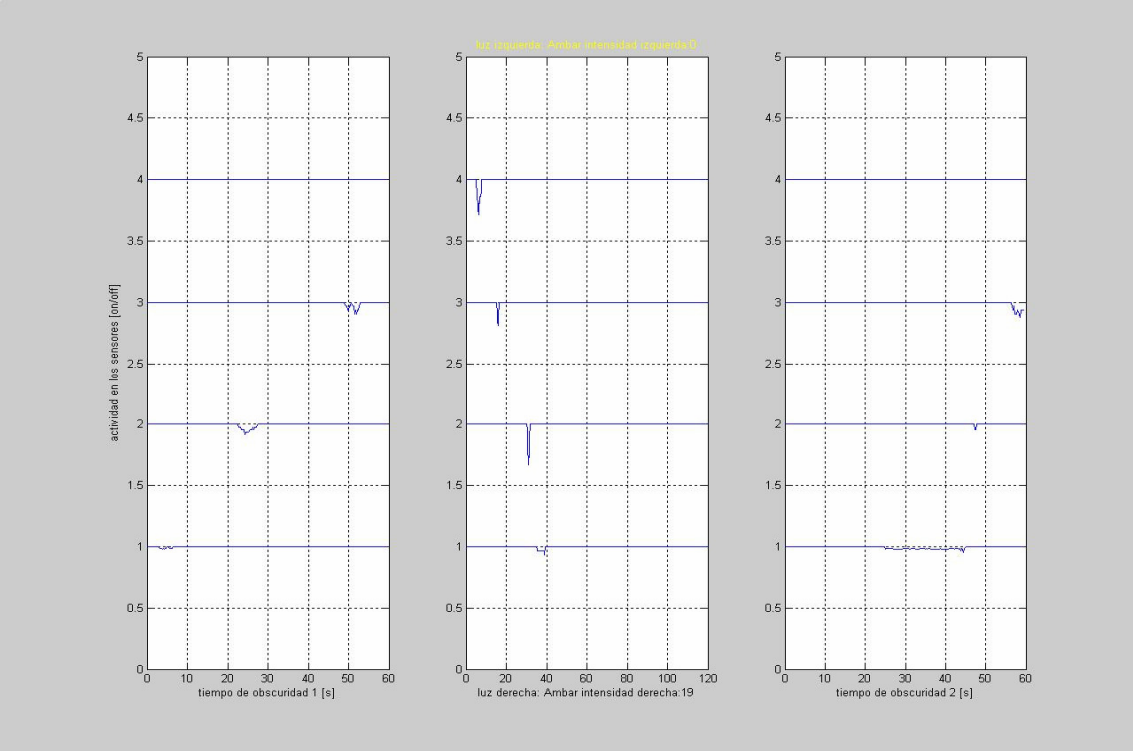


Fig. 4.11: Respuesta fotomotora a la luz ámbar. Extremo encendido: izquierdo. Intensidad relativa: 19. Flujo luminoso: 11 Lm. Se presenta una respuesta de atracción a la luz.

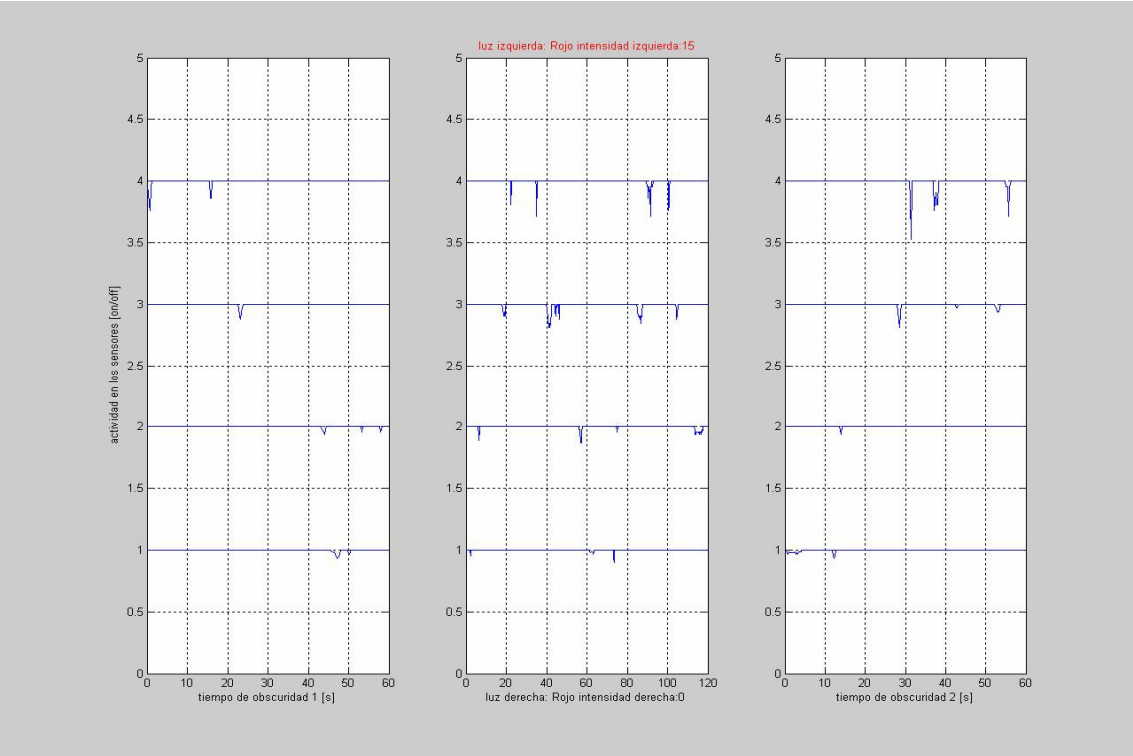


Fig. 4.12: Respuesta fotomotora a la luz roja. Extremo encendido: izquierdo. Flujo luminoso: 11.33 Lm. Se presenta una respuesta de atracción a la luz seguida por una conducta oscilatoria.

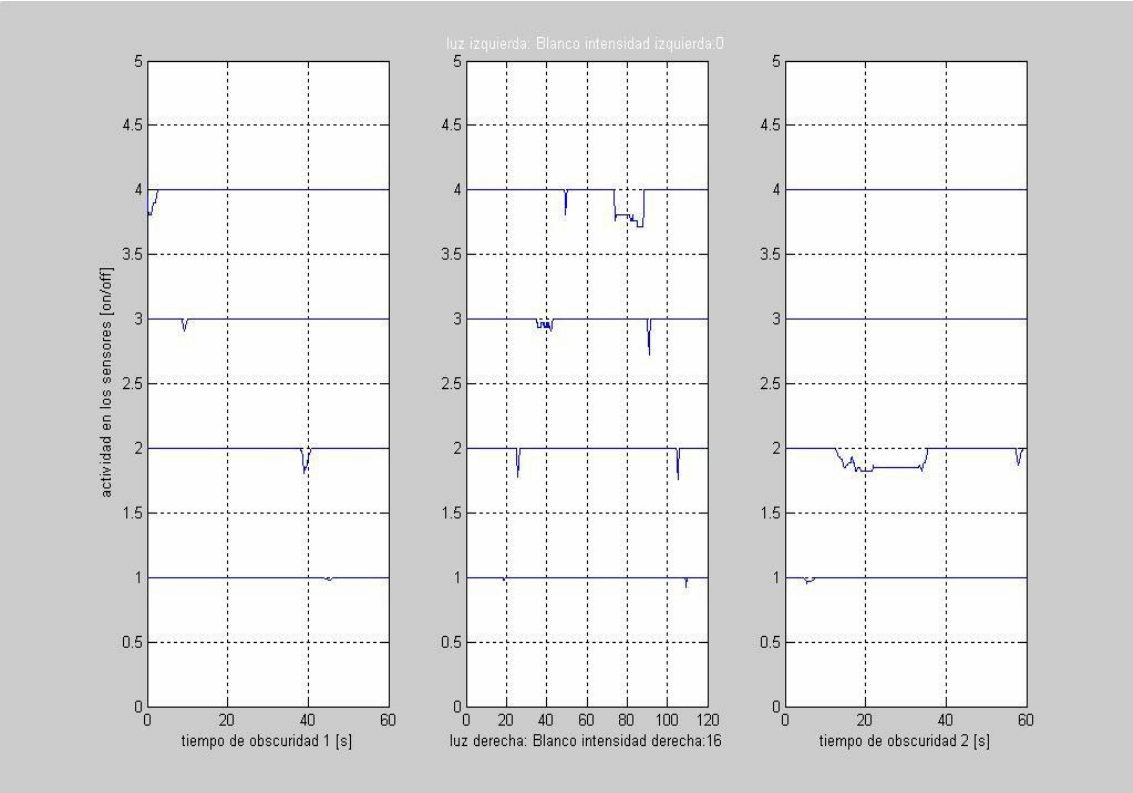


Fig. 4.13: Respuesta fotomotora a la luz blanca. Extremo encendido: derecho. Flujo luminoso: 44.66 Lm. Se presenta una respuesta de huída a la luz.

Las respuestas anteriores se muestran en el formato de Matlab en que se despliegan los resultados al usuario a través de la interfase gráfica. A continuación se presenta un ejemplo de la conducta graficada de manera continua, marcando la trayectoria de la mosca mediante líneas y especificando las distancias entre los sensores **[Fig. 4.14]**.

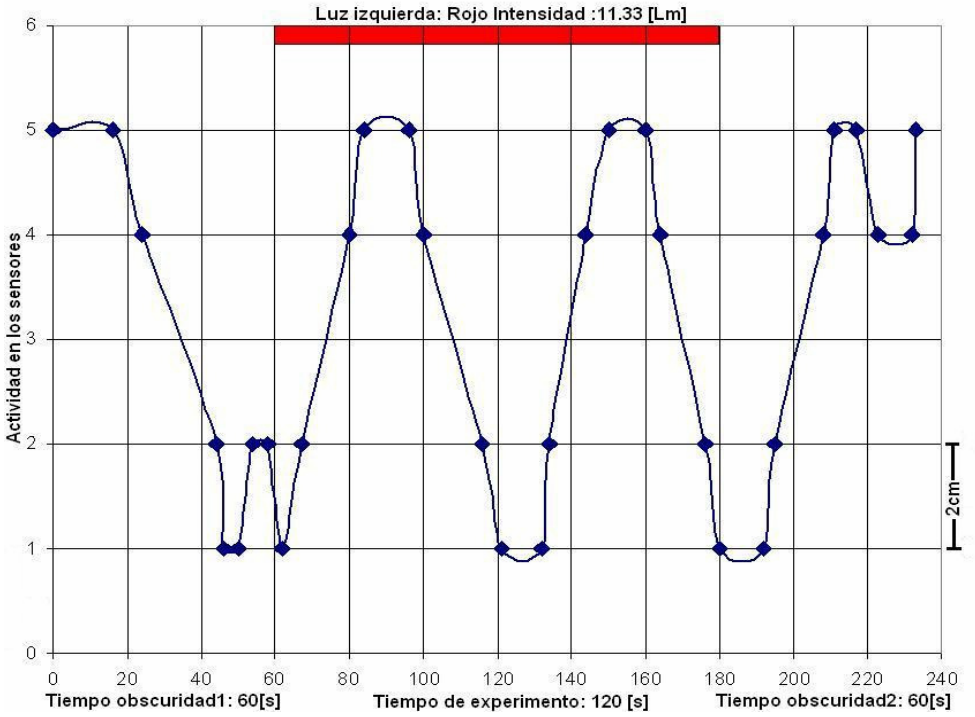


Fig. 4.14: Respuesta fotomotora a la luz roja en tiempo y distancias reales con trayectoria trazada. Extremo encendido: izquierdo. Flujo luminoso: 11.33 [Lm]. Se presenta una respuesta de atracción a la luz seguida por una conducta oscilatoria. Las latencias presentadas antes y después del tiempo de encendido de la fuente de luz son diferentes a las de que se presentan en presencia de la luz.

Con base en las respuestas obtenidas, se trabajó con los tiempos de respuesta de la mosca y las distancias entre sensores para obtener diferentes gráficas que muestran la conducta presentada por la mosca según el color y la intensidad luminosa.

Para cada color se obtuvieron gráficas de las respuestas de Atracción, Huída, Velocidad de atracción, Velocidad de huída y Periodo en función de la intensidad luminosa. Cada gráfica presenta el comportamiento de tres moscas en condiciones similares.

A continuación se muestra como ejemplo, todas las gráficas correspondientes a la luz roja **[Fig. 4.15]**, mientras que los demás colores se presentan en el **Apéndice H**.

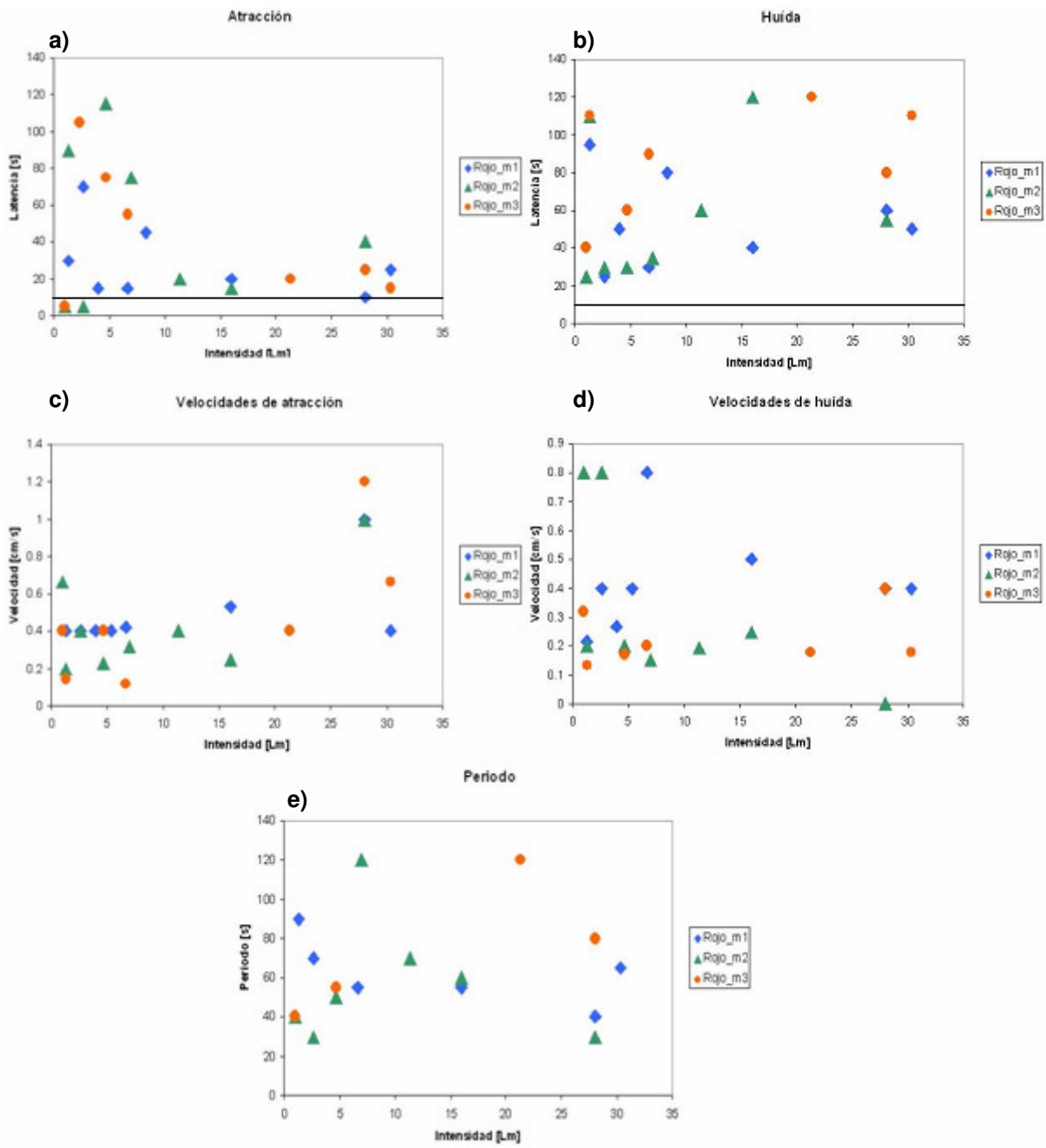


Fig. 4.15: Gráficas del análisis de comportamiento a la luz roja. a) Gráfica de atracción. Presenta una caída de los valores de latencia conforme aumenta la intensidad. b) Gráfica de huída. Conforme aumenta la intensidad también incrementa el tiempo de respuesta. c) Gráfica de velocidad de atracción. Presenta un incremento de la velocidad al aumentar la intensidad. d) Gráfica de velocidad de huída. Existe una mayor respuesta a intensidades por debajo de 15 lúmenes. e) Gráfica de periodo. Los valores de mayor respuesta se encuentran debajo de 15 lúmenes con periodos que están entre 20 y 120 segundos aproximadamente. Se debe recordar que el valor promedio de respuesta entre sensores (a 2 cm de distancia entre ellos) es: 8.256 ± 1.34 segundos (línea recta paralela al eje x en las gráficas de atracción e huída en función de la intensidad).

