UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

"PERFECCIONAMIENTO DEL DISPOSITIVO PARA LA CUANTIFICACIÓN DEL COMPORTAMIENTO DE LA MOSCA *DROSOPHILA* EVOCADO POR LUZ"

TESIS

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO ELÉCTRICO-ELECTRÓNICO

PRESENTA:

IGNACIO MEDINA RAMOS

ASESOR: ING. GIOVANNI FONSECA FONSECA

MEXICO, D.F, 2009

Agradecimientos y dedicatorias

- Quiero agradecer y dedicar esta tesis a mis padres, Ignacio Medina Bellmunt y Mercedes Ramos de Medina, por todo el apoyo que he tenido desde que empecé a pensar.
- Agradecer a mis hermanos María, Montserrat y Jaime por estar siempre ahí.
- A mis abuelos Fernando Ramos y María González de Castilla† por darme un hogar por 4 años y por su apoyo incondicional.
- A Giovanni por todo lo que me ha hecho pensar y mejorar intelectualmente, que me invitó a participar en el proyecto y sin él, esta tesis no hubiera sido posible.
- A Juan Manuel Gómez González por todo lo que aportó al proyecto y facilitarme mucho el avance de éste.
- Al CECiB y a todos los integrantes de Experimenta por las instalaciones, el equipo y que hicieron muy ameno el proyecto.

Índice:

1. Introducción

En el Centro de Ciencias para el Bachillerato (CECiB) en la Facultad de Ciencias se imparten talleres de ciencia a jóvenes de bachillerato, principalmente del CCH. Los talleres son de Matemáticas, Física, Química, Biología y Tecnología. En el taller de Tecnología el objetivo es hacer que los estudiantes diseñen y desarrollen un dispositivo en el que aprovechen algún tipo de energía natural. Sin embargo el área de tecnología, además de impartir el taller, tiene la función de diseñar y desarrollar dispositivos que sirvan como material didáctico en las sesiones de los demás talleres.

En el taller de Biología estudian el efecto de la luz, con diferentes intensidades y colores en los seres vivos. Los alumnos estudian principalmente el comportamiento en las moscas *Drosophila*, pero no contaban con un dispositivo que pudiese cuantificar ese comportamiento. Durante el año dos mil seis, en el centro se desarrolló el dispositivo; sin embargo tenía algunas restricciones por lo que en el presente trabajo se plantean mejoras a éste.

Las mejoras realizadas sirven para poder obtener resultados más precisos de los experimentos que se realicen y para tener una gama más amplia de experimentos posibles a realizar. En el dispositivo original, se utilizan cinco fuentes de luz de diferentes colores y veinte diferentes intensidades para cada color sin la posibilidad de poder mezclarse. El nuevo dispositivo tiene la posibilidad de mezclar dos diferentes longitudes de onda, cada una con intensidad propia, para poder tener una gama de colores mucho mayor a la del dispositivo original. El dispositivo contaba con cuatro sensores que monitoreaban la posición de la mosca, al nuevo dispositivo se le implementaron otros cuatro más, lo que permite obtener resultados más precisos. También se implementó una alarma sonora y visual de encendido para poder saber si está fallando un sensor o si existe una fuga de luz al interior del dispositivo.

2. Marco teórico

2.2. Mosca *Drosophila*

La mosca *Drosophila*, también llamada mosca de vinagre o de la fruta ya que se alimenta principalmente de frutas en proceso de fermentación, es un insecto díptero, o sea de dos alas, en el cual se agrupan los organismos donde el primer par de alas es funcional y el segundo par de alas fue transformado en órganos de equilibrio llamados halterios o balancines. **[1, 7]**

Clasificación científica

Reino:	Animalia
Filo:	Arthropoda
Clase:	Insecta
Orden:	Diptera
Suborden:	Brachycera
Familia:	Drosophilidae
Subfamilia:	Drosophilinae
Género:	*Drosophila*
Subgénero:	*Sophophora*
Complejo específico:	*melanogaster* complex
Especie:	***Drosophila melanogaster***

Tabla 1: Características de la mosca *Drosophila* melanogaster. **[1]** Imagen 1: Fotografía con acercamiento de una mosca *Drosophila*. **[1]**

A principios del siglo XX fue adoptada como animal de experimentación genética por Thomas Morgan. Esta moca se utiliza generalmente en el estudio genético ya que tiene solamente 4 pares de cromosomas y su ciclo de vida es muy breve, entre 10 y 21 días, por lo que se pueden estudiar varias generaciones en poco tiempo conociéndose el mapa completo de su genoma **[1]**. Además el 61% de los genes de las enfermedades conocidas en el hombre se pueden identificar en el genoma de estas moscas, y el 50% de las secuencias proteínicas de la mosca tiene análogos con las de los mamíferos, por lo que para propósitos de investigación el estudiar a la mosca *Drosophila* permite conocer de manera muy aproximada al ser humano **[1]**. También el tamaño de la mosca es una ventaja, ya que miden entre 1 y 1.5 milímetros de longitud y son fáciles de manipular. Además, se cuenta con un número enorme de mutantes.

Además de que la mosca *Drosophila* es utilizada en el estudio de la genética, también se utiliza para el estudio de su comportamiento, ya que responde a estímulos como el olor, la luz, el sabor, la humedad, la temperatura y la aceleración de la gravedad, entre otras. De estos estímulos, nos interesa su alta sensibilidad a los colores, respondiendo a estos en un rango de longitudes de onda entre 350 nm y 650 nm, que son longitudes de onda comprendidas entre la zona ultravioleta y el color rojo **[6, 8]**. La importancia de estudiar el comportamiento optomotor de la mosca radica en poder relacionar la función de cada uno de los genes con la percepción de la luz.

Existen dispositivos para el estudio del comportamiento de la mosca *Drosophila*. El más utilizado y el que ha funcionado de manera más útil es el laberinto en forma de "T". En este laberinto se introduce a la mosca por un orificio central y se le estimula en los extremos de la "T". La mosca elegirá irse a alguno de los brazos de la "T", por lo que existe la misma probabilidad de que la mosca se vaya a cualquiera de los extremos de la "T" sin que exista estímulo, y las probabilidades de error son menores cuando existe un estímulo. Este tipo de cámara es el que se eligió para el dispositivo final.

2.3. Microcontroladores

Un microcontrolador es un circuito integrado de alta escala de integración, el cual permite manipular y controlar entradas para la obtención de salidas deseadas mediante todo un proceso. Un microcontrolador se compone de varios componentes: microprocesador; memoria RAM (Random Access Memory) para contener los datos; memoria para el programa tipo ROM (Read-Only Memory) PROM (Programable Read-Only Memory), EPROM (Erasable Programable Read-Only Memory) o EEPROM/E^2PROM (Electrically Erasable Programable Read-Only Memory) puertos de entrada y de salida para comunicarse con el exterior; diferentes módulos para el control de periféricos como pueden ser convertidores Analógico-Digital, puertos serie, etc. y generador de pulsos de reloj para sincronizar el funcionamiento del sistema.

El microprocesador es el "cerebro" del microcontrolador el cual se encarga de realizar todas las operaciones aritméticas y logísticas necesarias para que el microcontrolador realice sus funciones. La diferencia entre microprocesadores y microcontroladores es que los microprocesadores contienen la Unidad Central de Proceso y solamente realizan los procesos de información, mientras que el microcontrolador aprovecha los procesos de información para controlar puertos de entradas y salidas **[3]**.

La memoria RAM es una memoria volátil, la cual almacena datos y al perder energía (alimentación eléctrica) pierde la información. El microcontrolador tiene para registrar temporalmente la información. Entre mayor sea la capacidad de memoria RAM, más rápido será el dispositivo ya que facilita al microprocesador el manejo de la información.

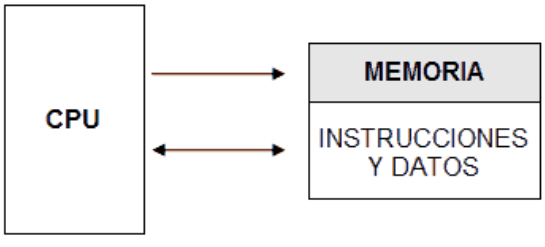
La memoria ROM, PROM, EPROM y E^2PROM son memorias no volátiles, las cuales, aún después de perder energía (alimentación eléctrica), los datos permanecen almacenados. En esta memoria se guarda el programa que se le introduce al microcontrolador para que realice el proceso deseado.

Los puertos de entrada y de salida son con los que se van a realizar las funciones programadas en el microcontrolador. En los puertos de entrada se conectan dispositivos que permiten obtener información del exterior (botones, sensores, interruptores, teclados, etc.) y en los puertos de salida se obtiene la señal requerida para que los actuadores conectados a ellas hagan las tereas programadas observando éstas en monitores, alarmas, LED's, osciloscopios, etc. Tanto las entradas como las salidas digitales, tendrán valores binarios y las entradas analógicas para poder procesarlas deberán ser transformadas a valores digitales por un convertidor analógico-digital que contenga el microcontrolador.

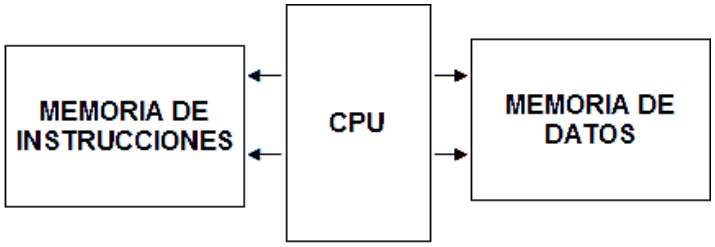
Dependiendo de las necesidades de aplicación, algunos microcontroladores contienen diversos y diferentes módulos para el control de periféricos. Éstos sirven para facilitar la implantación de las diferentes aplicaciones al no tener que colocarlos externamente. Estos módulos pueden ser convertidores analógico-digital, convertidores digital-analógico, puertos serie, termopares, etc.