Discusión

Capítulo V.- Docencia e Investigación

Docencia

La iniciativa para desarrollar este proyecto surge a partir de la experiencia del cuerpo docente del laboratorio de biología del Proyecto Experimenta que se desarrolla en la Facultad de Ciencias de la UNAM.

Este centro tiene como principal objetivo que los alumnos del bachillerato vivan la ciencia desde adentro. La idea es que el estudiante acuda a laboratorios de Física, Biología, Matemáticas, Química y Tecnología a plantearse preguntas o problemáticas relacionadas con un tema determinado. Durante la sesión los instructores orientan a los alumnos para que puedan encontrar respuesta a sus interrogantes a partir de su propia investigación. En pocas palabras los estudiantes hacen ciencia en cada una de las sesiones. Por medio del planteamiento de la pregunta, la identificación y control de las variables, la experimentación, cuantificación y discusión con sus pares, los estudiantes logran entender mejor el quehacer científico.

En particular, en el laboratorio de biología se empezó trabajando con la mosca *Drosophila* y algunas de las preguntas que los muchachos se formulaban eran: ¿Cómo ven las moscas? ¿Qué colores distinguen? Ellos se ingeniaban con cartones, papel de colores y lámparas para diseñar sus experimentos y llegar a la respuesta.

Con base en ello, una iniciativa de los instructores del laboratorio fue diseñar un dispositivo para hacerlo de manera cuantitativa con control de variables y con el cual los estudiantes pudieran hacer sus experimentos y comprobar los resultados que ellos obtuvieran con sus dispositivos “caseros”. De esta forma serían más conscientes del proceso de desarrollo de tecnología enfocada a la investigación científica, sabiendo que no basta conformarse con resultados preliminares aunque sean buenos sino que es necesario seguir mejorando los experimentos para lograr una mayor certeza en los resultados.

Para que el dispositivo pueda ser utilizado por los alumnos, debe ser lo más robusto posible a la vez que versátil. Otro factor importante fue que no tuviera partes móviles para evitar desajustes. Debe ser un dispositivo lo más fácil de reproducir y económico, sin dejar de ser un instrumento eficiente y profesional.

Investigación

Por otra parte existe un gran interés en investigar la conducta de la mosca *Drosophila* debido a las posibilidades de usar genética y mutantes para conocer las bases genéticas de la conducta.

Es así que el dispositivo desarrollado también tiene utilidad dentro de la investigación básica formal. Siendo el único dispositivo hecho a la medida de las necesidades y características deseadas por los investigadores interesados en realizar los estudios mencionados anteriormente.

Cabe destacar que la mosca *Drosophila* ha sido sujeta a muchas líneas de investigación en genética debido a su facilidad y rapidez para tener generaciones de mutantes; de manera que hacemos votos para que el dispositivo diseñado sirva como una herramienta útil para obtener resultados de trascendencia científica dentro del estudio de estos seres tan peculiares.

Conclusiones y recomendaciones

Conclusiones

Ante la alta sensibilidad que muestra la mosca *Drosophila* a estímulos visuales y la necesidad de tener un aparato para cuantificar su conducta, se diseñó e implementó el dispositivo aquí presentado.

Se cuenta con un dispositivo versátil ante el control de las variables de tiempo, longitud de onda e intensidad, lo que permite al usuario tener una gran cantidad de posibilidades para diseñar experimentos según sus necesidades.

Es de fácil manejo para la introducción de parámetros a través de la interfase de usuario así como la visualización de resultados mediante gráficas donde el usuario puede seguir el historial de movimiento de la mosca a la largo de la cámara de registro.

Estas gráficas permiten determinar y cuantificar las conductas de atracción, huída, selección y adaptación a la luz u obscuridad. También se puede obtener información acerca de la posición inicial y final, velocidad y latencias.

El dispositivo en su versión actual tiene un costo de $1449.50 mn, **Apéndice I,** más el costo de fabricación de la cámara, siendo económico y de alto rendimiento considerando el arreglo óptico de lentes con el que cuenta así como el microcontrolador. Además de hacer uso de requerimientos mínimos de cómputo, **Apéndice B**. Es robusto debido a que sus partes móviles son mínimas, al igual que el mantenimiento y equipo de cómputo requerido.

Estas características permiten que el dispositivo implementado esté en condiciones de ser utilizado tanto para el área de investigación como docencia.

Recomendaciones

El diseño del dispositivo puede ser mejorado en dos ámbitos: la interfase y la electrónica.

En la primera se requiere pasar a otra plataforma que gaste menos recursos y sea de uso libre, rediseñar el formato de las gráficas e introducir tanto un botón de paro como un reloj que indique el avance del tiempo de ensayo. Asimismo la automatización del cálculo de velocidades y la gráfica de las trayectorias.

En cuanto a la electrónica se propone diseñar y anexar un circuito que avise al usuario al momento de detectar fallas en alguno de los sensores y un circuito que indique si la fuente de luz está encendida ya que la cámara no debe permitir la entrada de luz. También eliminar en mayor medida el ruido que se presenta en los sensores principalmente para tiempos largos.

Otra variante puede ser la posibilidad de que se pueda exponer a la mosca a combinaciones de luces de colores, es decir poder encender más de un LED en los extremos de la cámara.

Apéndice A

Arreglo óptico

Objetivo: Diseñar un dispositivo óptico que permita colectar la luz de cinco diodos emisores de luz, LEDs, y colimarla para que incida la luz en un tubo de 5 mm de diámetro, por el que se mueven las moscas.

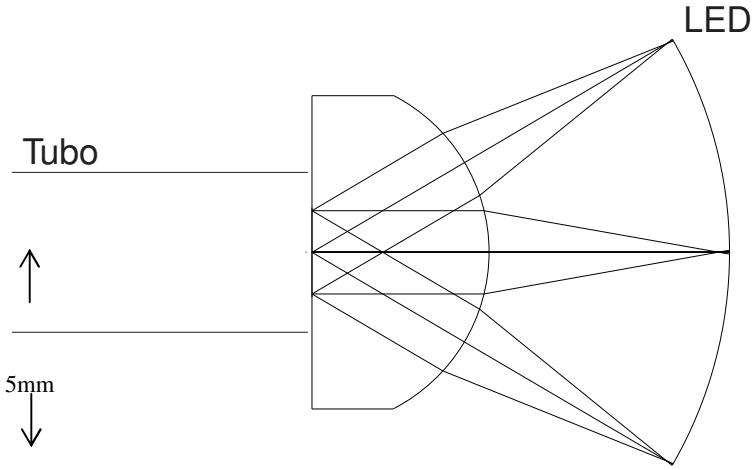


Fig. A1: EL arreglo óptico cuenta con una lente plano-convexa que colima la luz de los 5 LED's hacia una superficie de 5 mm de diámetro.

Los LED's se arreglan sobre una superficie esférica que es concéntrica con el primer vértice de la lente plano-convexa. De tal manera que la luz que sale de los LED's es colimada por la lente, iluminando el área de contacto entre el tubo y la lente. El tubo se coloca en contacto con la lente plano-convexa.

La lente tiene un espesor de 13 mm y un radio de curvatura de 13 mm y está hecha con vidrio SK4. Entre la lente y los LED's hay una distancia de 20 mm y el radio de curvatura de la montura de los LED's tiene un radio de 34 mm.

Apéndice B

Requerimientos y Especificaciones del Dispositivo

Requerimientos

Los requerimientos necesarios para tener un funcionamiento óptimo del dispositivo son:

Equipo de cómputo:

Sistema operativo: Win98/ME/2000/XP

Hardware:

Conexión puerto serie

Software:

MATLAB versión 6.5 o superiores

PICAXE Programming Editor 5.1 o superiores

Especificaciones

Las especificaciones del dispositivo son:

Intensidad Relativa[1]

Rango dinámico: 0 – 20

Resolución: 1

Tiempo para obscuridad

Rango dinámico: 1-14400 [s] (4 hrs)

Resolución: 1[s]

Tiempo para experimento

Rango dinámico: 1-14400 [s] (4 hrs)

Resolución: 1[s]

Voltaje de alimentación[2]**:** 9 V

Consumo de corriente: 120 mA

[1] El equivalente de la intensidad relativa en lúmenes se muestra en el Apéndice G

[2] El uso de baterías se recomienda sólo en casos de pruebas menores a 1 hr, en caso contrario se requiere conectar el dispositivo a un eliminador de voltaje fijo a 9V o bien una fuente de poder.

Apéndice C

PCB's para Circuito Impreso

A continuación se anexan los pcb's para realizar el circuito impreso en bicapa.

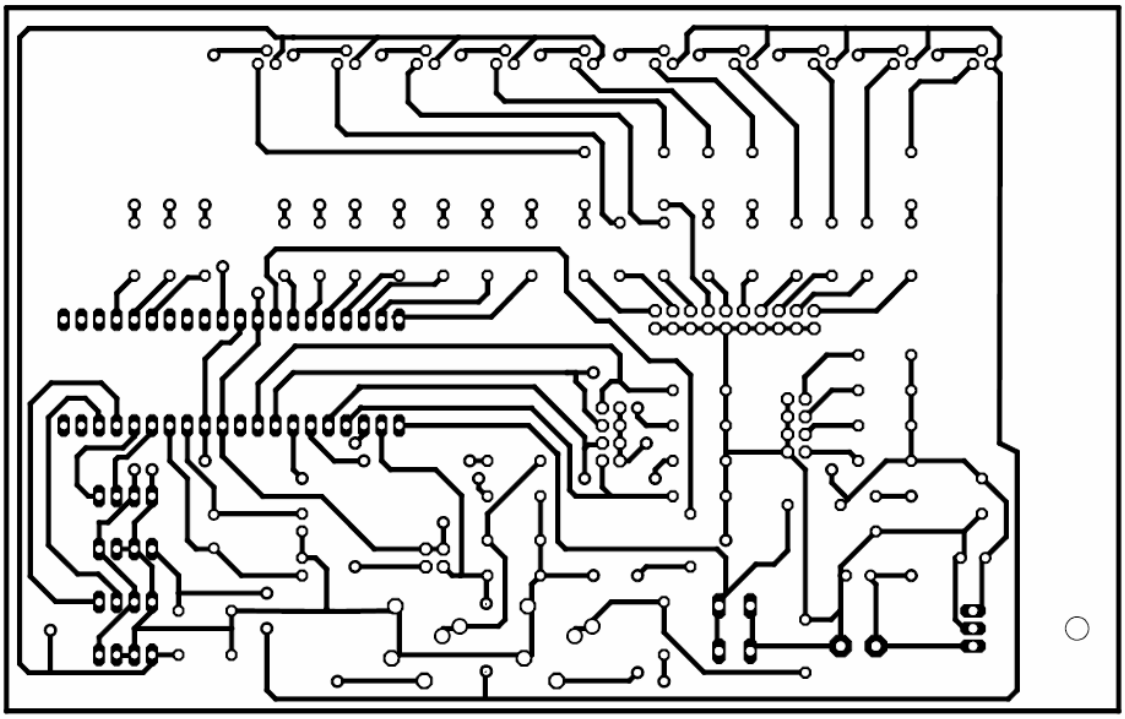


Fig. C1: Capa inferior para circuito impreso.

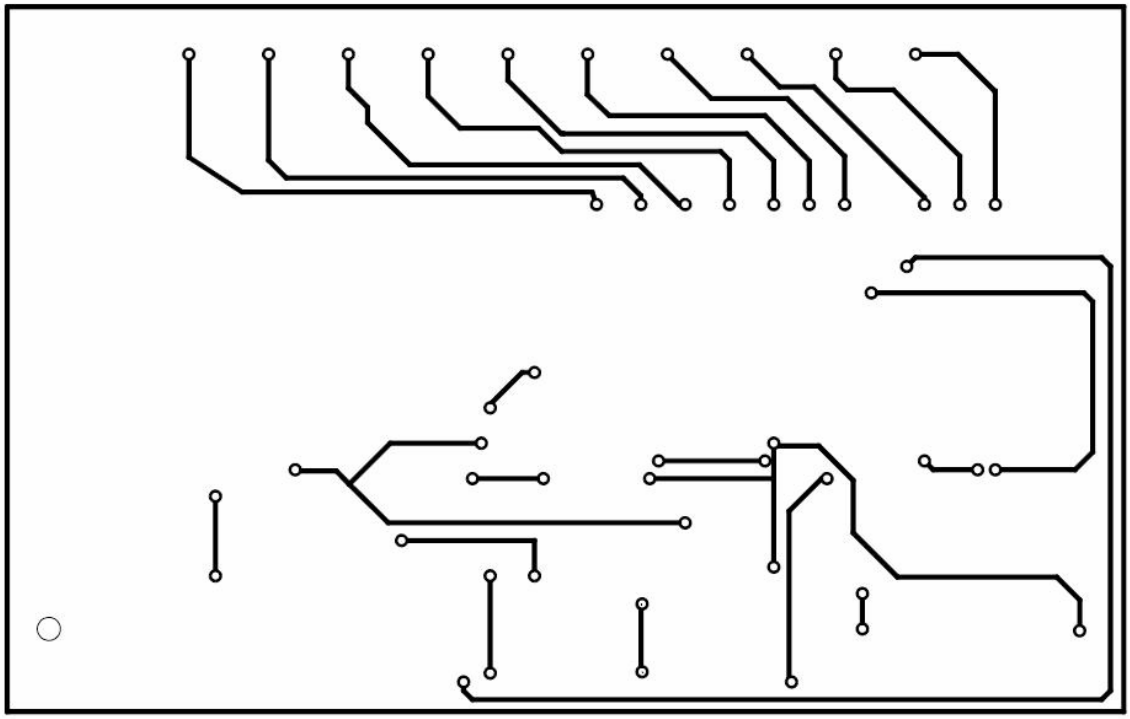


Fig. C2: Capa superior para circuito impreso.