Existen dos tipos de arquitecturas de los microcontroladores, la arquitectura Von Neumann y la arquitectura de Harvard. Inicialmente todos los microcontroladores estaban hechos mediante la arquitectura Von Neumann, pero en la actualidad es más utilizada la arquitectura Harvard.

La arquitectura Von Neumann se caracteriza por disponer solamente de una memoria principal donde se almacenan instrucciones y datos de forma indistinta. Para acceder a esta memoria existe un sistema de buses único.



La arquitectura Harvard contiene una memoria para datos y otra memoria para instrucciones, las cuales son totalmente independientes permitiendo que tengan accesos simultáneos.



Existen los microcontroladores tipo RISC (*Reduced Instruction Set Computer*) de set de instrucciones reducido y los tipo CISC (*Complex Instruction Set Computing*) de set de instrucciones complejo.

Los microcontroladores RISC tienen instrucciones de tamaño fijo, presentadas en un reducido número de formatos y sólo las instrucciones de carga y almacenamiento acceden a la memoria por datos.

Los microcontroladores CISC, en contraposición de los RISC, tienen un conjunto de instrucciones que se caracteriza por ser muy amplio y permitir operaciones complejas entre operandos situados en la memoria o en los registros internos.

3. Desarrollo

3.1. Análisis del problema

3.1.1. Diferencias tecnológicas con el dispositivo previamente realizado

El dispositivo original fue realizado con un microcontrolador tipo PICAXE 40X el cual tiene seiscientos registros de memoria y se debe poner un reloj o generador de pulsos externo, ya sea de 4 MHz o de 8 MHz para su funcionamiento. También tiene un número muy limitado de las variables que se pueden utilizar en su programación. En el momento del diseño y creación del dispositivo, ese microcontrolador era el más poderoso de la familia, por lo cual se diseñó pensando en las limitantes que tenía el microcontrolador.

La nueva generación de PICAXE son los X-1 y X-2 los cuales tienen mejoras que se pudieron aprovechar para el diseño del nuevo dispositivo. La primera mejora es el aumento en la capacidad de memoria; por ejemplo, tiene 1000 registros de memoria, además de tener osciladores internos con los que con un solo comando en su programación se puede tener el microcontrolador trabajando a cuatro u ocho MHz. Por último el número de variables para la programación incrementó considerablemente, de 14 variables para la programación pasó a 28, facilitando mucho su programación.

3.1.2. Propuesta de mejoras al dispositivo

El cambio principal y más complejo debe ser poder combinar colores de LED's de cada lado; es decir, hacer que la gama de colores posibles sea mayor. Otro cambio debe ser la posibilidad de poder escoger diferentes duraciones para los experimentos. En el dispositivo previo sólo era posible escoger un tiempo de obscuridad, el cual era el mismo tiempo de obscuridad inicial y final. El tercer cambio significativo será el aumento de sensores en la cámara de registro. El dispositivo cuenta con cuatro sensores para mantener simplicidad en el dispositivo, sin embargo la cámara cuenta con otros cuatro pares de orificios lo que permitirán dar mayor precisión al hacer el experimento; es decir, si la mosca decidía pasar por un sensor, pero regresar antes de llegar al otro, no se podía saber con exactitud hasta dónde había llegado. Ahora, con más sensores, la separación entre éstos disminuye y con ello también lo hace el margen de error. Por último se le agregará un sistema de alarma de fallo, dado que había veces que se hacían pruebas largas y en cualquier momento del experimento dejaba de funcionar un sensor, lo que no se podía saber sino hasta que se terminaba el experimento. Ahora, en el momento que deje de funcionar un sensor, el sistema avisará con un LED rojo y con una alarma auditiva. También se colocarán foto resistores en los extremos de la cámara de registro que permitirán saber si hay luz en el interior y su intensidad aproximada.

Dispositivo anterior	Dispositivo nuevo
Mismo tiempo de obscuridad inicial y obscuridad final	Diferentes tiempos de obscuridad inicial y obscuridad final
4 sensores infrarrojos	8 sensores infrarrojos
Posibilidad de encender 1 LED de cada lado con 20 diferentes intensidades relativas e independientes	Posibilidad de encender 2 LED's de cada lado con 30 diferentes intensidades relativas e independientes
Sin alarma de fallo	Posee alarma de fallo
Sin contador de paso por los sensores	Con contador para el paso de la mosca en cada sensor.

Tabla 2: Diferencias del dispositivo anterior con el dispositivo nuevo

3.1.3. Dispositivo previamente realizado

El dispositivo original consiste en una cámara de registro, un microcontrolador, dos potenciómetros digitales, cuatro emisores de rayos infrarrojos, cuatro receptores de infrarrojo y LED's de cinco colores distintos: amarillo, azul, blanco, rojo y verde.

La cámara de registro es un tubo hueco de 5 [mm] de diámetro y 11 [cm] de largo el cual tiene en los extremos unas superficies semiesféricas en las cuales se encuentran los cinco LED's de colores a distancias iguales, éstos convergen en una lente convexa para poder homogeneizar la incidencia de la luz en el tubo hueco. A lo largo del tubo hueco hay 8 pares de orificios para poner los emisores y receptores de infrarrojos, separados un centímetro cada par.

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

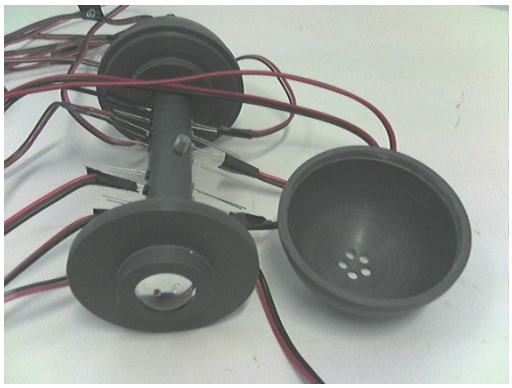


Figura 2: Cámara de registro destapada en la cual se pueden observar las lentes que homogenizan la luz. **[2]**

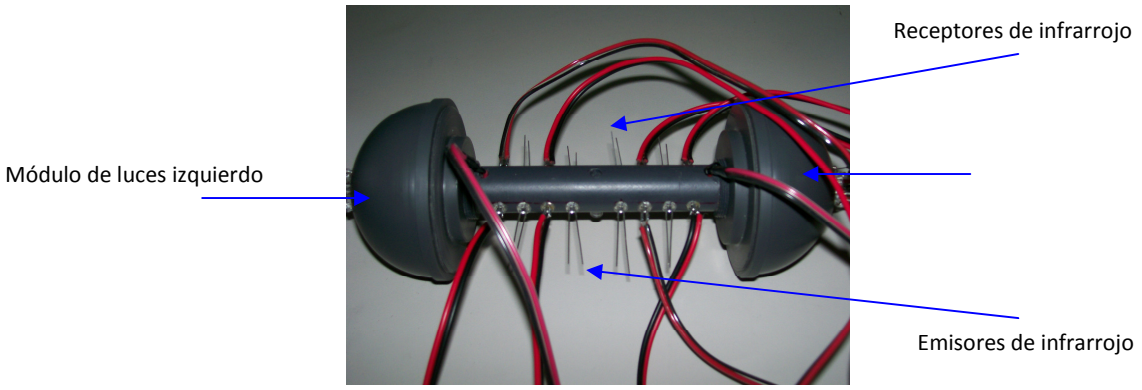


Figura 3: Cámara de registro en la cual se observan los módulos de luces, así como los emisores y receptores de rayos infrarrojos. **[2]**

El microcontrolador enciende el color del LED de cada lado a la intensidad deseada, revisa el estado de los sensores de infrarrojos, tanto en el tiempo de obscuridad de inicio y terminación del experimento (el cual es el mismo) como en el tiempo de duración del experimento.

3.2.Desarrollo del hardware

3.2.1. Dispositivos utilizados

Para realizar el dispositivo se utilizaron tres microcontroladores tipo PICAXE: 40X-1, 28X-1 y 18A. Estos microcontroladores funcionan con cinco volts; entonces, para poder utilizar pilas de 9 [V], se puso un convertidor de 9 [V] a 5 [V], para que todo el sistema funcione con 5 volts.

Para el monitoreo de la mosca se pusieron ocho sensores infrarrojos; esto es, ocho emisores y receptores de infrarrojo. El flujo de emisor a receptor es constante y aparece en una gráfica como una línea recta, cuando se interrumpe, es cuando aparecen las perturbaciones en la línea. Es decir, cuando pasa la mosca entre emisor y receptor, aparece una perturbación en la gráfica y así es como se puede saber qué sensor detectó a la mosca y en el tiempo en que lo hizo.

Para la intensidad de luz se utilizaron cuatro potenciómetros digitales de 10 [kΩ] tipo X9C103P. Estos potenciómetros digitales son como los analógicos, pero para aumentar o disminuir su valor, se hace por medio de pulsos eléctricos. Tiene una entrada para introducir los pulsos y otra para subir o bajar la resistencia.

El valor del potenciómetro se sensa con un ADC del microcontrolador para saber su valor, el convertidor es de 8 bits, así que el valor que sensaba se encontraba entre 0 y 255, siendo 255, 10 [kΩ] del potenciómetro. Para determinar el valor del potenciómetro para encender los LED's al mínimo de encendido, se midieron valores con un detector de luz y se observó a simple vista cuando el LED se encontrara apagado, con estos datos se comparó qué valor era menor y se escogió éste. El valor de encendido máximo, se escogió como el valor máximo que da el potenciómetro. Cuando se requiere la intensidad cero, entonces el programa posiciona al potenciómetro a un valor anterior al valor mínimo. Cuando se quiere la máxima intensidad, se

introduce el valor de la intensidad máxima y el programa ajusta al potenciómetro a un valor anterior al valor mínimo y de ahí cuenta el número de pulsos del valor de la intensidad máxima y así se obtiene la mayor intensidad posible.

Se utilizaron cinco LED's ultra brillantes de cada lado de la cámara. Los colores son: azul, rojo, verde, ámbar y blanco. Estos LED's son más brillantes que los LED's normales y funcionan con el mismo valor de corriente que es de 20 [mA].

3.2.1.1. PIC

Existen microcontroladores llamados PIC cuya arquitectura es Harvard y su set de instrucciones es tipo RISC, el cual permite realizar cada instrucción en un ciclo de reloj. Existen diferentes tipos de PIC's, donde varía principalmente el número de entradas y salidas, si tienen convertidores analógico-digital, la frecuencia de oscilación que soportan y el consumo de corriente, entre otras.

Para la programación de un PIC se utilizan programadores tipo ICSP (In Circuit Serial Programming) o LVP (Low Voltage Programming). Existen varios tipos de estos programadores para transferir los datos de la computadora al PIC. Normalmente se necesitan varios softwares para su programación por lo que la programación de éstos se puede volver un poco complicada.

3.2.1.2. PICAXE

Existen los microcontroladores PICAXE que son microcontroladores PIC muy versátiles y de bajo costo. Los PICAXE son más fáciles de programar y de utilizar que los PIC ya que no necesitan circuitos programadores y el software de la programación es más amigable.

Los PICAXE cuentan con una memoria EEPROM la cual tiene grabado un pequeño software creado por la empresa Revolution Education. Realmente este software se programa en ciertos modelos de PIC ya diseñados, así que el funcionamiento es muy parecido a los PIC.

El software diseñado para la programación de los PICAXE es gratuito. Existen dos maneras en las que se pueden programar los PICAXE, el primero es realizando un diagrama de flujo, al cual el mismo software lo convierte a lenguaje ensamblador y lo programa en el PICAXE. La otra manera de programar los PICAXE es en lenguaje Basic el cual de igual manera lo transforma en lenguaje ensamblador y lo transfiere al PICAXE. Este software permite también hacer simulaciones previas a la transferencia del programa al PICAXE para poder encontrar errores o ver si realmente el programa funciona como se desea. Esto con la finalidad de ahorrar tiempo de programación y tiempo de vida del PICAXE que depende directamente del número de veces que se programa o se utiliza. Otra ventaja es que se pueden programar directamente desde el puerto serie o USB de la computadora al PICAXE sin necesidad de un circuito programador como lo es con los PIC.

Los PICAXE se programan mediante un cable de puerto serie o USB a plug estéreo conectado a la entrada y a la salida de programación mediante resistencias:

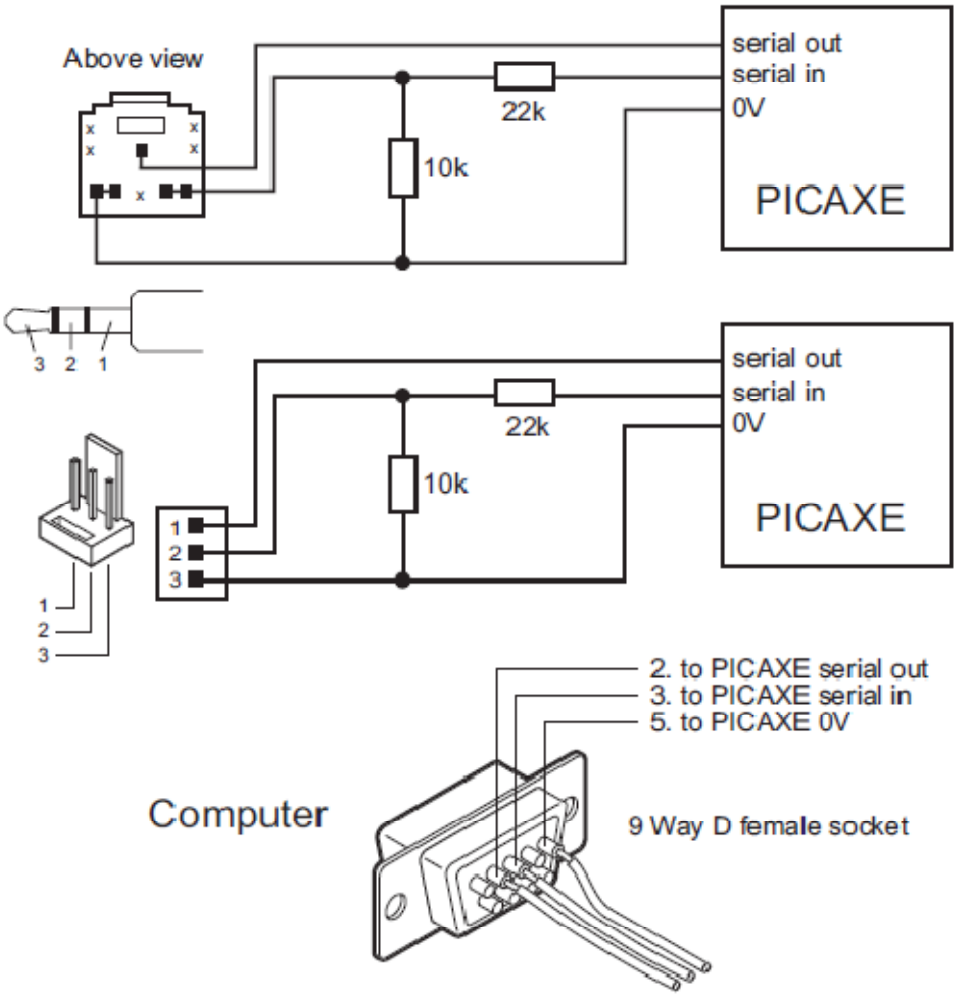


Figura 4: Conexión para cable de comunicación serie para la programación del PICAXE.

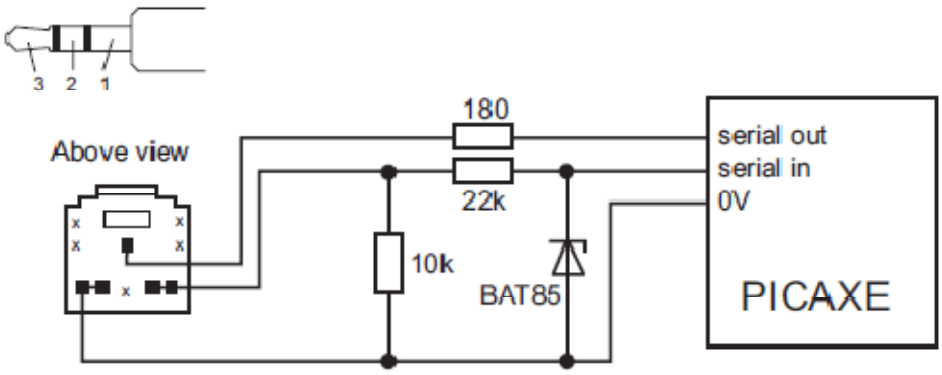


Figura 5: Conexión para cable de comunicación serie para la comunicación del PICAXE con la computadora.

Existen varios tipos de PICAXE con diferentes características:

Tabla 3: Modelos más recentes de los PICAXE en el mercado

Modelo PICAXE	Modelo Microchip	Pines	Líneas de programación	Salidas	Entradas	ADC *8bits **8/10bits	Frecuencia en MHz
PICAXE-08	PIC12F629-I/P	8	40	1-4	1-4	1*	4
PICAXE-08M	PIC12F683-I/P	8	80	1-4	1-4	3**	8
PICAXE-14M	PIC16F684	14	80	6	5	2**	8
PICAXE-18M	PIC16F88	18	80	8	5	3**	8
PICAXE-18X	PIC16F88-I/P	18	600	9	5	3**	8
PICAXE-20M	PIC16F677	20	80	8	8	4**	8
PICAXE-28X1	PIC16F886	28	1000	9-17	0-12	0-4**	20
PICAXE-28X2	PIC18F2420	28	1000	9-17	0-12	0-12**	40
PICAXE-40X1	PIC16F887	40	1000	9-17	8-20	3-7**	20
PICAXE-40X2	PIC18F4420	40	1000	9-17	8-20	0-12**	40

Nota1: Los PICAXE-28X-1/2 y PICAXE-40X-1/2 sustituyen a los antiguos PICAXE-28X/A y PICAXE 40X respectivamente, el PICAXE-18X sustituye al PICAXE-18A y el PICAXE-18M sustituye al PICAXE-18.

Nota2: Todos los PICAXE funcionan con una frecuencia de 4 MHz originalmente la cual se puede cambiar según la conveniencia.

3.2.1.3. LED (Light-Emitting Diode)

El LED es un dispositivo semiconductor que emite luz al ser polarizado directamente. Su funcionamiento es similar al de los diodos comunes de silicio o de germanio. Al pasar una corriente a través del diodo semiconductor se inyectan electrones en la región P y huecos en la región N. La intensidad de la luz depende de la magnitud de corriente que se haga pasar por el semiconductor para la recombinación de los electrones y huecos. Los LED's son de bajo consumo de energía y su tiempo de vida es muy extenso por lo que no necesitan mantenimiento.

Existen los LED's normales y los LED's súper brillantes, estos últimos, son los que se utilizaron en este proyecto. La diferencia consiste, como lo dice su nombre, en que los LED's súper brillantes tienen mayor intensidad luminosa que los LED's normales.

3.2.1.3.1. Proceso de caracterización de los LED's

La caracterización de los LED's ya se tenía de trabajos anteriores y constó de tres etapas: espectro óptico, divergencia angular y relación intensidad-corriente.