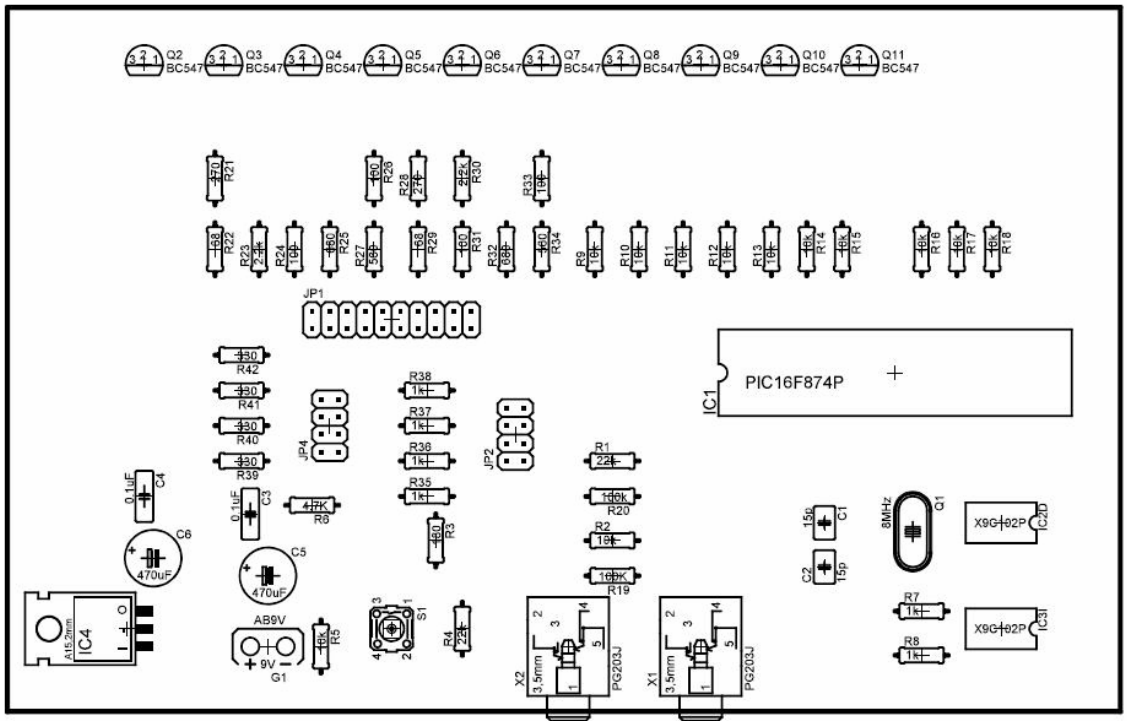


Fig. C3: Capa superior de componentes para circuito impreso[3].

[3] Nota: todos los pcb están en tamaño real.

Apéndice D

Programa del PIC

A continuación se muestra el programa desarrollado en lenguaje Basic, con el que fue programado el PIC.

```
let dirsc =%11111111
'symbol UDder= portc 0
'symbol UDizq =portc 3
'symbol INCder =portc 1
'symbol INCizq =portc 2
symbol valorder=b2
symbol valorizq=b3
symbol colorder=b4
symbol colorizq=b5
symbol duracion=w6
symbol rd=7
symbol yd=6
symbol gd=5
symbol bd=4
symbol wd=3
symbol ri=2
symbol yi=1
'symbol gi=portc 7
'symbol bi=portc 6
'symbol wi=portc 5
'ADC0 es foto der en b0
'ADC1 es foto izq en b1
'ADC2 es der interna en b7
'ADC3 es der externa en b8
'ADC5 es izq interna en b9
'ADC6 es izq externa en b10
```

Definición de variables, entradas y salidas

```
main:
      let dirsc =%11111111
      low portc 6
      low portc 5
      serin 7,N2400,b8,b9,b10
      let w0=b9*60
      let w2=b10*3600
      let w0=w0+b8+w2
      let w0=w0*4
      gosub tiempoobs
```

Adquiere el tiempo de obscuridad

Realiza la subrutina tiempo de obscuridad

Etapa de Selección de color.
El siguiente bloque de instrucciones adquiere los parámetros de intensidad y color para cada uno de los extremos de la cámara así como la duración del experimento. Una vez obtenidos estos datos realiza una serie de discriminaciones para “saber” que combinación de colores escogió el usuario y entonces realiza la subrutina correspondiente.

```
      serin 7,N2400,valorizq,valorder,colorizq,b12,colorder,b13,
            b8,b9,b10
      let w6=0
```

```
      let w3=b9*60
      let w6=b10*3600
      let w6=w6+b8+w3
      let w6=w6*4
      if colorizq="b" and colorder="b" then azulazul
      if colorizq="b" and colorder="r" then azulrojo
      if colorizq="b" and colorder="g" then azulverde
      if colorizq="b" and colorder="w" then azulblanco
      if colorizq="b" and colorder="y" then azulambar
      if colorizq="r" and colorder="b" then rojoazul
      if colorizq="r" and colorder="r" then rojorojo
      if colorizq="r" and colorder="g" then rojoverde
      if colorizq="r" and colorder="w" then rojoblanco
      if colorizq="r" and colorder="y" then rojoambar
      if colorizq="g" and colorder="b" then verdeazul
      if colorizq="g" and colorder="r" then verderojo
      if colorizq="g" and colorder="g" then verdeverde
      if colorizq="g" and colorder="w" then verdeblanco
      if colorizq="g" and colorder="y" then verdeambar
      if colorizq="w" and colorder="b" then blancoazul
      if colorizq="w" and colorder="r" then blancorojo
      if colorizq="w" and colorder="g" then blancoverde
      if colorizq="w" and colorder="w" then blancoblanco
      if colorizq="w" and colorder="y" then blancoambar
      if colorizq="y" and colorder="b" then ambarazul
      if colorizq="y" and colorder="r" then ambarrojo
      if colorizq="y" and colorder="g" then ambarverde
      if colorizq="y" and colorder="w" then ambarblanco
      if colorizq="y" and colorder="y" then ambarambar

azulazul:
      high portc 6
      high bd
      goto subirder

azulrojo:
      high portc 6
      high rd
      goto subirder

azulverde:
      high portc 6
      high gd
      goto subirder

azulblanco:
      high portc 6
      high wd
      goto subirder

azulambar:
      high portc 6
      high yd
      goto subirder
```

```
rojoazul:
      high ri
      high bd
      goto subirder

rojorojo:
      high ri
      high rd
      goto subirder

rojoverde:
      high ri
      high gd
      goto subirder

rojoblanco:
      high ri
      high wd
      goto subirder

rojoambar:
      high ri
      high yd
      goto subirder

verdeazul:
      high portc 7
      high bd
      goto subirder

verderojo:
      high portc 7
      high rd
      goto subirder

verdeverde:
      high portc 7
      high gd
      goto subirder

verdeblanco:
      high portc 7
      high wd
      goto subirder

verdeambar:
      high portc 7
      high yd
      goto subirder

blancoazul:
      high portc 5
      high bd
      goto subirder
```

```
blancorojo:
      high portc 5
      high rd
      goto subirder


blancoverde:
      high portc 5
      high gd
      goto subirder

blancoblanco:
      high portc 5
      high wd
      goto subirder

blancoambar:
      high portc 5
      high yd
      goto subirder

ambarazul:
      high yi
      high bd
      goto subirder

ambarrojo:
      high yi
      high rd
      goto subirder

ambarverde:
      high yi
      high gd
      goto subirder

ambarblanco:
      high yi
      high wd
      goto subirder

ambarambar:
      high yi
      high yd
      goto subirder
```

Etapa de control de intensidad

Las siguientes subrutinas envían los pulsos al potenciómetro digital para que éste cambie su valor de resistencia interna de tal forma que la intensidad varíe hasta llegar al valor deseado.

```
subirder:
      high portc 0
      high portc 1
      pause 1
      low portc 1
      pause 1
```

Genera los pulsos para controlar al potenciómetro digital

```
        readadc 0, b8        'Mide valor del ADC0 del PIC correspondiente al número de pasos en
                             el potenciómetro digital derecho
        if b8= 80 then incrementard
        if b8< 80 then subirder
        if b8> 80 then bajarder
        goto subirder
```

Discrimina a cuál subrutina dirigirse dependiendo del valor del ADC0

```
bajarder:
        low portc 0
        high portc 1
        pause 1
        low portc 1
        pause 1
        readadc 0, b8        'Mide valor del ADC0 del PIC correspondiente al número de pasos en
                             el potenciómetro digital derecho
        if b8= 80 then incrementard
        if b8> 80 then bajarder
        if b8< 80 then subirder
        goto bajarder
```

Genera los pulsos para controlar al potenciómetro digital

Discrimina a cuál subrutina dirigirse dependiendo del valor del ADC0

```
subirizq:
        high portc 3
        high portc 2
        pause 1
        low portc 2
        pause 1
        readadc 1, b9        'Mide valor del ADC1 del PIC correspondiente al número de pasos en
                             el potenciómetro digital izquierdo
        if b9= 80 then incrementari
        if b9< 80 then subirizq
        if b9> 80 then bajarizq
        goto subirizq
```

Genera los pulsos para controlar al potenciómetro digital

Discrimina a cuál subrutina dirigirse dependiendo del valor del ADC1

```
bajarizq:
        low portc 3
        high portc 2
        pause 1
        low portc 2
        pause 1
        readadc 1, b9        'Mide valor del ADC1 del PIC correspondiente al número de pasos en
                             el potenciómetro digital izquierdo
        if b9= 80 then incrementari
        if b9> 80 then bajarizq
        if b9< 80 then subirizq
        goto bajarizq
```

Genera los pulsos para controlar al potenciómetro digital

Discrimina a cuál subrutina dirigirse dependiendo del valor del ADC1

Cuando el ADC0 y el ADC1 han llegado al valor determinado entonces las subrutinas de incremento para derecha (incrementard) e izquierda (incrementari) se encargan de ir subiendo paso a paso el potenciómetro digital hasta el valor guardado en valorder y valorizq, determinados por el usuario, con una pausa entre cada valor para poder realizar las lecturas.

```
incrementard:
      let b9=0
      for b9=0 to valorder
      high portc 0
      high portc 1
      pause 1
      low portc 1
      pause 1
      pause 5
      next b9
      gosub subirizq

incrementari:
      let b9=0
      for b9=0 to valorizq
      high portc 3
      high portc 2
      pause 1
      low portc 2
      pause 1
      pause 5
      next b9
      gosub espera
```

Cuando el valor de intensidad ha llegado a ser el deseado, la subrutina “espera” realiza el encendido de los LED’s durante el tiempo que determinó el usuario y al finalizar apaga los LED’s.

```
espera:
      gosub tiempo
      let pins= %00000000
      low portc 7
      low portc 6
      low portc 5
      gosub tiempoobs2
      goto main
```

Monitoreo de la Mosca.
Mientras se realiza la subrutina del tiempo de duración del experimento se monitorea el valor de los sensores y se envían dichos valores hacia la computadora.

```
tiempo:
      let w3=0
      for w3=0 to w6
            pause 490
            readadc 2, b11
            let b11=b11-128
            readadc 3, b8
            let b8=b8-128
            readadc 5, b9
            let b9=b9-128
            readadc 6, b10
            let b10=b10-128
            serout 0, N2400,(b8,b11,b9,b10) ‘envía el valor a la computadora
      next w3
      return
```

Monitorea el valor de los sensores infrarrojos

```
tiempoobs:
      let w3=0
      for w3=0 to w0
            pause 488
            readadc 2, b11
            let b11=b11-128
            readadc 3, b8
            let b8=b8-128
            readadc 5, b9
            let b9=b9-128
            readadc 6, b10
            let b10=b10-128
            serout 0, N2400,(b8,b11,b9,b10) 'envía el valor a la computadora
      next w3
      return
```

Monitorea el valor de los sensores infrarrojos

```
tiempoobs2:
      let w3=0
      for w3=0 to w0
            pause 490
            readadc 2, b11
            let b11=b11-128
            readadc 3, b8
            let b8=b8-128
            readadc 5, b9
            let b9=b9-128
            readadc 6, b10
            let b10=b10-128
            serout 0, N2400,(b8,b11,b9,b10) 'envía el valor a la computadora
      next w3
      return
```

Monitorea el valor de los sensores infrarrojos

Apéndice E

Datasheets

Transistor bipolar npn

FAIRCHILD SEMICONDUCTOR®

BC546/547/548/549/550

BC546/547/548/549/550

Switching and Applications

- High Voltage: BC546, V_{CEO}=65V
- Low Noise: BC549, BC550
- Complement to BC556 ... BC560

1. Collector 2. Base 3. Emitter

NPN Epitaxial Silicon Transistor

Absolute Maximum Ratings T_a=25°C unless otherwise noted

Symbol	Parameter	Value	Units
V_{CBO}	Collector-Base Voltage : BC546 : BC547/550 : BC548/549	80 50 30	V V V
V_{CEO}	Collector-Emitter Voltage : BC546 : BC547/550 : BC548/549	65 45 30	V V V
V_{EBO}	Emitter-Base Voltage : BC546/547 : BC548/549/550	6 5	V V
I_C	Collector Current (DC)	100	mA
P_C	Collector Power Dissipation	500	mW
T_J	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-65 ~ 150	°C

Electrical Characteristics T_a=25°C unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
I_{CBO}	Collector Cut-off Current	V_{CB}=30V, I_E=0			15	nA
h_{FE}	DC Current Gain	V_{CE}=5V, I_C=2mA	110		800	
V_{CE} (sat)	Collector-Emitter Saturation Voltage	I_C=10mA, I_B=0.5mA I_C=100mA, I_B=5mA		90 200	250 600	mV mV
V_{BE} (sat)	Base-Emitter Saturation Voltage	I_C=10mA, I_B=0.5mA I_C=100mA, I_B=5mA		700 900		mV mV
V_{BE} (on)	Base-Emitter On Voltage	V_{CE}=5V, I_C=2mA V_{CE}=5V, I_C=10mA	580	660	700 720	mV mV
f_T	Current Gain Bandwidth Product	V_{CE}=5V, I_C=10mA, f=100MHz		300		MHz
C_{ob}	Output Capacitance	V_{CB}=10V, I_E=0, f=1MHz		3.5	6	pF
C_{ib}	Input Capacitance	V_{EB}=0.5V, I_C=0, f=1MHz		9		pF
NF	Noise Figure : BC546/547/548 : BC549/550 : BC549 : BC550	V_{CE}=5V, I_C=200μA f=1KHz, R_G=2KΩ V_{CE}=5V, I_C=200μA R_G=2KΩ, f=30~15000MHz		2 1.2 1.4 1.4	10 4 4 3	dB dB dB dB

h_{FE} Classification

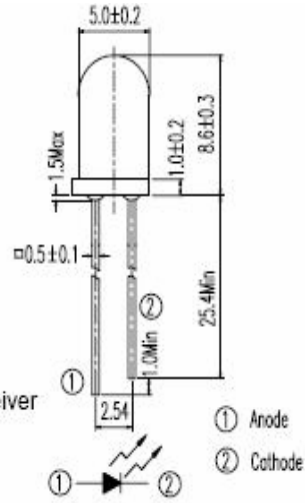
Classification	A	B	C
h_{FE}	110 ~ 220	200 ~ 450	420 ~ 800

©2002 Fairchild Semiconductor Corporation

Rev. A2, August 2002

LED emisor de infrarrojo

5mm Infrared LED MODEL NO : IR333C-A



Features :

- High radiant intensity
- Peak wavelength λp=940nm
- View angle 20°
- High reliability
- 2.54mm Lead spacing

Description :

- Infrared Emitting Diode (IR333C-A) is a high intensity diode, molded in a water clear plastic package.