En la etapa de espectro óptico se muestran las longitudes de onda para cada uno de los LED's de los diferentes colores. Los resultados se obtuvieron mediante pruebas hechas con un espectrómetro tipo Ocean Optics HR4000 High-Resolution Fiber Optic Spectrometer. Los resultados obtenidos con los LED's súper brillantes fueron los siguientes:

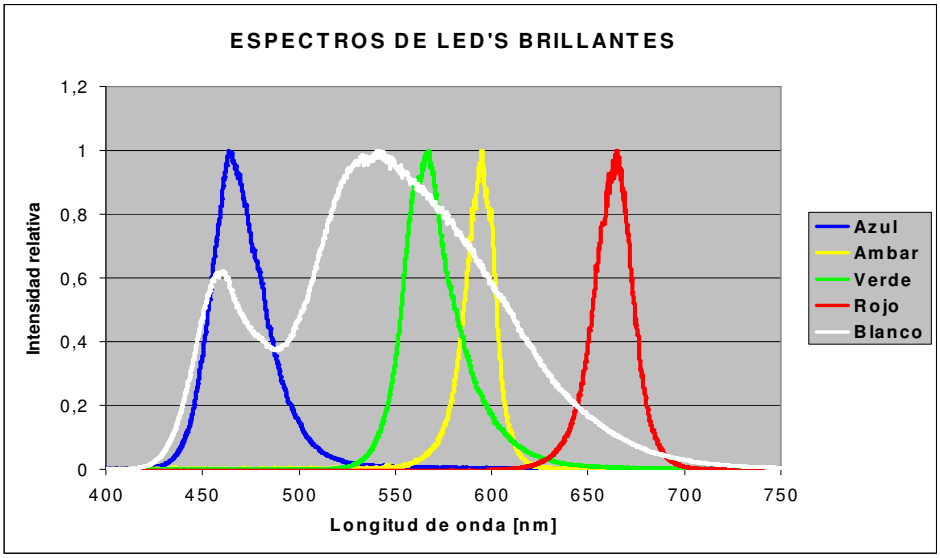


Figura 6: Espectros de los LED'S, se encuentran dentro del rango de visión de la mosca.

Cabe señalar que el espectro de los LED's súper brillantes es más estrecho que el de los LED's normales, por lo que los colores son mucho más definidos. También se nota que el LED blanco no cubre el espectro completo uniformemente como debería de ser y tiene un pico en las longitudes de onda del azul, sin embargo sí se distingue el color blanco del LED.

La etapa de divergencia angular se realizó con un dispositivo analizador de datos tipo Casio Data Analyzer EA-200 midiendo la cantidad de flujo luminoso en lúmenes haciendo girar cada LED en una platina rotatoria. Estos datos se midieron para obtener resultados de la divergencia que tiene cada haz de los LED's y así conocer la zona en donde se concentra la mayor cantidad de luz. Los resultados con los LED's súper brillantes fueron los siguientes:

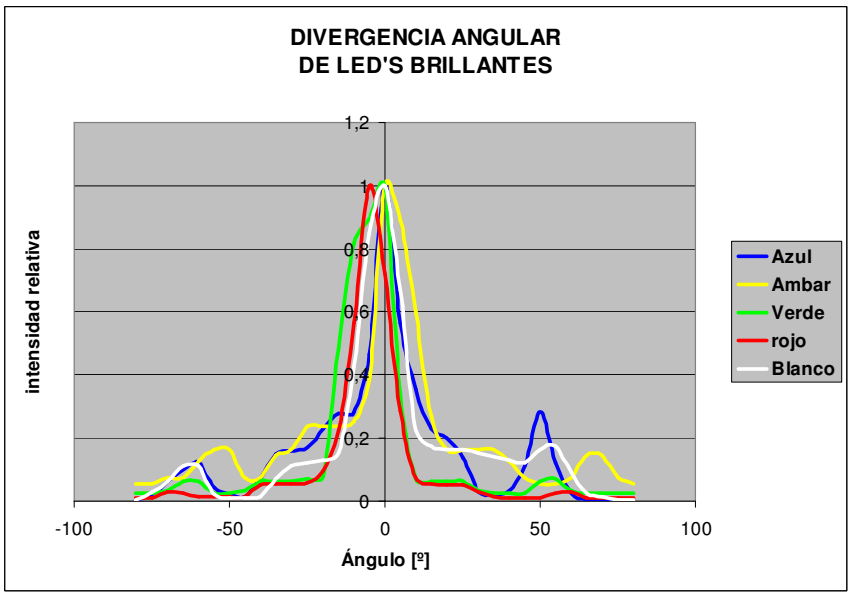


Figura 7: Divergencia angular de los LED's brillantes.

Con estos datos se puede decir que la mayor cantidad de luz se concentra en un cono de divergencia de alrededor de 40º.

La etapa de relación intensidad-corriente es en la que se midió precisamente la relación que hay entre la corriente de excitación y la intensidad luminosa de cada LED. Las mediciones se hicieron con el mismo dispositivo analizador de datos tipo Casio Data Analyzer EA-200 pasando cada vez mayor corriente al LED y midiendo su intensidad luminosa. Los resultados fueron los siguientes:

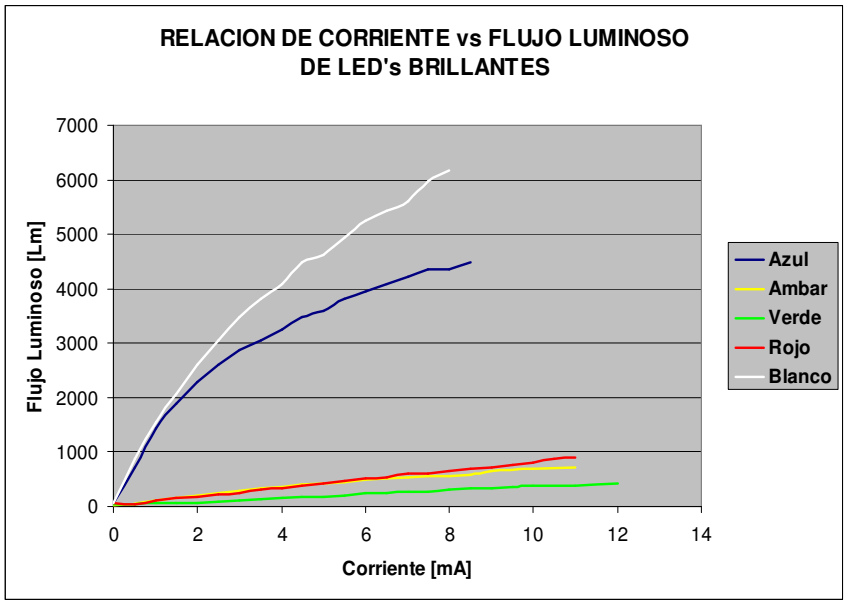


Figura 8: Relación de corriente contra flujo luminoso en LED's brillantes.

Se puede ver que los resultados no son lineales. Esto se debe en gran medida a la resolución de 0.333 lúmenes del analizador de datos utilizado, así que con esta gráfica se hicieron modelos matemáticos para poder obtener la relación lineal, los resultados son los siguientes:

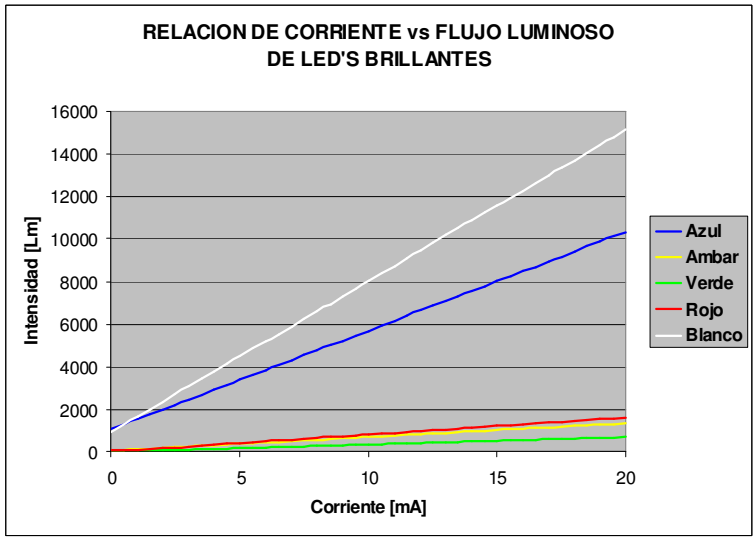


Figura 9: Relación de corriente contra flujo luminoso mediante modelos matemáticos en LED's brillantes.

En estas gráficas se puede ver que la relación intensidad-corriente es diferente en cada tipo de LED, siendo la del blanco y del azul mucho mayores, así que para poder tener la misma relación en todos los LED's se utilizaron diferentes resistencias para cada LED y así tener la misma intensidad en todos los LED's con la misma corriente. Los valores de las resistencias son las siguientes:

Blanco	22 [kΩ]
Azul	680 [Ω]
Rojo	660 [Ω]
Ámbar	338 [Ω]
Verde	100 [Ω]

3.2.2. Cambios realizados

El primer cambio hecho fue el de separar el tiempo de obscuridad inicial y el tiempo de obscuridad final. Para hacer este cambio simplemente, como la nueva tecnología de los PICAXE tenía más memoria, se agregaron los datos de entrada para cada obscuridad en la programación: segundos de obscuridad inicial, segundos de obscuridad final, minutos de obscuridad inicial, minutos de obscuridad final, horas de obscuridad inicial y horas de obscuridad final. Previamente solo había tres datos que eran los mismos para la obscuridad inicial que para la obscuridad final: segundos de obscuridad, minutos de obscuridad y horas de obscuridad. Ahora teniendo la posibilidad de meter los tiempos separados para la obscuridad inicial y final, simplemente en la programación se agregó el comando para que existiera sensado en la cámara de registro tanto en el tiempo de obscuridad inicial como en el tiempo de obscuridad final. Esto es hacer un barrido de recolección de datos en los sensores cada cuarto de segundo, para que cada segundo tuviese cuatro datos de cada sensor o treinta y dos datos por segundo contando todos los sensores.