The device is spectrally matched with phototransistor, photodiode and infrared receiver module.

Absolute Maximum Ratings at T_A = 25℃

Parameter	Symbol	Rating	Unit	Notice
Continuous Forward Current	I_F	50	mA	
Peak Forward Current Pulse width=100μs, Duty cycle=1%	I_{FP}	1.0	A	
Reverse Voltage	V_R	5	V	
Operating Temperature	Topr	-40 ~ +85	℃	
Storage Temperature	Tstg	-40 ~ +85	℃	
Soldering Temperature	Tsol	260	℃	4mm from mold body less than 5 seconds
Power Dissipation at(or below) 25℃ Free Air Temperature	Pd	100	mW	

Electronic Optical Characteristics :

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Radiant Intensity	Ee	7.8	20	----	mW/sr	I_F=20mA
		----	85	----		I_F=100mA,tp=100μs, t_P/T=0.01
		----	750	----		I_F=1A,tp=100μs, t_P/T=0.01
Peak Wavelength	λ_P	----	940	----	nm	I_F=20mA
Spectral Bandwidth	Δλ	----	45	----	nm	I_F=20mA
Forward Voltage	V_F	----	1.2	1.5	V	I_F=20mA
		----	1.4	1.85		I_F=100mA,tp=100μs, t_P/T=0.01
		----	2.6	4.0		I_F=1A,tp=100μs, t_P/T=0.01
Reverse Current	I_R	----	----	10	μA	V_R=5V
View Angle	2θ1/2	----	20	----	deg	I_F=20mA

Phototransistor

5mm Phototransistor MODEL NO : PT1302B/C2

Features :

- Fast response time
- High photo sensitivity
- Axial terminal
- Plastic case with IR filter

Description :

- PT1302B/C2 is a high speed and high sensitive silicon NPN phototransistor molder in a standard ϕ5 mm package. The package is an IR filter , spectrally mathch to infrared emitter diode.

Absolute Maximum Ratings at T_A = 25℃

Parameter	Symbol	Rating	Unit	Notice
Collector-Emitter Voltage	V_{CEO}	30	V	
Emitter-Collector- Voltage	V_{ECO}	5	V	
Collector Current	I_c	20	mA	
Operating Temperature	Topr	-25 ~ +85	℃	
Storage Temperature	Tstg	-40 ~ +85	℃	
Soldering Temperature	Tsol	260	℃	4mm from mold body less than 5 seconds
Power Dissipation at(or below) 25℃ Free Air Temperature	Pc	75	mW	

Electronic Optical Characteristics :

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Collector-Emitter Breakdown Voltage	BV_{CEO}	30	----	----	V	I_c=100 μA Ee=0mW/cm²
Emitter-Collector Breakdown Voltage	BV_{ECO}	5	----	----	V	I_E=100 μA Ee=0mW/cm²
Collector-Emitter Saturation Voltage	$V_{CE(sat)}$	----	----	0.4	V	I_c=2mA Ee=1mW/cm²
Rise Time	t_r	----	15	----	μS	V_{CE}=5V I_c=1mA R_L=1000Ω
Fall Time	t_f	----	15	----		
Collector Dark Current	I_{CEO}	----	----	100	nA	V_{CE}=20V Ee=0mW/cm²
On State Collector Current	$I_{C(on)}$	0.7	1.0	----	mA	V_{CE}=5V Ee=1mW/cm²
Wavelength of Peak Sensitivity	λ_p	----	980	----	nm	----
Rang of Spectral Bandwidth	$\lambda_{0.5}$	----	700---1200	----	nm	----

Potenciómetro digital

APPLICATION NOTE
AVAILABLE
AN20 • AN42–53 • AN71 • AN73 • AN88 • AN91–92 • AN115

XICOR

Terminal Voltages ±5V, 100 Taps

X9C102/103/104/503

Digitally-Controlled (XDCP) Potentiometer

FEATURES

- Solid-State Potentiometer
- Three-Wire Serial Interface
- 100 Wiper Tap Points
  - —Wiper Position Stored in Nonvolatile Memory and Recalled on Power-up
- 99 Resistive Elements
  - —Temperature Compensated
  - —End to End Resistance, ±20%
  - —Terminal Voltages, ±5V
- Low Power CMOS
  - —V_{CC} = 5V
  - —Active Current, 3mA Max.
  - —Standby Current, 500µA Max.
- High Reliability
  - —Endurance, 100,000 Data Changes per Bit
  - —Register Data Retention, 100 Years
- X9C102 = 1 kΩ
- X9C103 = 10 kΩ
- X9C503 = 50 kΩ
- X9C104 = 100 kΩ
- Packages
  - —8-Lead SOIC and DIP

DESCRIPTION

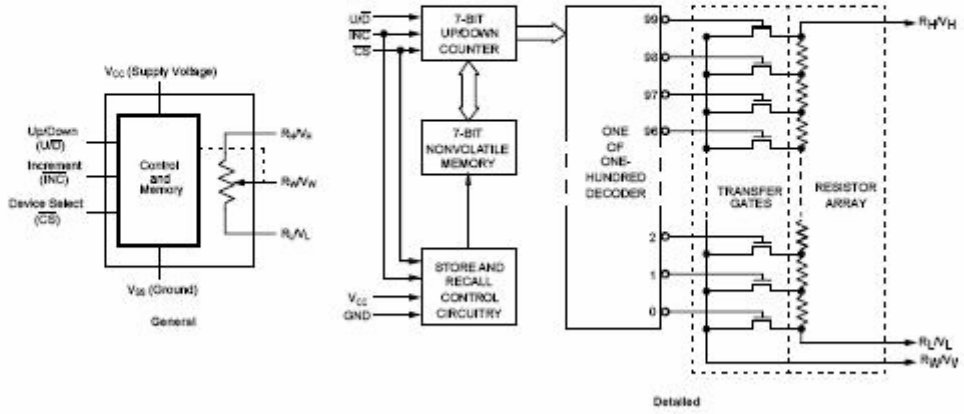
The X9Cxxx are Xicor digitally-controlled (XDCP) potentiometers. The device consists of a resistor array, wiper switches, a control section, and nonvolatile memory. The wiper position is controlled by a three-wire interface.

The potentiometer is implemented by a resistor array composed of 99 resistive elements and a wiper switching network. Between each element and at either end are tap points accessible to the wiper terminal. The position of the wiper element is controlled by the $\overline{CS}$, U/$\overline{D}$, and $\overline{INC}$ inputs. The position of the wiper can be stored in nonvolatile memory and then be recalled upon a subsequent power-up operation.

The device can be used as a three-terminal potentiometer or as a two-terminal variable resistor in a wide variety of applications including:

- control
- parameter adjustments
- signal processing

FUNCTIONAL DIAGRAMS



E²POT™ is a trademark of Xicor, Inc. 11/5/98

©Xicor, Inc. 1994, 1995 Patents Pending
3883-2.4 2/12/99 T2/C0/D0 SH

Characteristics subject to change without notice

PIN DESCRIPTIONS

R_H/V_H and R_L/V_L

The high (V_H/R_H) and low (V_L/R_L) terminals of the X9C102/103/104/503 are equivalent to the fixed terminals of a mechanical potentiometer. The minimum voltage is –5V and the maximum is +5V. The terminology of V_H/R_H and V_L/R_L references the relative position of the terminal in relation to wiper movement direction selected by the U/$\overline{D}$ input and not the voltage potential on the terminal.

R_W/V_W

V_W/R_W is the wiper terminal, and is equivalent to the movable terminal of a mechanical potentiometer. The position of the wiper within the array is determined by the control inputs. The wiper terminal series resistance is typically 40Ω.

Up/Down (U/$\overline{D}$)

The U/$\overline{D}$ input controls the direction of the wiper movement and whether the counter is incremented or decremented.

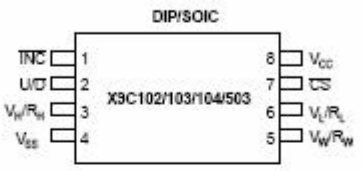
Increment ($\overline{INC}$)

The $\overline{INC}$ input is negative-edge triggered. Toggling $\overline{INC}$ will move the wiper and either increment or decrement the counter in the direction indicated by the logic level on the U/$\overline{D}$ input.

Chip Select ($\overline{CS}$)

The device is selected when the $\overline{CS}$ input is LOW. The current counter value is stored in nonvolatile memory when $\overline{CS}$ is returned HIGH while the $\overline{INC}$ input is also HIGH. After the store operation is complete the X9C102/103/104/503 device will be placed in the low power standby mode until the device is selected once again.

PIN CONFIGURATION



3863 FHD F02.2

PIN NAMES

Symbol	Description
V_H/R_H	High Terminal
V_W/R_W	Wiper Terminal
V_L/R_L	Low Terminal
V_{SS}	Ground
V_{CC}	Supply Voltage
U/$\overline{D}$	Up/Down Control Input
$\overline{INC}$	Increment Control Input
$\overline{CS}$	Chip Select Control Input
NC	No Connection

PRINCIPLES OF OPERATION

There are three sections of the X9Cxxx: the input control, counter and decode section; the nonvolatile memory; and the resistor array. The input control section operates just like an up/down counter. The output of this counter is decoded to turn on a single electronic switch connecting a point on the resistor array to the wiper output. Under the proper conditions the contents of the counter can be stored in nonvolatile memory and retained for future use. The resistor array is comprised of 99 individual resistors connected in series. At either end of the array and between each resistor is an electronic switch that transfers the potential at that point to the wiper.

The wiper, when at either fixed terminal, acts like its mechanical equivalent and does not move beyond the last position. That is, the counter does not wrap around when clocked to either extreme.

The electronic switches on the device operate in a "make before break" mode when the wiper changes tap positions. If the wiper is moved several positions, multiple taps are connected to the wiper for t_{IW} ($\overline{INC}$ to V_W change). The R_{TOTAL} value for the device can temporarily be reduced by a significant amount if the wiper is moved several positions.

When the device is powered-down, the last wiper position stored will be maintained in the nonvolatile memory. When power is restored, the contents of the memory are recalled and the wiper is set to the value last stored.

INTRUCTIONS AND PROGRAMMING

The $\overline{INC}$, U/$\overline{D}$ and $\overline{CS}$ inputs control the movement of the wiper along the resistor array. With $\overline{CS}$ set LOW the device is selected and enabled to respond to the U/$\overline{D}$ and $\overline{INC}$ inputs. HIGH to LOW transitions on $\overline{INC}$ will increment or decrement (depending on the state of the U/$\overline{D}$ input) a seven-bit counter. The output of this counter is decoded to select one of one-hundred wiper positions along the resistive array.

The value of the counter is stored in nonvolatile memory whenever $\overline{CS}$ transistions HIGH while the $\overline{INC}$ input is also HIGH.

The system may select the X9Cxxx, move the wiper, and deselect the device without having to store the latest wiper position in nonvolatile memory. After the wiper movement is performed as described above and once the new position is reached, the system must keep $\overline{INC}$ LOW while taking $\overline{CS}$ HIGH. The new wiper position will be maintained until changed by the system or until a power-down/up cycle recalled the previously stored data.

This procedure allows the system to always power-up to a preset value stored in nonvolatile memory; then during system operation minor adjustments could be made. The adjustments might be based on user preference: system parameter changes due to temperature drift, etc...

The state of U/$\overline{D}$ may be changed while $\overline{CS}$ remains LOW. This allows the host system to enable the device and then move the wiper up and down until the proper trim is attained.

MODE SELECTION

$\overline{CS}$	$\overline{INC}$	U/$\overline{D}$	Mode
L	↘	H	Wiper Up
L	↘	L	Wiper Down
↗	H	X	Store Wiper Position
H	X	X	Standby Current
↗	L	X	No Store, Return to Standby

SYMBOL TABLE

WAVEFORM	INPUTS	OUTPUTS
	Must be steady	Will be steady
	May change from Low to High	Will change from Low to High
	May change from High to Low	Will change from High to Low
	Don't Care: Changes Allowed	Changing: State Not Known
	N/A	Center Line Is High Impedance

ABSOLUTE MAXIMUM RATINGS*

Temperature under Bias .....–65°C to +135°C
Storage Temperature .....–65°C to +150°C
Voltage on $\overline{CS}$, $\overline{INC}$, U/$\overline{D}$ and V_{CC}
with Respect to V_{SS} ..... –1V to +7V
Voltage on V_H and V_L
Referenced to V_{SS} ..... –8V to +8V
$\Delta V = |V_H – V_L|$
X9C102 .....4V
X9C103, X9C503, and X9C104.....10V
Lead Temperature (Soldering, 10 seconds)..... +300°C

***COMMENT**

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and the functional operation of the device at these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED OPERATING CONDITIONS

Temperature	Min.	Max.
Commercial	0°C	+70°C
Industrial	–40°C	+85°C
Military	–55°C	+125°C

3863 PGM T03.1

Supply Voltage (V_{CC})	Limits
X9C102/103/104/503	5V ±10%

3863 PGM T04.2

POTENTIOMETER CHARACTERISTICS (Over recommended operating conditions unless otherwise stated.)

Symbol	Parameter	Limits			Units	Test Conditions/Notes
		Min.	Typ.	Max.		
R_{TOTAL}	End to End Resistance Variation	–20		+20	%	
V_{VH}	V_H Terminal Voltage	–5		+5	V	
V_{VL}	V_L Terminal Voltage	–5		+5	V	
	Power Rating			16	mW	X9C102
	Power Rating			10	mW	X9C103/104/503
I_W	Wiper Current			±1	mA	
R_W	Wiper Resistance		40	100	Ω	Wiper Current = ±1mA
	Noise		–120		dBV	Ref. 1kHz
	Resolution		1		%	
	Absolute Linearity(1)	–1		+1	M(3)	$V_{W(n)(actual)} – V_{W(n)(expected)}$
	Relative Linearity(2)	–0.2		+0.2	Ml(3)	$V_{W(n+1)(actual)} – [V_{W(n)} + Ml]$
	RTOTAL Temperature Coefficient		±300		ppm/°C	X9C103/503/104
	RTOTAL Temperature Coefficient		±600		ppm/°C	X9C102
	Ratiometric Temperature Coefficient			±20	ppm°C	
$C_H/C_L/C_W$	Potentiometer Capacitances		10/10/25		pF	see circuit #3

Notes: (1) Absolute Linearity is utilized to determine actual wiper voltage versus expected voltage = $[V_{W(n)(actual)} – V_{W(n)(expected)}] = ±1$ Ml Maximum.
(2) Relative Linearity is a measure of the error in step size between taps = $V_{W(n+1)} – [V_{W(n)} + Ml] = +0.2$ Ml.
(3) 1 Ml = Minimum Increment = $R_{TOT}/99$
(4) Typical values are for T_A = 25°C and nominal supply voltage.
(5) This parameter is periodically sampled and not 100% tested.

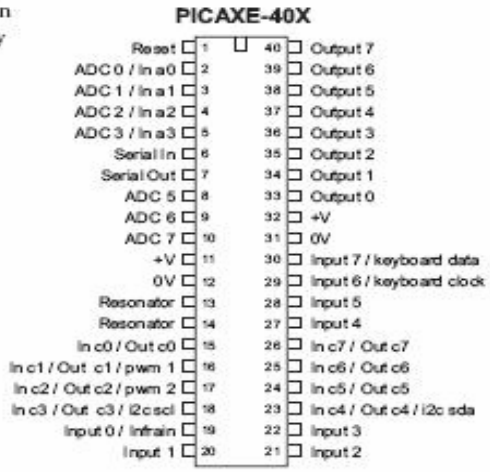
Patigrama PICAXE 40X

Configuring PICAXE-40X Input-Output Pins

To provide greater flexibility, the input/output pin configiration of the PICAXE-40X can be varied by the user.

PORTA (legs 2 to 5) provide 4 analogue inputs (default) or up to 4 digital inputs.
PORTB (leg 32 to 40) provide 8 fixed outputs.
PORTC (leg 15-18. 23-26) provide 8 digital inputs (default) or up to 8 outputs.
PORTD (leg 19-22, 27-30) provide 8 digital inputs
PORTE (leg 8 to 10) provide 3 analogue inputs

This gives a maximum of 20 digital inputs, 16 outputs, 7 analogue inputs



PORTA Functions

Leg	Default Function	Second Function
2	analogue 0	porta input 0
3	analogue 1	porta input 1
4	analogue 2	porta input 2
5	analogue	porta input 3

PORTB / PORTE Functions

PORTB pins are fixed as outputs and cannot be altered.
PORTE pins are fixed as analogue inputs and cannot be altered.

PORTC Functions

Leg	Default	Second Function	Special Function
15	input portc 0	output portc 0	
16	input portc 1	output portc 1	pwm 1 (output)
17	input portc 2	output portc 2	pwm 2 (output)
18	input portc 3	output portc 3	i2c scl clock (input)
23	input portc 4	output portc 4	i2c sda data (input)
24	input portc 5	output portc 5	
25	input portc 6	output portc 6	
26	input portc 7	output portc 7	

The portC pins can be used as the default inputs, changed to outputs, or used with their special function via use of the i2cslave or pwmout command

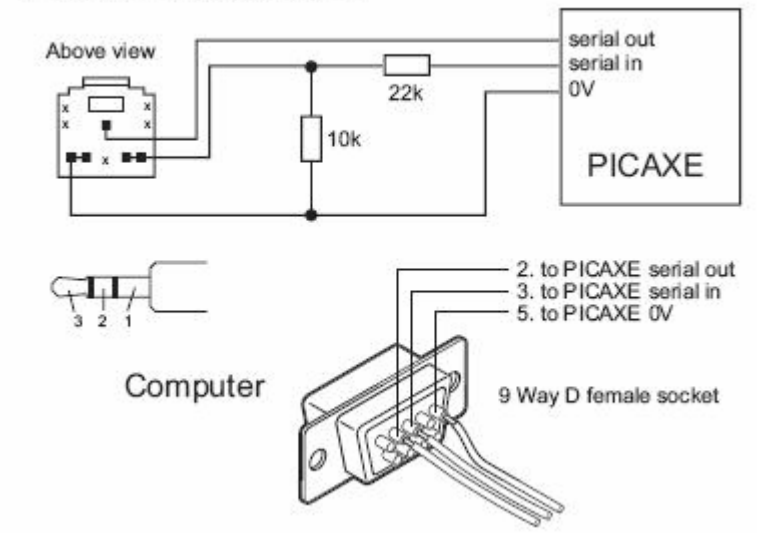
PORTD Functions

Leg	Default Function	Special Function
19	input 0	infrared (input)
20	input 1	
21	input 2	
22	input 3	
27	input 4	
28	input 5	
29	input 6	keyboard clock (input)
30	input 7	keyboard data (input)

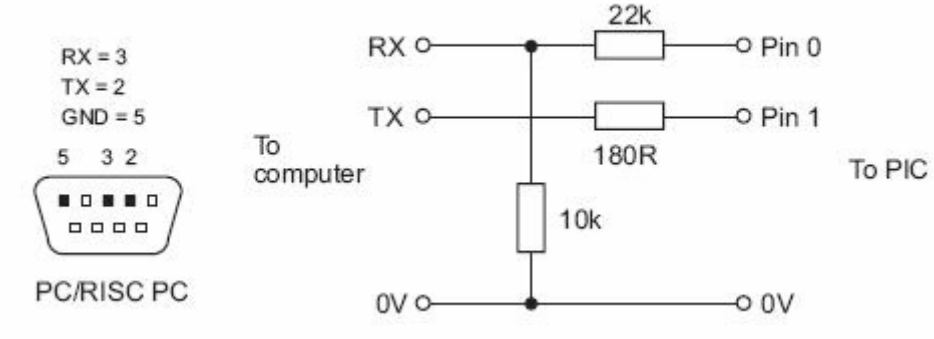
Notes:

1) The 10k/22k resistors must be included for reliable operation.
2) The reset pin must be tied high with the 4k7 resistor to operate.
3) An external 4MHz 3-pin ceramic resonator is required.

Circuito para programación



Circuito para comunicación serie de datos



Apéndice F

Programa de la GUI en Matlab

```
%Inicio de programa generado automática por GUIDE
function varargout = interface_dispositivo(varargin)
%  INTERFACE_DISPOSITIVO M-file for interface_dispositivo.fig
%  INTERFACE_DISPOSITIVO, by itself, creates a new INTERFACE_DISPOSITIVO or raises %  the
existing singleton*.
%  H = INTERFACE_DISPOSITIVO returns the handle to a new INTERFACE_DISPOSITIVO
%  or the handle to the existing singleton*.
%      INTERFACE_DISPOSITIVO('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in INTERFACE_DISPOSITIVO.M with the given input
%  arguments.
%   INTERFACE_DISPOSITIVO('Property','Value',...) creates a new
%   INTERFACE_DISPOSITIVO or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before interface_dispositivo_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to interface_dispositivo_OpeningFcn via varargin.
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help interface_dispositivo

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @interface_dispositivo_OpeningFcn, ...
                   'gui_OutputFcn',  @interface_dispositivo_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin & isstr(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 interface_dispositivo is made visible.
function interface_dispositivo_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to interface_dispositivo (see VARARGIN)
```

```
% Choose default command line output for interface_dispositivo
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes interface_dispositivo wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = interface_dispositivo_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 during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

%Se obtiene y guarda en la variable str3 el valor correspondiente a horas del tiempo de
obscuridad
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global str3
str3=str2double(get(handles.edit1,'String'));
if str3>5
   errordlg('el valor debe ser menor a 5!', 'Error Dialog Box', 'modal');
end

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

%Se obtiene y guarda en la variable str4 el valor correspondiente a minutos del tiempo de obscuridad

```
function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global str4
str4=str2double(get(handles.edit2,'String'));
if str4>60
    errordlg('el valor debe ser menor a 60!', 'Error Dialog Box', 'modal');
end

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

%Se obtiene y guarda en la variable str5 el valor correspondiente a segundos del tiempo de obscuridad

```
function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global str5
str5=str2double(get(handles.edit3,'String'));
if str5>60
    errordlg('el valor debe ser menor a 60!', 'Error Dialog Box', 'modal');
end

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

%Se obtiene y guarda en la variable str6 el valor correspondiente a horas del tiempo de
experimento
function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global str6
str6=str2double(get(handles.edit4,'String'));
if str6>5
   errordlg('el valor debe ser menor a 5!', 'Error Dialog Box', 'modal');
end

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

%Se obtiene y guarda en la variable str7 el valor correspondiente a minutos del tiempo de
experimento
function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global str7
str7=str2double(get(handles.edit5,'String'));
if str7>60
   errordlg('el valor debe ser menor a 60!', 'Error Dialog Box', 'modal');
end

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5 as a double

% --- Executes during object creation, after setting all properties.
```

```
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

%Se obtiene y guarda en la variable str8 el valor correspondiente a segundos del tiempo de
experimento
function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global str8
str8=str2double(get(handles.edit6,'String'));
if str8>60
   errordlg('el valor debe ser menor a 60!', 'Error Dialog Box', 'modal');
end

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6 as a double


% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

%Se obtiene y guarda en la variable colorizq el color seleccionado del lado izquierdo
% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global colorizq
   colorizq = get(hObject,'Value');
  return

% Hints: contents = get(hObject,'String') returns popupmenu1 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1
```

```
% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

%Se obtiene y guarda en la variable colorder el color seleccionado del lado derecho
% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global colorder
   colorder = get(hObject,'Value');
  return

% Hints: contents = get(hObject,'String') returns popupmenu2 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu2

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

%Se obtiene y guarda en la variable str1 el valor de intensidad para el lado izquierdo
function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global str1
str1=str2double(get(handles.edit7,'String'));
if str1>21
   errordlg('el valor debe ser menor a 21!', 'Error Dialog Box', 'modal');
end

% Hints: get(hObject,'String') returns contents of edit7 as text
%        str2double(get(hObject,'String')) returns contents of edit7 as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

%Se obtiene y guarda en la variable str2 el valor de intensidad para el lado derecho
function edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global str2
str2=str2double(get(handles.edit8,'String'));
if str2>21
   errordlg('el valor debe ser menor a 21!', 'Error Dialog Box', 'modal');
end

% Hints: get(hObject,'String') returns contents of edit8 as text
%        str2double(get(hObject,'String')) returns contents of edit8 as a double



% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Declaración de variables, envío y recepción de datos
global colorizq
global colorder
global str1
global str2
global str3
global str4
global str5
global str6
global str7
global str8
global str9
global str19
global str10
global str110
global rdata
str19=3600*str3+60*str4+str5
```

```
str9=str19*4;
enviard(str5);
enviard(str4);
enviard(str3);
recibir1
matrizando1(rdata)
enviard(str1);
enviard(str2);
if colorizq==2
   enviarc('b');
elseif colorizq==3
   enviarc('g');
elseif colorizq== 4
   enviarc('y');
elseif colorizq== 5
   enviarc('r');
elseif colorizq==6
   enviarc('w');
elseif colorizq==7
   enviarc('p');
end
if colorder==2
   enviarc('b');
elseif colorder==3
   enviarc('g');
  elseif colorder== 4
   enviarc('y');
  elseif colorder== 5
   enviarc('r');
  elseif colorder==6
   enviarc('w');
  elseif colorizq==7
   enviarc('p');
end
str110=3600*str6+60*str7+str8
str10=str110*4;
enviard(str8);
enviard(str7);
enviard(str6);
recibir2
matrizando2(rdata)
recibir3
matrizando3(rdata)

function enviarc(msj)
s=serial('com1','baudrate',4800);
fopen(s)
fprintf(s, msj);
fclose(s)
return

function enviard(msj)
s=serial('com1','baudrate',4800);
fopen(s)
fwrite(s,msj)
```

```
fclose(s)
return

function rdata=recibir1
global rdata
global str9
global str11
global str19
str11=5*str9;
s=serial('com1','baudrate',4800,'Timeout',str19,'InputBufferSize',str11);
fopen(s)
rdata=fread(s)
fclose(s)
return

function rdata=recibir2
global rdata
global str10
global str110
global str12
str12=5*str10;
s=serial('com1','baudrate',4800,'Timeout',str110,'InputBufferSize',str12);
fopen(s)
rdata=fread(s)
fclose(s)
return

function rdata=recibir3
global rdata
global str9
global str11
global str19
str11=5*str9;
s=serial('com1','baudrate',4800,'Timeout',str19,'InputBufferSize',str11);
fopen(s)
rdata=fread(s)
fclose(s)
return
```