Para poder hacer los otros dos cambios, se tuvo que analizar cuántas entradas, salidas y convertidores analógico-digital ADC se iban a necesitar. El número de entradas que se necesitan es mínimo, así que sólo se analizó el número de salidas y de ADC's:

	Salidas	**ADC's**
Potenciómetros digitales	8 (2 c/u)	4
LED's	20	
Comunicación compu.	1	
Sensores		8
Total	**29**	**12**

Se puede ver que son necesarias veinte salidas para los LED's teniendo sólo diez LED's. Esto es porque cada LED ocupa dos salidas al tener la posibilidad de encender dos LED's de cada lado a la vez.

El PICAXE más grande, que es el 40X-1, tiene ocho salidas y ocho puertos c que se pueden utilizar como salidas, por lo que se tiene la posibilidad de tener hasta dieciséis salidas, además tiene solamente siete ADC. Esto quiere decir que hacen falta trece salidas y cinco ADC's.

Existen otros microprocesadores que cubren las necesidades que se tienen para el problema. Se decidió quedarse con los PICAXE ya que por su precio es asequible, los PIC tienen las mismas características que los PICAXE en cuanto a salidas y ADC's y en el laboratorio de CECiB ya se contaban con algunos PICAXE que se utilizarán para la realización del proyecto.

Lo primero que se hizo fue reducir el número de ADC's que se iban a necesitar. Para saber el valor de los potenciómetros digitales, es necesario sensarlos por medio de los ADC's, es por eso que se necesita un ADC para cada potenciómetro. Sin embargo, fue posible hacer el sensado de los cuatro potenciómetros digitales con solamente un ADC. Esto fue posible por medio de interruptores con transistores.

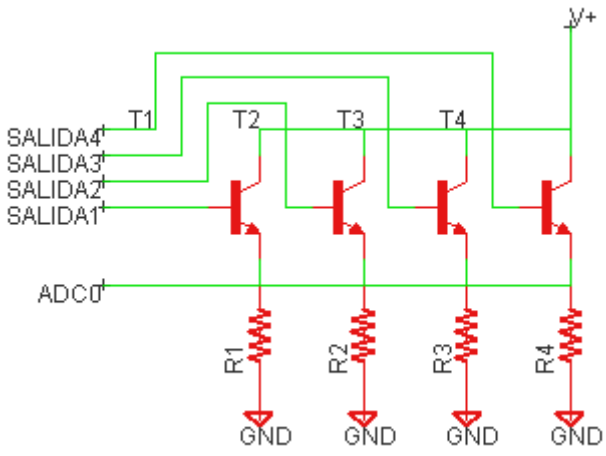


Figura 10: Interruptores con transistores de selección de LED's para utilizar un solo ADC.

Para poder controlar los interruptores con transistores se necesitó una salida extra de los microcontroladores para cada base de los transistores para cada potenciómetro digital. Este control hace que sólo los datos de un potenciómetro digital sean sensados por el ADC del PICAXE. Para esto es necesario que la salida del PICAXE que está conectada con la base del transistor del potenciómetro digital que se quiere sensar esté encendida y las otras tres apagadas.

El problema de hacer el switcheo era que se necesitaba una salida extra para cada potenciómetro digital para poder controlar este switcheo desde el PICAXE. Es

decir que el número de ADC's necesarios se redujo de doce a nueve, pero el número de salidas necesarias aumentó de diecinueve a veintitrés.

Como se puede ver, es imposible realizar el dispositivo con un PICAXE 40X-1, ya que faltarían salidas y ADC's, entonces se hizo que el PICAXE 40X-1 se comunicara con otro tipo de PICAXE para obtener lo necesario. Entonces se colocó un PICAXE 28X-1 que se comunicara con el PICAXE 40X-1. Éste tiene el mismo número de salidas, pero menos ADC's que el 40X-1. Se necesitaban nueve ADC's, el 40X-1 tiene 7, sólo faltaban dos más, y el 28X-1 tiene cuatro, entonces el número de ADC's disponibles cumplía con las expectativas. El problema sigue siendo las salidas, ya que ahora se requerían más para hacer posible la comunicación entre los PICAXE, así es que se colocó otro PICAXE que se comunicara con el 28X-1. Fue entonces que, para mantener la mayor simplicidad posible, se utilizó un PICAXE tipo 18A. Así es que las necesarias son:

	Salidas	**ADC's**
Potenciómetros digitales	12 (3 c/u)	1
LED's	20	
Comunicación compu.	1	
Comun. entre micros.	2	
Sensores		8
Total	**35**	**9**

Salidas y ADC's disponibles en un PICAXE 40X-1, uno 28X-1 y un 18A

	Salidas	**ADC's**
40X-1	16	7
28X-1	16	4
18A	8	3
Total	**40**	**14**

3.2.3. Electrónica

La electrónica del dispositivo cuenta con 6 partes: selección de color, intercambio de datos, programación, control de intensidad, monitoreo y alarma.

Selección de color:

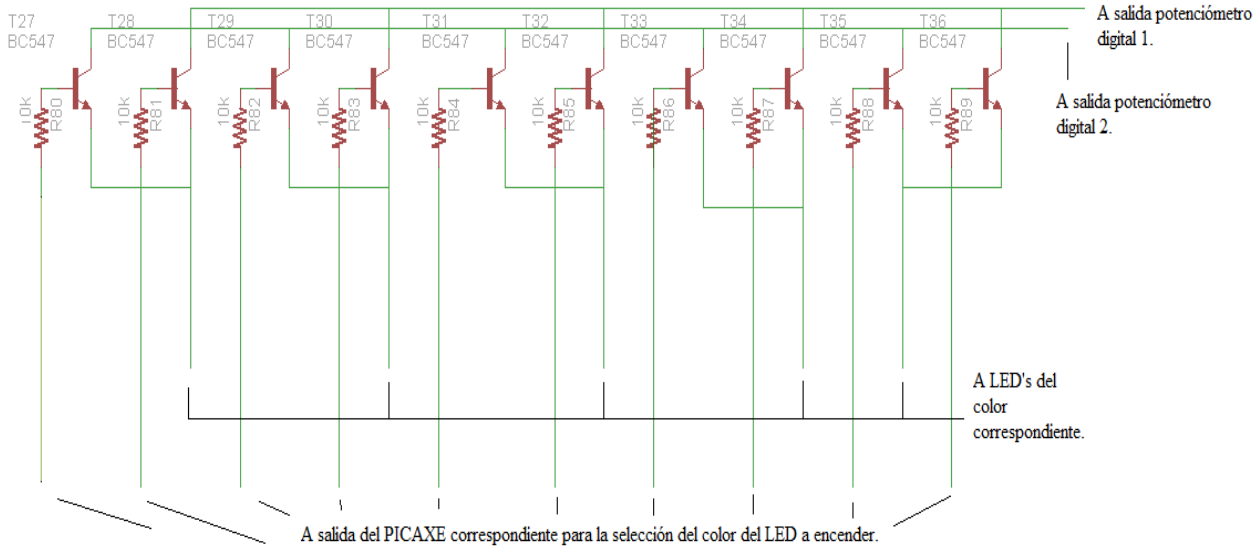


Figura 11: Arreglo de transistores y resistores para la selección del color del LED.

Intercambio de datos, con la computadora y entre PICAXES.

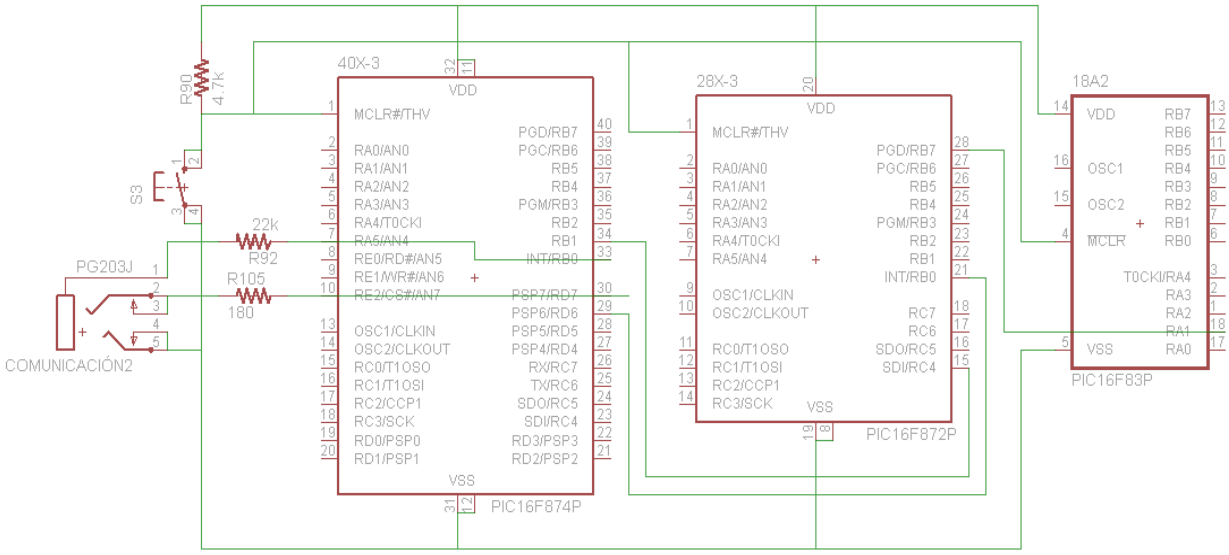


Figura 12: Arreglo para la comunicación de PICAXE a PICAXE y del PICAXE maestro a la computadora.