%Acomodo en una matriz de los datos recibidos correspondientes al tiempo de obscuridad inicial

```
function datos=matrizando1(A)
global str9
global str19
global str1
global str2
global t
[filas columnas]=size(A);
filas2=filas/4;
B=zeros(filas2,4);
ra=0;
for rb=1:filas2,
```

```
    for j=1:4,
        ra=ra+1;
        B(rb,j)=A(ra,1);
    end
end
t=zeros(filas2,1);
filas3=(filas2/4)-0.25;
for i=0:0.25:filas3
    j=4*i+1;
    t(j,1)=i;
end
c1=zeros(filas2,1);
for i=1:filas2
    c1(i,1)=B(i,1);
end
C1=zeros(filas2,1);
Z1=max(c1);
C1=1.*c1./Z1;
c2=zeros(filas2,1);
for i=1:filas2
    c2(i,1)=B(i,2);
end
C2=zeros(filas2,1);
Z2=max(c2);
C2=2.*c2./Z2;
c3=zeros(filas2,1);
for i=1:filas2
    c3(i,1)=B(i,3);
end
C3=zeros(filas2,1);
Z3=max(c3);
C3=3.*c3./Z3;
c4=zeros(filas2,1);
for i=1:filas2
    c4(i,1)=B(i,4);
end
C4=zeros(filas2,1);
Z4=max(c4);
C4=4.*c4./Z4;

%Gráfica de datos de tiempo de obscuridad inicial
figure
subplot(131),plot(t,C1,'b',t,C2,'b',t,C3,'b',t,C4,'b')
grid
axis([0 str19 0 5])
xlabel('tiempo de obscuridad 1 [s]')
ylabel('actividad en los sensores [on/off]')
return

%Acomodo en una matriz de los datos recibidos correspondientes al tiempo de experimento
function datos=matrizando2(A)
global str10
global str110
global colorizq
global colorder
```

```
global t
[filas columnas]=size(A);
filas2=filas/4;
B=zeros(filas2,4);
ra=0;
for rb=1:filas2,
   for j=1:4,
      ra=ra+1;
      B(rb,j)=A(ra,1);
   end
end
t=zeros(filas2,1);
filas3=(filas2/4)-0.25;
for i=0:0.25:filas3
   j=4*i+1;
   t(j,1)=i;
end
c1=zeros(filas2,1);
for i=1:filas2
   c1(i,1)=B(i,1);
end
C1=zeros(filas2,1);
Z1=max(c1);
C1=1.*c1./Z1;
c2=zeros(filas2,1);
for i=1:filas2
   c2(i,1)=B(i,2);
end
C2=zeros(filas2,1);
Z2=max(c2);
C2=2.*c2./Z2;
c3=zeros(filas2,1);
for i=1:filas2
   c3(i,1)=B(i,3);
end
C3=zeros(filas2,1);
Z3=max(c3);
C3=3.*c3./Z3;
c4=zeros(filas2,1);
for i=1:filas2
   c4(i,1)=B(i,4);
end
C4=zeros(filas2,1);
Z4=max(c4);
C4=4.*c4./Z4;

%Gráfica de datos de tiempo de obscuridad experimento
subplot(132),plot(t,C1,'b',t,C2,'b',t,C3,'b',t,C4,'b')
global colorizq
global str1
if colorizq==2
title(['luz izquierda: Azul ','intensidad izquierda:',num2str(str1)],'Color','b')
elseif colorizq==3
title(['luz izquierda: Verde ','intensidad izquierda:',num2str(str1)],'Color','g')
elseif colorizq== 4
```

```
title(['luz izquierda: Ambar ','intensidad izquierda:',num2str(str1)],'Color','y')
elseif colorizq== 5
title(['luz izquierda: Rojo ','intensidad izquierda:',num2str(str1)],'Color','r')
elseif colorizq==6
title(['luz izquierda: Blanco ','intensidad izquierda:',num2str(str1)],'Color','w')
end
global colorder
global str2
if colorder==2
xlabel(['luz derecha: Azul ','intensidad derecha:',num2str(str2)],'Color','b')
elseif colorder==3
xlabel(['luz derecha: Verde ','intensidad derecha:',num2str(str2)],'Color','g')
elseif colorder== 4
xlabel(['luz derecha: Ambar ','intensidad derecha:',num2str(str2)],'Color','y')
elseif colorder== 5
xlabel(['luz derecha: Rojo ','intensidad derecha:',num2str(str2)],'Color','r')
elseif colorder==6
xlabel(['luz derecha: Blanco ','intensidad derecha:',num2str(str2)],'Color','w')
end
grid
axis([0 str110 0 5])
return

%Acomodo en una matriz de los datos recibidos correspondientes al tiempo de obscuridad
final
function datos=matrizando3(A)
global str9
global str19
global str1
global str2
global t
[filas columnas]=size(A);
filas2=filas/4;
B=zeros(filas2,4);
ra=0;
for rb=1:filas2,
   for j=1:4,
      ra=ra+1;
      B(rb,j)=A(ra,1);
   end
end
t=zeros(filas2,1);
filas3=(filas2/4)-0.25;
for i=0:0.25:filas3
   j=4*i+1;
   t(j,1)=i;
end
c1=zeros(filas2,1);
for i=1:filas2
   c1(i,1)=B(i,1);
end
C1=zeros(filas2,1);
Z1=max(c1);
C1=1.*c1./Z1;
c2=zeros(filas2,1);
```

```
for i=1:filas2
   c2(i,1)=B(i,2);
end
C2=zeros(filas2,1);
Z2=max(c2);
C2=2.*c2./Z2;
c3=zeros(filas2,1);
for i=1:filas2
   c3(i,1)=B(i,3);
end
C3=zeros(filas2,1);
Z3=max(c3);
C3=3.*c3./Z3;
c4=zeros(filas2,1);
for i=1:filas2
   c4(i,1)=B(i,4);
end
C4=zeros(filas2,1);
Z4=max(c4);
C4=4.*c4./Z4;

%Gráfica de datos de tiempo de obscuridad final
subplot(133),plot(t,C1,'b',t,C2,'b',t,C3,'b',t,C4,'b')
grid
axis([0 str19 0 5])
xlabel('tiempo de obscuridad 2 [s]')
return
```

Apéndice G

Cuantificación de la Intensidad Luminosa

Intensidad relativa	Intensidad [Lm] a 1 cm	Intensidad [Lm] a 2 cm	Intensidad [Lm] a 3 cm	Intensidad [Lm] a 4 cm
1	0.66	0.66	0.66	0.66
2	0.66	0.66	0.66	0.66
3	1	0.66	0.66	0.66
4	1	1	0.66	0.66
5	1	1	0.66	0.66
6	1.33	1	0.66	0.66
7	1.33	1	0.66	0.66
8	2.33	1.33	0.66	0.66
9	2.33	1.33	1	0.66
10	2.66	2.33	1	0.66
11	4	2.33	1	0.66
12	4.66	2.66	1	0.66
13	6.33	3	1.33	0.66
14	7	4.66	1.33	1
15	8.66	6.33	2.33	1
16	11.33	7	2.66	1.33
17	15.66	9	3	1.33
18	21.33	13.33	4.66	2.33
19	24.33	15.66	6.33	2.33
20	28	18.33	6.66	2.66

Tabla G1: Caracterización del LED rojo izquierdo, se muestra la intensidad relativa con su respectiva intensidad en lúmenes a 1, 2, 3 y 4 centímetros de la zona donde incide la luz homogéneamente.

Intensidad relativa	Intensidad [Lm] a 1 cm	Intensidad [Lm] a 2 cm	Intensidad [Lm] a 3 cm	Intensidad [Lm] a 4 cm
1	0.66	0.66	0.66	0.66
2	0.66	0.66	0.66	0.66
3	0.66	0.66	0.66	0.66
4	1	0.66	0.66	0.66
5	1	0.66	0.66	0.66
6	1.33	0.66	0.66	0.66
7	1.33	1	0.66	0.66
8	2.33	1	0.66	0.66
9	2.66	1	0.66	0.66
10	3	1	0.66	0.66
11	4	1.33	1	0.66
12	4.66	1.33	1	1
13	5	2	1	1
14	6.66	2.33	1	1
15	8.33	2.66	1.33	1
16	10.66	3	1.33	1.33
17	13	4.66	2.33	1.33
18	16	6.33	2.66	2.33
19	22	7	4	2.66
20	30.33	10.66	4.66	3

Tabla G2: Caracterización del LED rojo derecho, se muestra la intensidad relativa con su respectiva intensidad en lúmenes a 1, 2, 3 y 4 centímetros de la zona donde incide la luz homogéneamente.

Intensidad relativa	Intensidad [Lm] a 1 cm	Intensidad [Lm] a 2 cm	Intensidad [Lm] a 3 cm	Intensidad [Lm] a 4 cm
1	0.66	0.66	0.66	0.66
2	0.66	0.66	0.66	0.66
3	0.66	0.66	0.66	0.66
4	0.66	0.66	0.66	0.66
5	0.66	0.66	0.66	0.66
6	0.66	0.66	0.66	0.66
7	0.66	0.66	0.66	0.66
8	1	0.66	0.66	0.66
9	1	0.66	0.66	0.66
10	1	0.66	0.66	0.66
11	1	1	0.66	0.66
12	1.33	1	0.66	0.66
13	1.33	1	0.66	0.66
14	2	1	0.66	0.66
15	2.33	1	0.66	0.66
16	2.66	1.33	0.66	0.66
17	3	1.33	0.66	0.66
18	4	1.33	1	0.66
19	4.33	2.33	1	0.66
20	4.66	2.33	1	0.66

Tabla G3: Caracterización del LED ámbar izquierdo, se muestra la intensidad relativa con su respectiva intensidad en lúmenes a 1, 2, 3 y 4 centímetros de la zona donde incide la luz homogéneamente.

Intensidad relativa	Intensidad [Lm] a 1 cm	Intensidad [Lm] a 2 cm	Intensidad [Lm] a 3 cm	Intensidad [Lm] a 4 cm
1	0.66	0.66	0.66	0.66
2	0.66	0.66	0.66	0.66
3	0.66	0.66	0.66	0.66
4	1	0.66	0.66	0.66
5	1	0.66	0.66	0.66
6	1.33	0.66	0.66	0.66
7	1.33	0.66	0.66	0.66
8	2.33	1	0.66	0.66
9	2.66	1	0.66	0.66
10	3	1	0.66	0.66
11	4.33	1.33	1	0.66
12	5	1.33	1	1
13	6.33	2.33	1	1
14	7	2.33	1	1
15	8.66	2.66	1.33	1
16	11	3	1.33	1
17	13.33	4.66	2.33	1.33
18	16	5	2.33	1.33
19	19	6.66	2.66	2
20	22	7	3	2.33

Tabla G4: Caracterización del LED ámbar derecho, se muestra la intensidad relativa con su respectiva intensidad en lúmenes a 1, 2, 3 y 4 centímetros de la zona donde incide la luz homogéneamente.

Intensidad relativa	Intensidad [Lm] a 1 cm	Intensidad [Lm] a 2 cm	Intensidad [Lm] a 3 cm	Intensidad [Lm] a 4 cm
1	0.66	0.666	0.66	0.66
2	0.66	0.666	0.66	0.66
3	0.66	0.666	0.66	0.66
4	0.66	0.666	0.66	0.66
5	0.66	0.666	0.66	0.66
6	0.66	0.666	0.66	0.66
7	0.66	0.666	0.66	0.66
8	0.66	0.666	0.66	0.66
9	0.66	0.666	0.66	0.66
10	1	0.666	0.66	0.66
11	1	0.666	0.66	0.66
12	1	1	0.66	0.66
13	1.33	1	0.66	0.66
14	1.33	1	0.66	0.66
15	2.33	1	0.66	0.66
16	2.66	1.33	0.66	0.66
17	3	2	1	0.66
18	4.66	2.66	1	0.66
19	6.66	3	1.33	1
20	9	4.66	1.33	1

Tabla G5: Caracterización del LED verde izquierdo, se muestra la intensidad relativa con su respectiva intensidad en lúmenes a 1, 2, 3 y 4 centímetros de la zona donde incide la luz homogéneamente.

Intensidad relativa	Intensidad [Lm] a 1 cm	Intensidad [Lm] a 2 cm	Intensidad [Lm] a 3 cm	Intensidad [Lm] a 4 cm
1	0.66	0.66	0.66	0.66
2	0.66	0.66	0.66	0.66
3	0.66	0.66	0.66	0.66
4	0.66	0.66	0.66	0.66
5	0.66	0.66	0.66	0.66
6	0.66	0.66	0.66	0.66
7	0.66	0.66	0.66	0.66
8	0.66	0.66	0.66	0.66
9	1	0.66	0.66	0.66
10	1	0.66	0.66	0.66
11	1	0.66	0.66	0.66
12	1.33	1	0.66	0.66
13	1.33	1	0.66	0.66
14	2.33	1	0.66	0.66
15	2.66	1.33	1	0.66
16	3	1.33	1	0.66
17	4.66	2.33	1	1
18	6.66	3	1.33	1
19	10.66	4.66	2.33	1.33
20	16	7	3	2.33

Tabla G6: Caracterización del LED verde derecho, se muestra la intensidad relativa con su respectiva intensidad en lúmenes a 1, 2, 3 y 4 centímetros de la zona donde incide la luz homogéneamente.

Intensidad relativa	Intensidad [Lm] a 1 cm	Intensidad [Lm] a 2 cm	Intensidad [Lm] a 3 cm	Intensidad [Lm] a 4 cm
1	0.66	0.66	0.66	0.66
2	0.66	0.66	0.66	0.66
3	0.66	0.66	0.66	0.66
4	0.66	0.66	0.66	0.66
5	0.66	0.66	0.66	0.66
6	0.66	0.66	0.66	0.66
7	0.66	0.66	0.66	0.66
8	0.66	0.66	0.66	0.66
9	0.66	0.66	0.66	0.66
10	0.66	0.66	0.66	0.66
11	1	0.66	0.66	0.66
12	2.33	1	0.66	0.66
13	3	2.33	1	0.66
14	6.33	3	1.33	1
15	8.66	5	2.33	1.33
16	13	7	3	2.33
17	16.33	10.66	4.66	2.66
18	22	13.33	6.33	4.33
19	30.33	16.33	7	4.66
20	34.33	21.33	8.66	6.33

Tabla G7: Caracterización del LED azul izquierdo, se muestra la intensidad relativa con su respectiva intensidad en lúmenes a 1, 2, 3 y 4 centímetros de la zona donde incide la luz homogéneamente.

Intensidad relativa	Intensidad [Lm] a 1 cm	Intensidad [Lm] a 2 cm	Intensidad [Lm] a 3 cm	Intensidad [Lm] a 4 cm
1	0.66	0.66	0.66	0.66
2	0.66	0.66	0.66	0.66
3	0.66	0.66	0.66	0.66
4	0.66	0.66	0.66	0.66
5	0.66	0.66	0.66	0.66
6	0.66	0.66	0.66	0.66
7	0.66	0.66	0.66	0.66
8	0.66	0.66	0.66	0.66
9	0.66	0.66	0.66	0.66
10	0.66	0.66	0.66	0.66
11	1	0.66	0.66	0.66
12	1.33	1	0.66	0.66
13	2.66	2.33	1	0.66
14	4.66	3	1.33	1
15	6.66	4.66	2.66	1.33
16	8.66	6.66	3	2.33
17	11.33	8.66	4.66	2.66
18	13.66	11	6.33	4.33
19	18.66	13.66	8.33	5
20	22	16	9	6.33

Tabla G8: Caracterización del LED azul derecho, se muestra la intensidad relativa con su respectiva intensidad en lúmenes a 1, 2, 3 y 4 centímetros de la zona donde incide la luz homogéneamente.

Intensidad relativa	Intensidad [Lm] a 1 cm	Intensidad [Lm] a 2 cm	Intensidad [Lm] a 3 cm	Intensidad [Lm] a 4 cm
1	0.66	0.66	0.66	0.66
2	0.66	0.66	0.66	0.66
3	0.66	0.66	0.66	0.66
4	0.66	0.66	0.66	0.66
5	0.66	0.66	0.66	0.66
6	0.66	0.66	0.66	0.66
7	0.66	0.66	0.66	0.66
8	0.66	0.66	0.66	0.66
9	0.66	0.66	0.66	0.66
10	0.66	0.66	0.66	0.66
11	1	0.66	0.66	0.66
12	3	2.33	1	0.66
13	8.33	4.66	2.33	1
14	13.66	8.33	4.33	2.33
15	22	11.33	6.33	2.66
16	31	16.33	8.33	4.33
17	41.33	22	10.66	6.33
18	53	30.33	13.33	7
19	70	37.66	18.33	9
20	85.6	48.33	22	11.33

Tabla G9: Caracterización del LED blanco izquierdo, se muestra la intensidad relativa con su respectiva intensidad en lúmenes a 1, 2, 3 y 4 centímetros de la zona donde incide la luz homogéneamente.

Intensidad relativa	Intensidad [Lm] a 1 cm	Intensidad [Lm] a 2 cm	Intensidad [Lm] a 3 cm	Intensidad [Lm] a 4 cm
1	0.66	0.66	0.66	0.66
2	0.66	0.66	0.66	0.66
3	0.66	0.66	0.66	0.66
4	0.66	0.66	0.66	0.66
5	0.66	0.66	0.66	0.66
6	0.66	0.66	0.66	0.66
7	0.66	0.66	0.66	0.66
8	0.66	0.66	0.66	0.66
9	0.66	0.66	0.66	0.66
10	0.66	0.66	0.66	0.66
11	0.66	0.66	0.66	0.66
12	1.33	1	0.66	0.66
13	5	2.66	1.33	1
14	11.33	5	2.66	2.33
15	19	9	5	3
16	30.66	13.66	8.33	5
17	44.66	21.33	11.33	8.33
18	57.33	28	15.66	10.66
19	75	37.66	21.33	13.33
20	96.33	49	27.66	18.66

Tabla G10: Caracterización del LED blanco derecho, se muestra la intensidad relativa con su respectiva intensidad en lúmenes a 1, 2, 3 y 4 centímetros de la zona donde incide la luz homogéneamente.