Programación

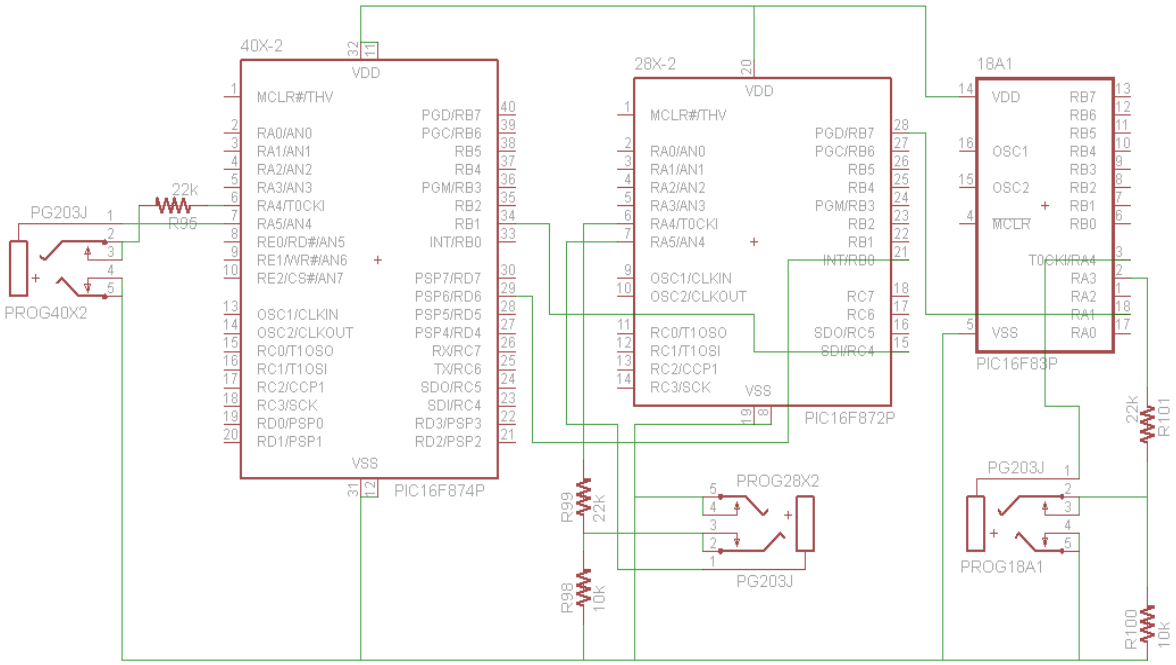


Figura 13: Arreglo para la programación de cada PICAXE de manera independiente.

Control de intensidad

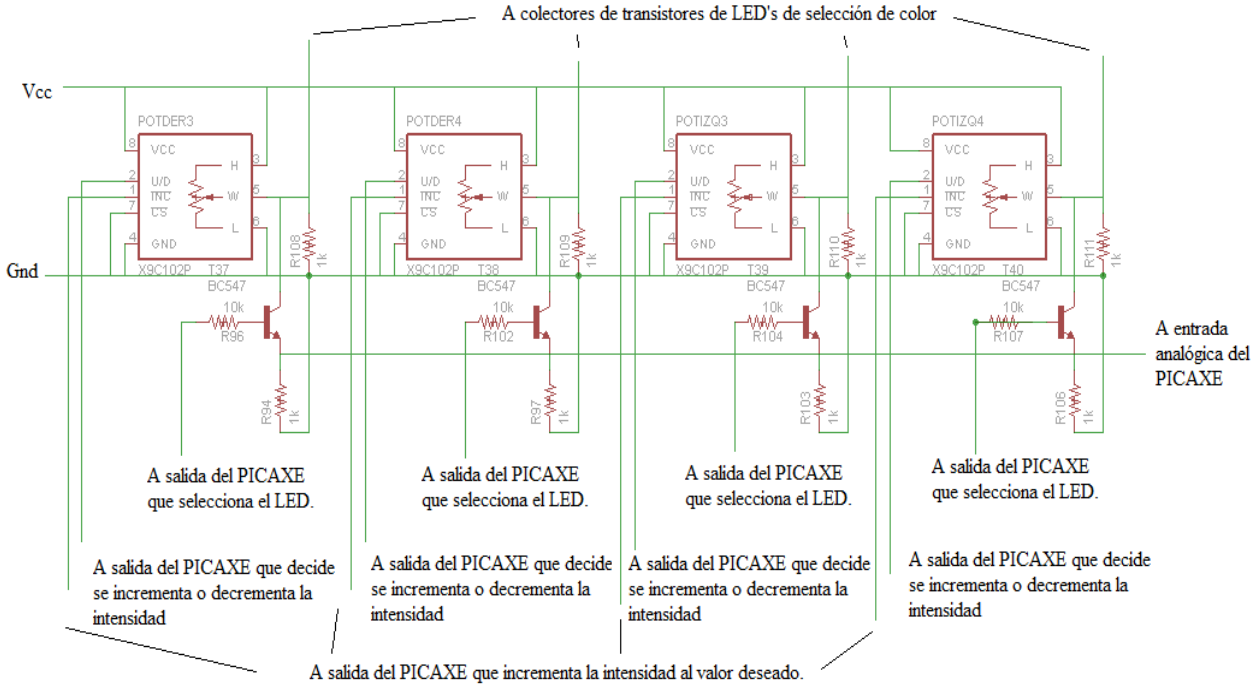


Figura 14: Arreglo en los potenciómetros digitales para el control de intensidad luminosa en los LED's.

Monitoreo

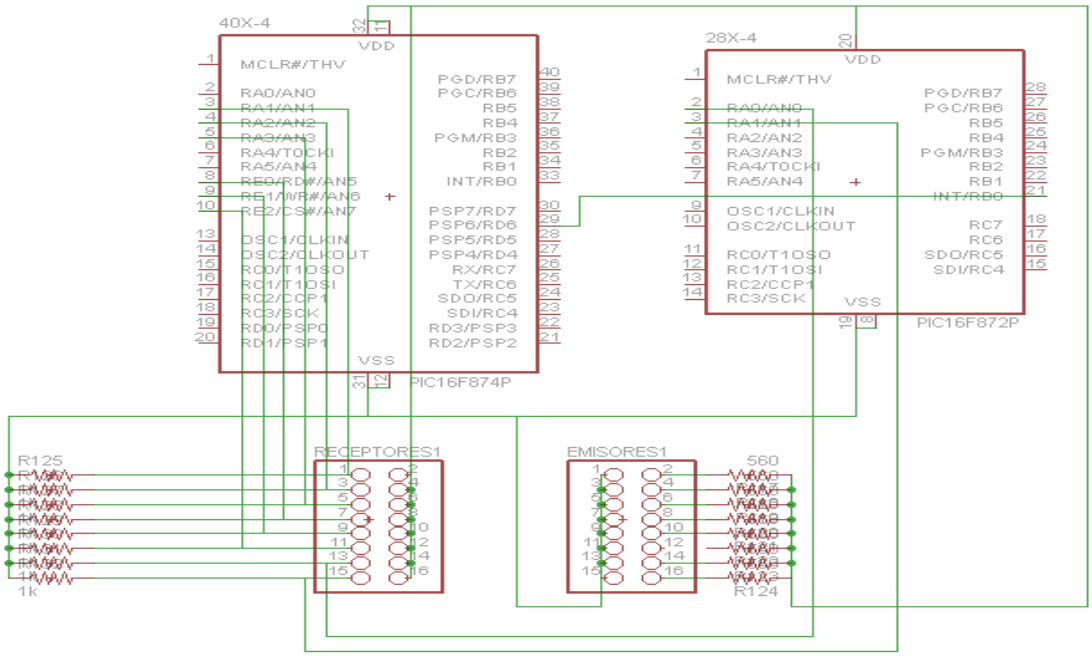


Figura 15: Conexión de emisores de infrarrojos y receptores de infrarrojos para monitorear a la mosca, cuando se interrumpa el flujo de éstos.

Alarma

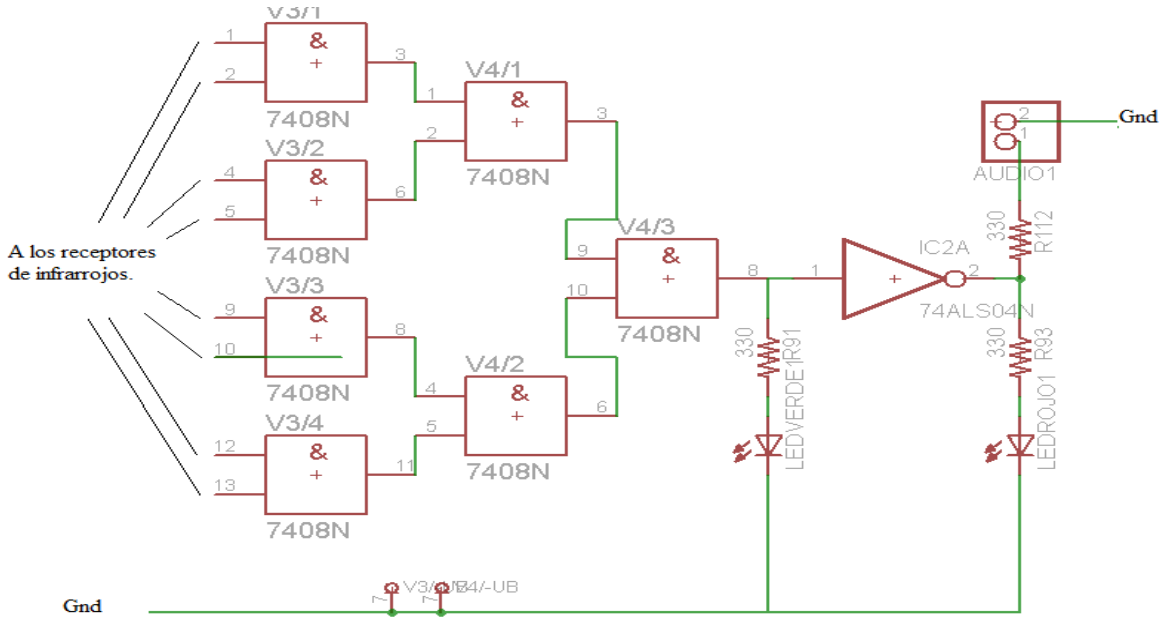


Figura 16: Conección de componentes lógicos para la alarma visual y sonora.