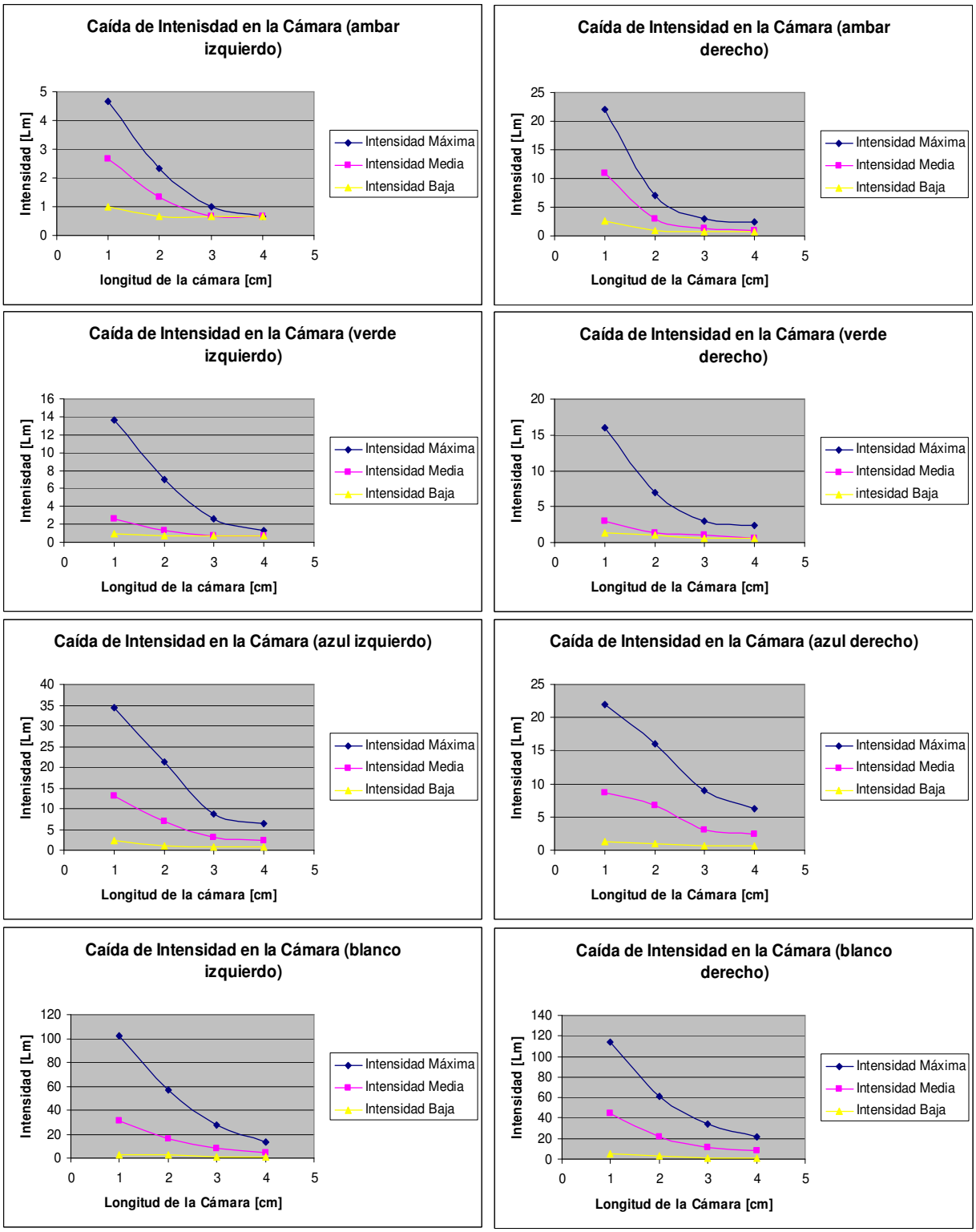


Fig. G1: Gráficas de la caída de intensidad luminosa a lo largo de la cámara para las diferentes fuentes de luz

Apéndice H

Resultados.

Gráficas Estado Basal

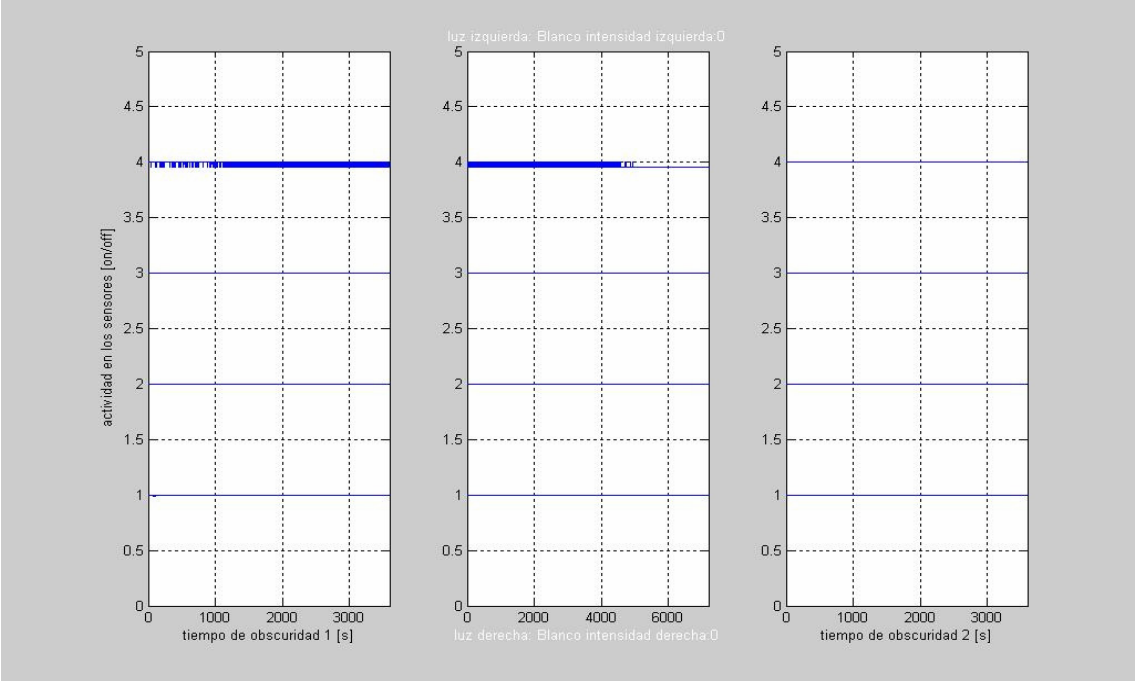


Fig. H1: Estado basal de la cámara de registro en 4hrs de experimento (1hr de obscuridad inicial, 2hrs de experimento y 1hr de obscuridad final) sin mosca en el interior.

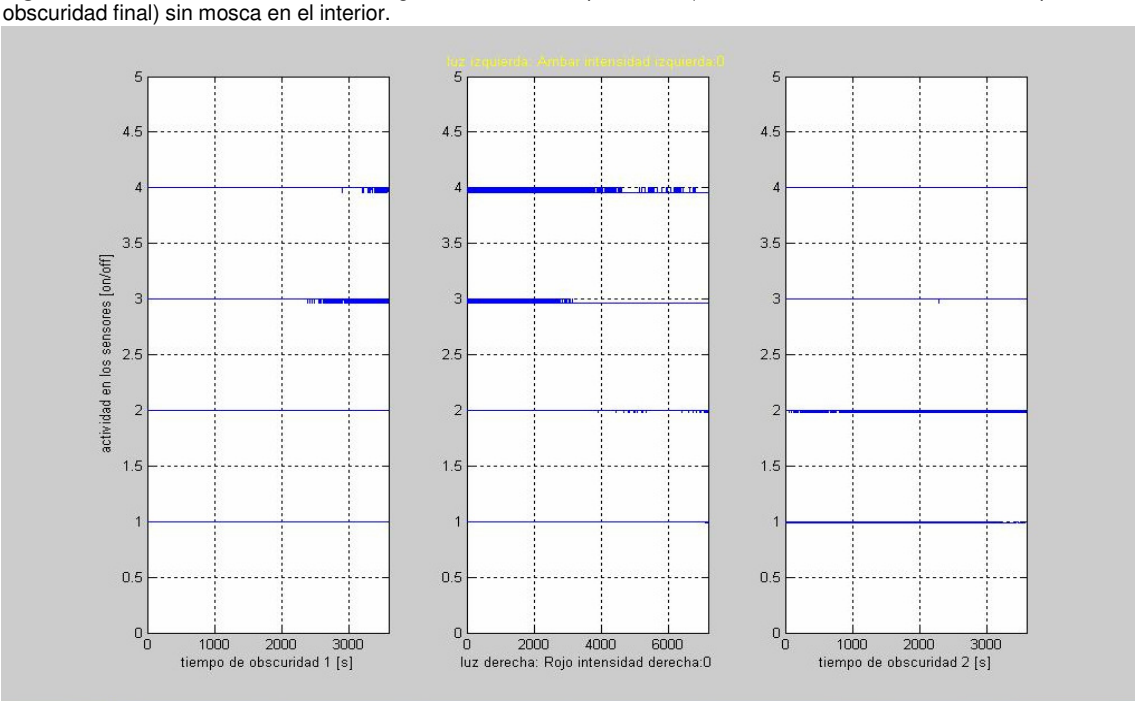


Fig. H2: Estado basal de la cámara de registro en 4hrs de experimento (1hr de obscuridad inicial, 2hrs de experimento y 1hr de obscuridad final) sin mosca en el interior.

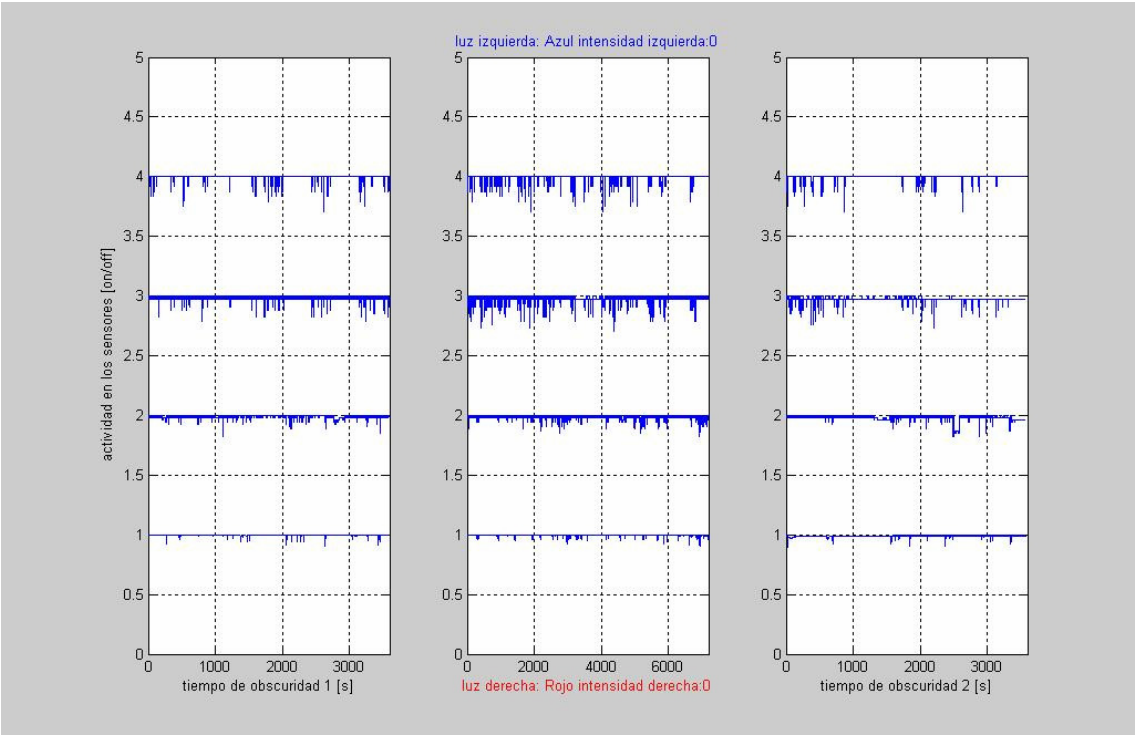


Fig. H3: Estado basal de la cámara de registro en 4hrs de experimento (1hr de obscuridad inicial, 2hrs de experimento y 1hr de obscuridad final) con una mosca en el interior.

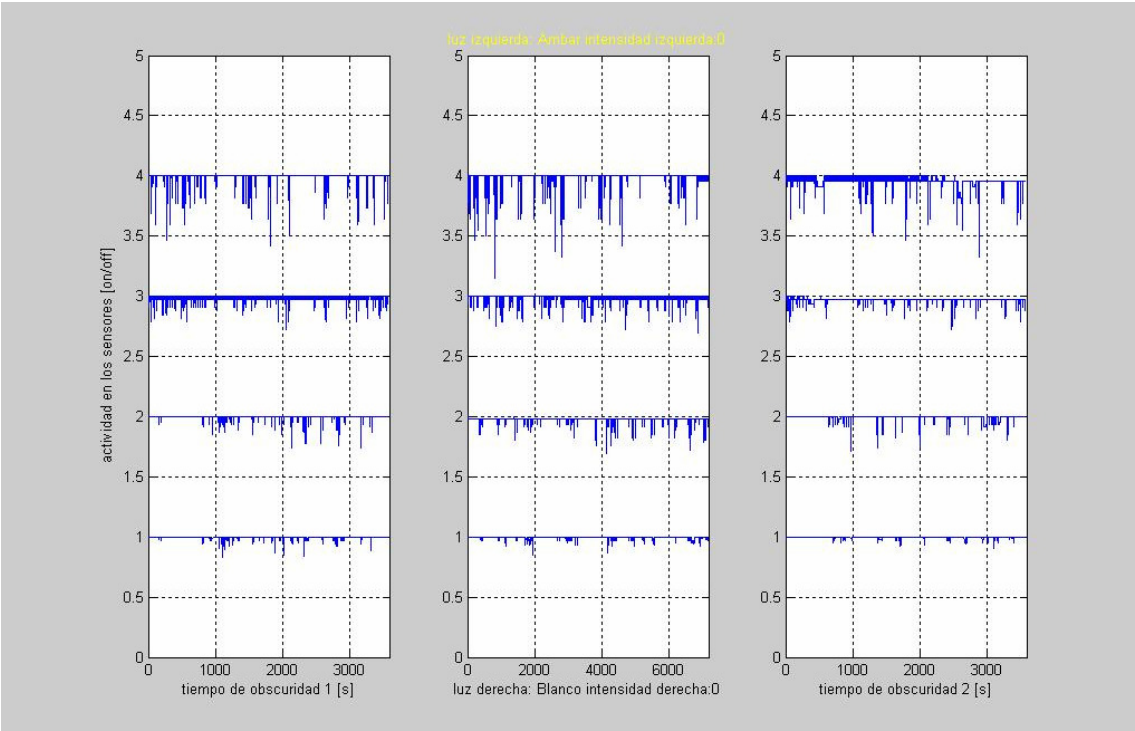


Fig. H4: Estado basal de la cámara de registro en 4hrs de experimento (1hr de obscuridad inicial, 2hrs de experimento y 1hr de obscuridad final) con una mosca en el interior.