3.2.4. Programación y uso de los microcontroladores

La diferencia con el dispositivo con el que se contaba es que ahora necesita controlar dos microcontroladores, con un tercero fungiendo como maestro. Primero se definió un microcontrolador como maestro y los dos microcontroladores esclavos funcionarían en cascada, es decir uno sería esclavo del otro.

El microcontrolador maestro controlará a un esclavo, además de los potenciómetros, o sea la intensidad de luz de los LED's, además de hacer el control de seis de los ocho sensores de movimiento. El primer esclavo controlará al esclavo2, el encendido de los LED's, y controlará los otros dos sensores de movimiento. Por último, el esclavo2 controlará el encendido de los LED's restantes.

Para la programación del maestro, a cada potenciómetro se le asignaron tres salidas del microcontrolador. Una salida para el control de subida o bajada, otra para los pulsos que aumentará o disminuirá el valor del potenciómetro, y otra para la base del transistor que le corresponde a cada uno de los potenciómetros para el manejo del sensado. El primer ADC (convertidor analógico digital) mide el valor de los potenciómetros, y el resto de los ADC's serán usados para los sensores de movimiento. Por último se utilizó una salida para instrucciones al esclavo y una entrada para recibir información de éste.

Lo primero que hace el programa maestro es pedir y recibir toda la información que se necesita ingresar al programa. Esta información es: tiempos de obscuridad inicial en el formato de horas, minutos y segundos; tiempos de obscuridad final en el mismo formato; valores para dos colores e intensidades de ambos lados, cuatro en total; también hay que proporcionar tiempos del experimento en horas, minutos y segundos. En total son 17 valores los que requiere el programa.

La etapa siguiente es la del control de intensidad. En esta etapa se inicializa a cada potenciómetro en el valor previamente deseado. Si el valor del potenciómetro es mayor al del valor donde está totalmente apagado el LED, ejecuta una subrutina la cual en lugar de aumentar el valor del potenciómetro lo disminuye.

Ya que cada LED está en el valor inicial, el programa pasa a otra subrutina la cual le manda al potenciómetro el número de pulsos para ajustar el valor de intensidad deseada. La subrutina siguiente es la encargada del monitoreo de la mosca, primero para el tiempo de obscuridad inicial, luego para el tiempo del experimento y por último el de obscuridad final. Para cada subrutina de monitoreo, se hace el sensado de cada ADC del maestro y también recibe la información del resto de los otros dos ADC del esclavo. Se ha temporizado de tal forma que cada subrutina se ejecute cuatro veces por segundo.

Cada dos salidas del esclavo1 y del esclavo2 le corresponden a un LED, esto es para poder tener la posibilidad de encender dos LED's de cada lado al mismo tiempo, si la necesidad de encender tres LED's al mismo tiempo de cada lado se diera, se necesitarían tres salidas para cada LED.

La programación del esclavo comienza con la recepción de los colores que fueron seleccionados por el usuario y al mismo tiempo mandar los colores correspondientes al esclavo2. A continuación hay una subrutina para el sensado de los ADC's para los sensores de movimiento. Esta subrutina primero recibe del maestro los datos del tiempo de obscuridad inicial y con ese dato la subrutina se va a repetir según

su valor. A continuación, la subrutina lee los datos en los ADC's para mandarlos al maestro y al terminar, continúa con la subrutina de selección de salidas para LED's.

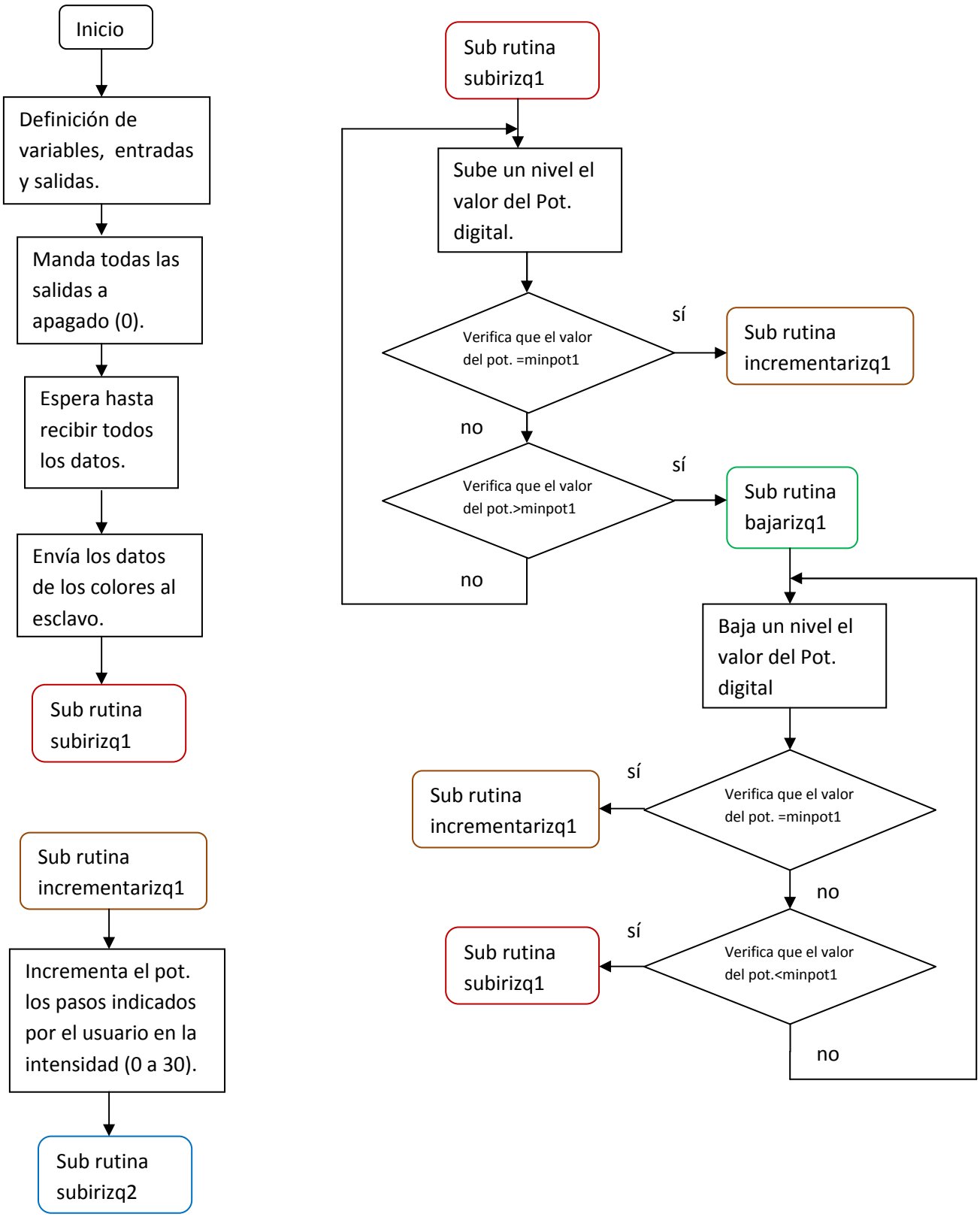
Esta subrutina es la de selección de color y encendido de LED's. El programa reconocerá los colores introducidos y según los colores, el programa encenderá las salidas del microcontrolador correspondientes a los LED's que se desean encender. Esta etapa consume mucha memoria, ya que la combinación de colores amplió las diferentes formas de encender las salidas.

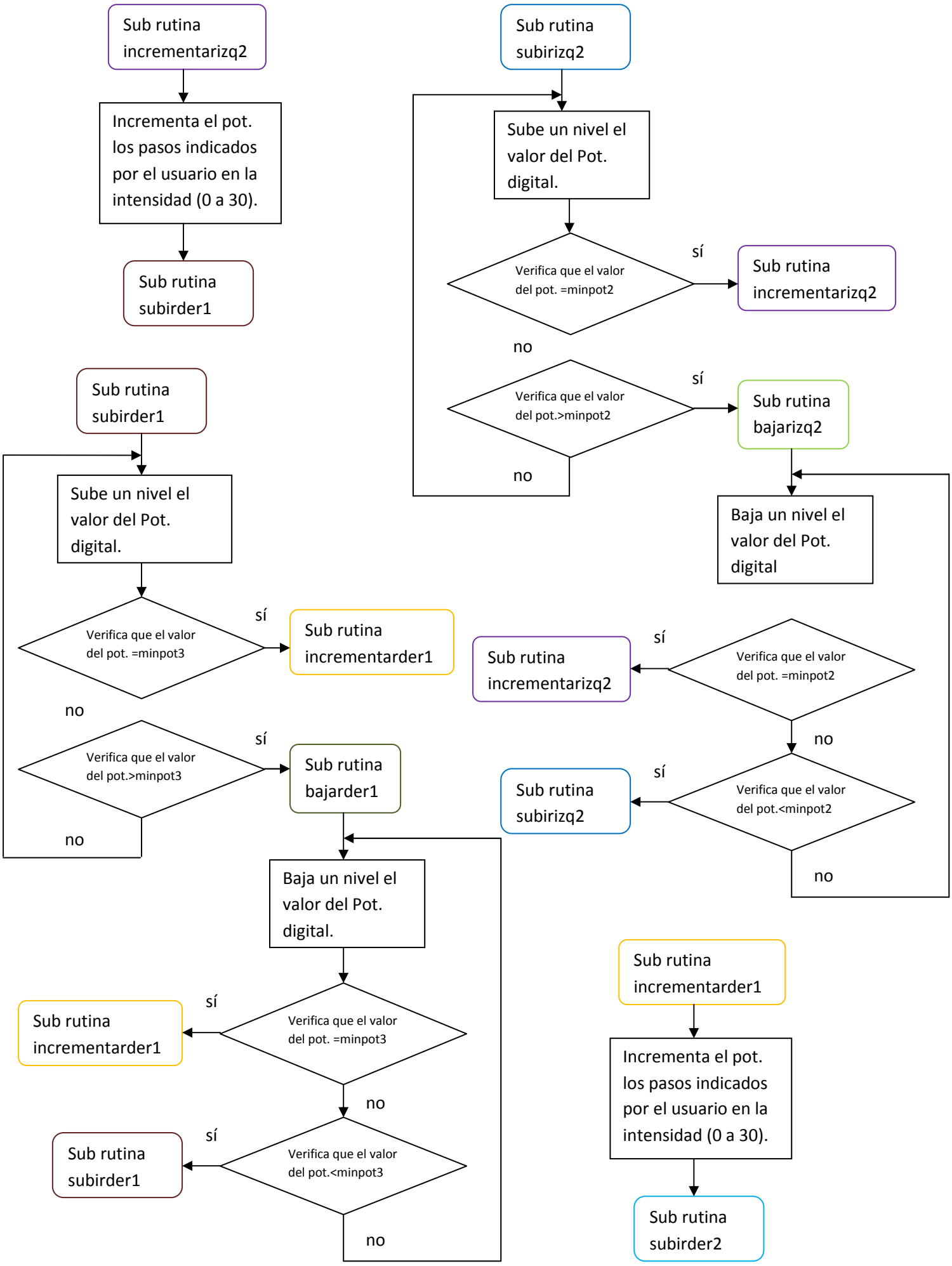
Una vez encendidos los LED's deseados, las siguientes subrutinas son iguales que la primera subrutina de la programación del esclavo. La primera es para el monitoreo de la mosca en el tiempo del experimento. La segunda para apagar los LED's además de hacer el monitoreo de la mosca cuando está en el tiempo de obscuridad final. Al terminar estas subrutinas, el programa regresa al principio a la espera de recibir datos de nueva cuenta del maestro.