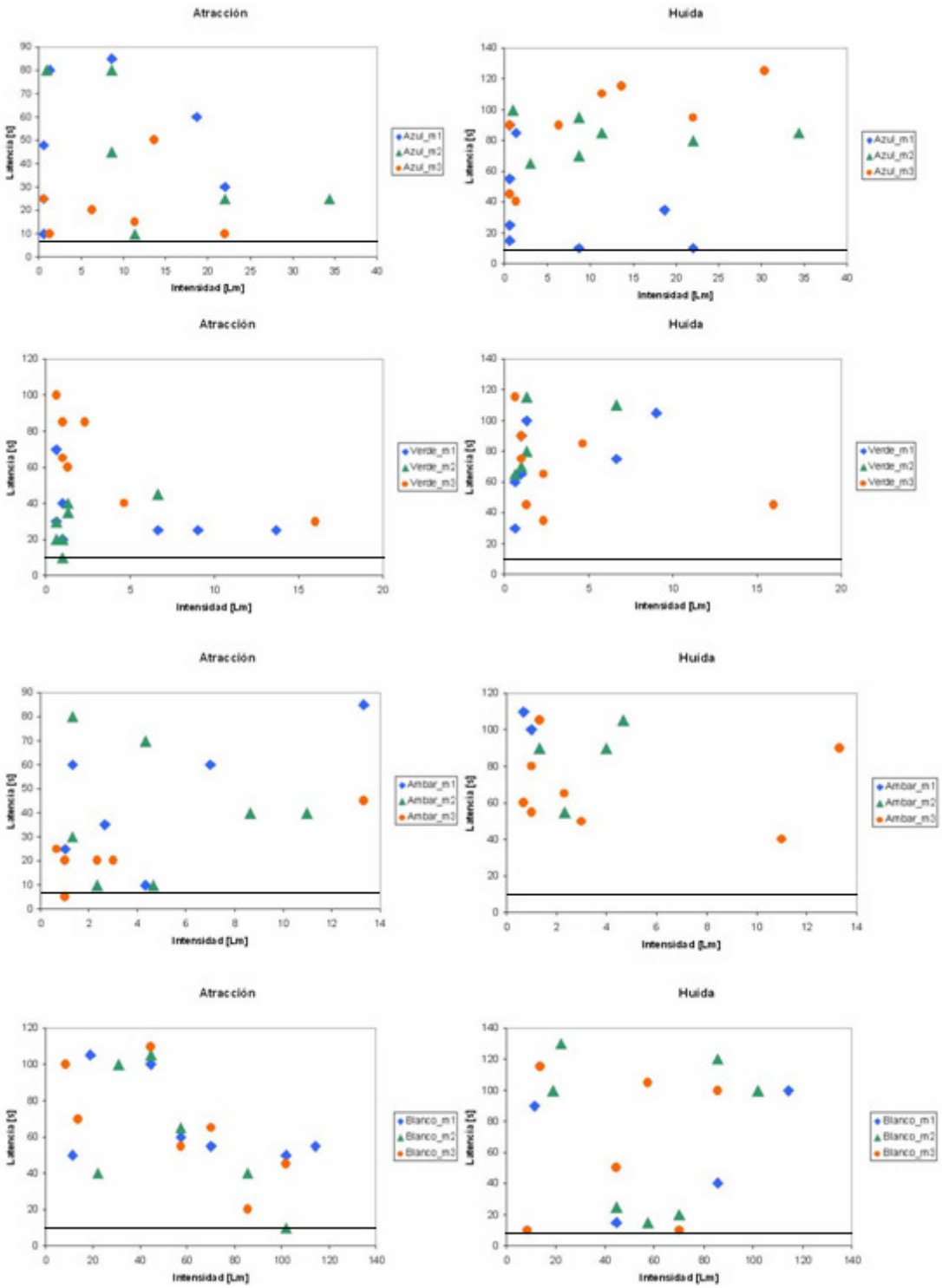


Fig. H5: Gráficas de atracción y huída en función de la intensidad para los colores: azul, verde, ámbar, rojo y blanco. Se debe recordar que el valor promedio de respuesta entre sensores (a 2 cm de distancia entre ellos) es: 8.256 ± 1.34 segundos (línea recta paralela al eje x en las gráficas de atracción e huída en función de la intensidad).

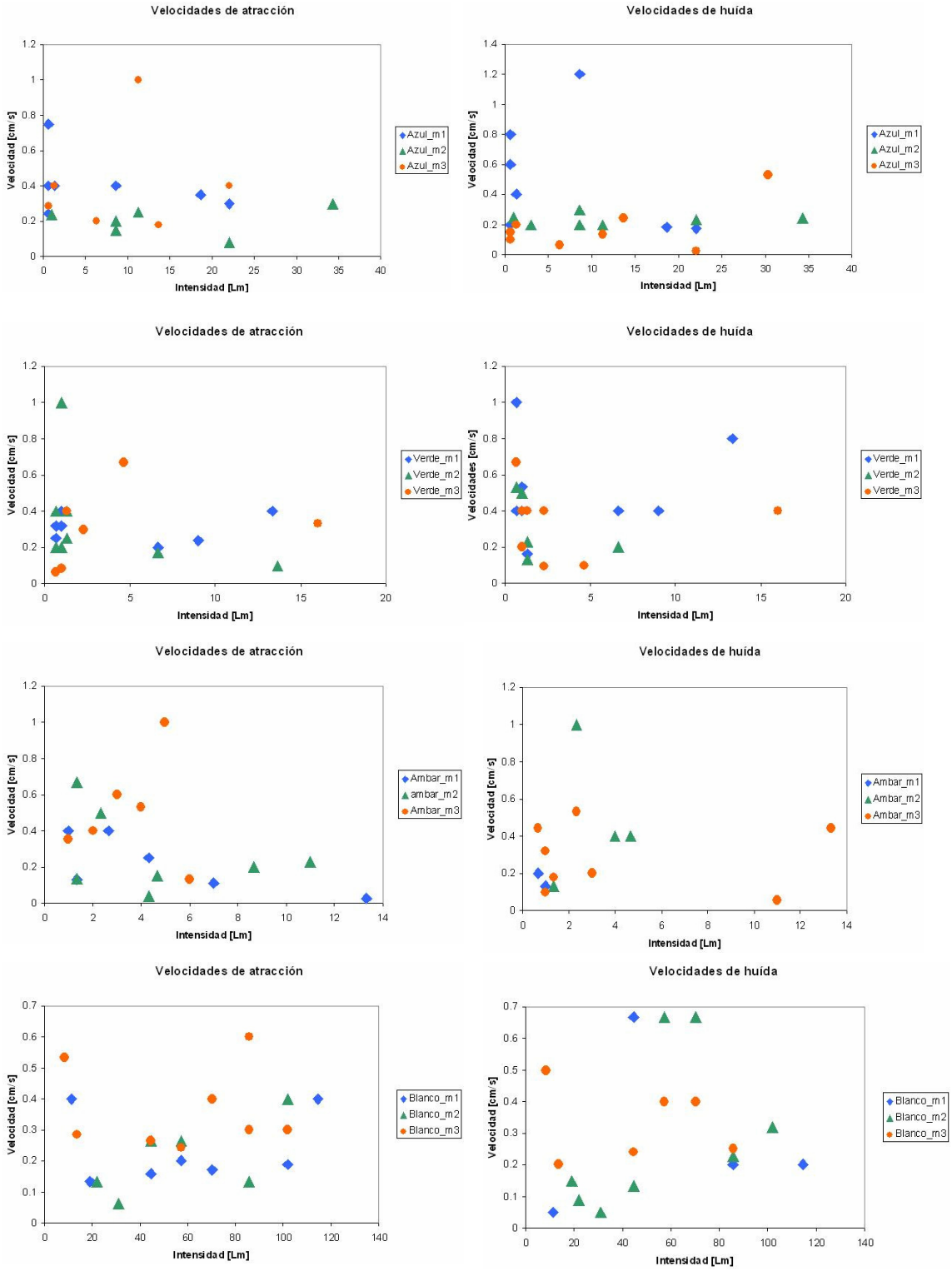


Fig. H6: Gráficas de velocidad de atracción y velocidad de huída en función de la intensidad para los colores: azul, verde, ámbar, rojo y blanco. Se debe recordar que el valor promedio de respuesta entre sensores (a 2 cm de distancia entre ellos) es: 8.256 ± 1.34 segundos.

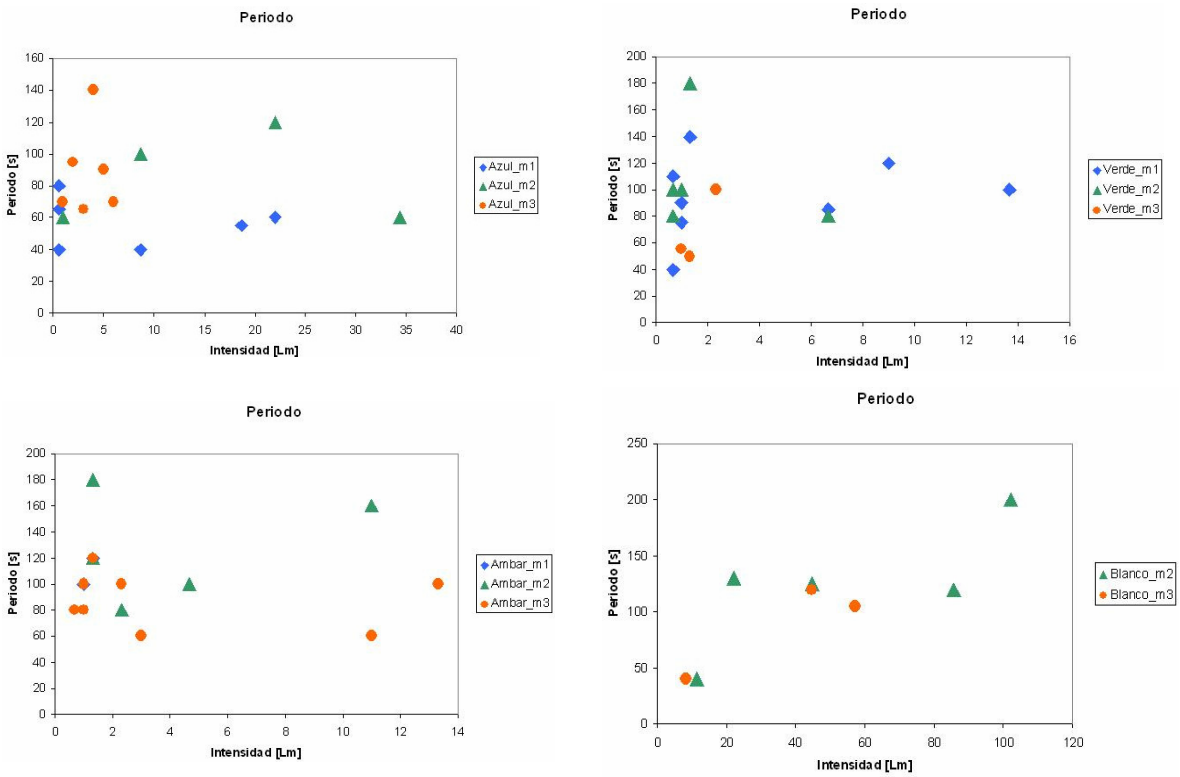


Fig. H7: Gráficas de periodo en función de la intensidad para los colores: azul, verde, ámbar, rojo y blanco.

Apéndice I

Cotización por Dispositivo

Cantidad	Nombre	Costo x unidad	Costo total
1	Microcontrolador	$303.00	$303.00
2	Potenciometro digital	$35.00	$70.00
1	Cristal de cuarzo 8MHz	$13.00	$13.00
1	Regulador de voltaje	$7.00	$7.00
1	Micro switch push	$3.00	$3.00
2	Jacks 3.5 mm stereo	$5.00	$10.00
1	Plug 3.5mm stereo	$4.00	$4.00
1	Terminales con 2 tornillos	$6.00	$6.00
2	Capacitores 470uF	$7.00	$14.00
2	Capacitores 0.1uF	$3.00	$6.00
2	Capacitores 15pF	$2.00	$4.00
4	Resistencia 330 ohms	$0.50	$2.00
6	Resistencia 1kohms	$0.50	$3.00
1	Resitencia 4.7kohms	$0.50	$0.50
12	Resistencia 10kohms	$0.50	$6.00
2	Resistencia 22kohms	$0.50	$1.00
2	Resistencia 100kohms	$0.50	$1.00
2	Resistencia 68 ohms	$0.50	$1.00
2	Resistencia 270 ohms	$0.50	$1.00
2	Resistencia 2.2kohms	$0.50	$1.00
4	Resistencia 100 ohms	$0.50	$2.00
2	Resistencia 680 ohms	$0.50	$1.00
2	Resistencia 560 ohms	$0.50	$1.00
10	Transistores bc547	$3.00	$30.00
2	LED azul superbrillante 5mm	$6.00	$12.00
2	LED verde superbrillante 5mm	$6.00	$12.00
2	LED ambar superbrillante 5mm	$6.00	$12.00
2	LED rojo superbrillante 5mm	$6.00	$12.00
2	LED blanco superbrillante 5mm	$10.00	$20.00
4	LED emisor de IR 5mm	$5.00	$20.00
4	Phototransistor 5mm	$6.00	$24.00
1	Base para c.i 40 patas	$4.00	$4.00
2	Base para c.i 8 patas	$2.00	$4.00
1	Broche porta-pilas 9V	$4.00	$4.00
1	Conector DB9-Hembra	$10.00	$10.00
2	Metro cable para microfono stereo	$15.00	$30.00
2	Lentes	$300.00	$600.00
1	Fenólica 15x20cm	$31.00	$31.00
4	Metro de cable bicolor calibre 22	$6.00	$24.00
1	Gabinete de plastico 196x110x36 cm	$140.00	$140.00
		Total	$1,449.50

*Nota: La manufactura de la cámara se considera aparte y dependerá del fabricante.

Bibliografía

Bibliografía La Mosca *Drosophila*

[1] Bate M. and Martínez A. *The development of Drosophila Melanogaster*, Cold Spring Harbor Laboratoy Press, 1993, Vol 3, 22(1,2) p1279-1281
[2] Freeman Matthew (2005) Eye development: stable cell fate decision in insects colour vision. *Current Biology* 15, 924-926.
[3] Ramos Patricia y colaboradores. *Manual de laboratorio de Genética para Drosophila Melanogaster*,MacGraw-Hill México 1993, 1, p3
[4] Roberts, D. B and Standen G.N. *Drosophila: a practical approach*, IRL Press, 2ª edición, 1998, 1(2), p1, 9 (1) p 265-266, 9 (6) p 284, 9 (8) p306-309.
[5] Zucker Charles S. (1996). The biology of vision in Drosophila. *PNAS* 93, 571-576.

Bibliografía Fuente de Luz Fría (LED's)

[6] Boylestad and Nashelsky, *Electrónica: Teoría de circuitos y dispositivos electrónicos*. Pearson Education, 8ª edición, México 2003, 1(16), p 40-41.
[7] Malvino Albert Paul *Principios de electrónica* MacGraw-Hill, 3ª edición 1991, 4(3), p 115-116.

Otras Fuentes de Consulta

[8] Manuales 1, 2 y 3 del microcontrolador PICAXE, disponible en www.picaxe.com.