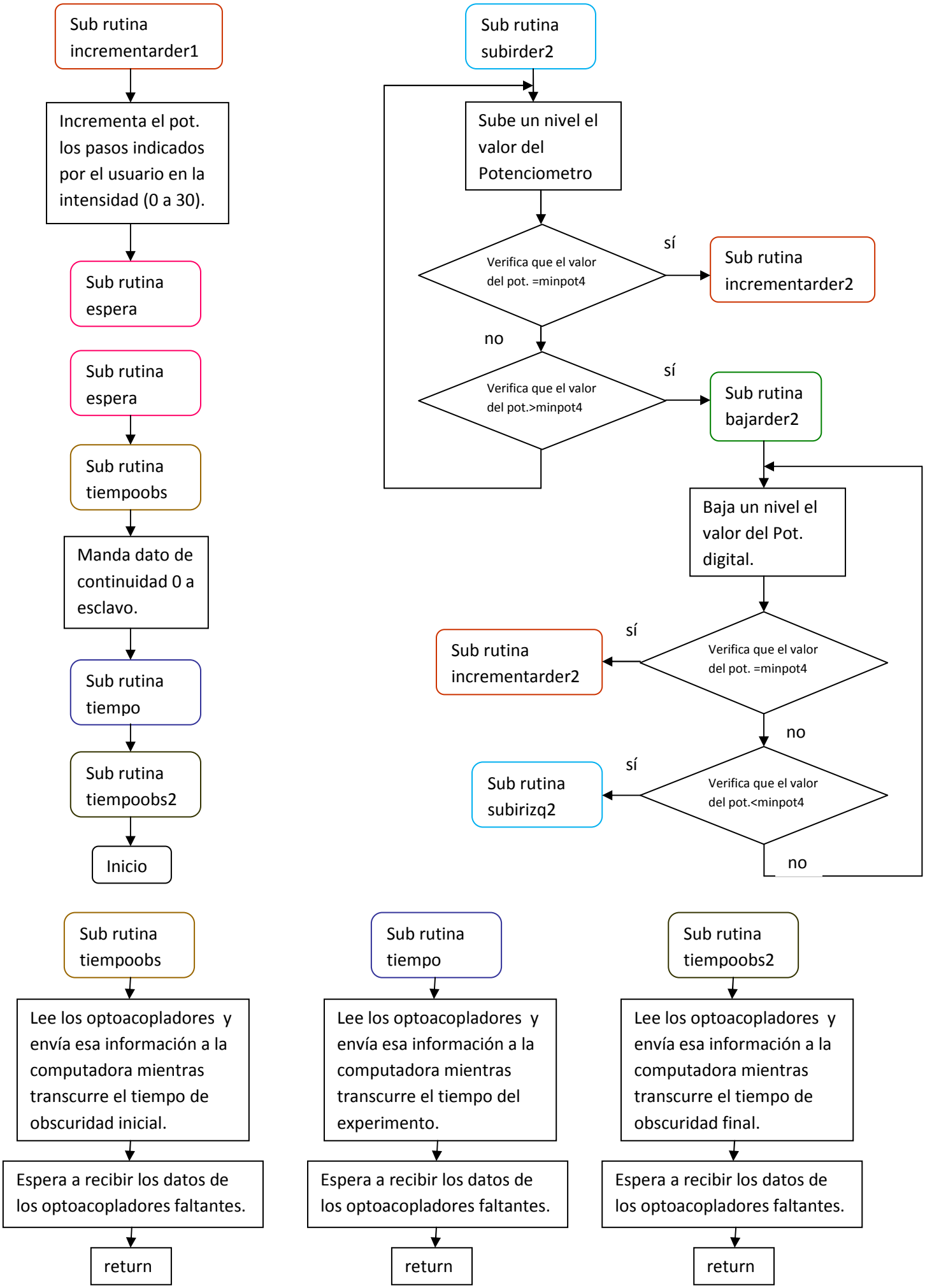
Por último el esclavo2 utiliza sus salidas para los LED's que no tuvieron cabida en el esclavo1. Lo único que hace el programa del esclavo2 es recibir los datos de los colores y encender los LED's según la elección inicial, esperar el tiempo de experimento y después apagar estos LED's y regresar a la espera de recibo de datos de nueva cuenta del esclavo.

3.2.4.1. Diagrama de flujo de los PICs

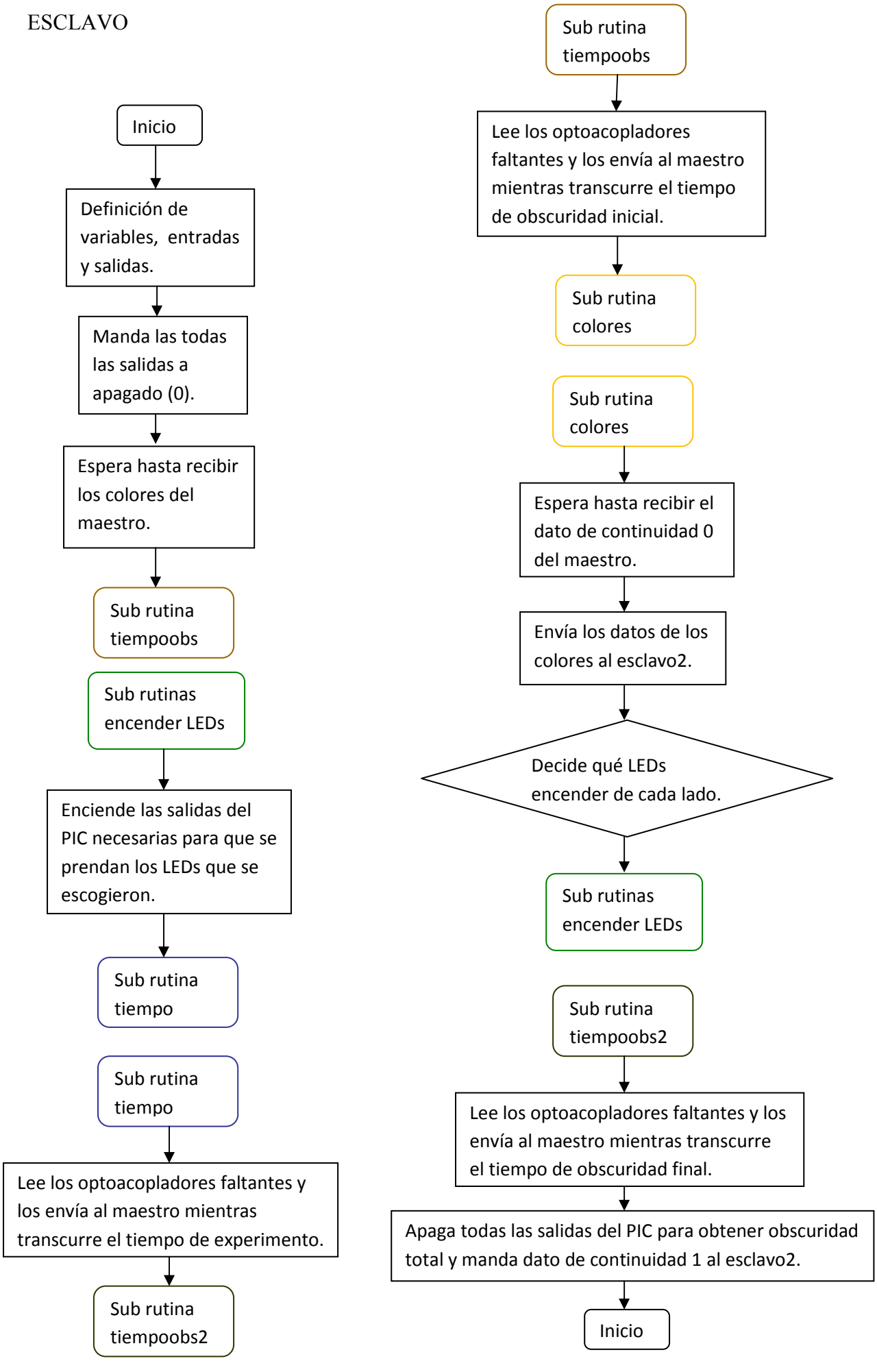
MAESTRO



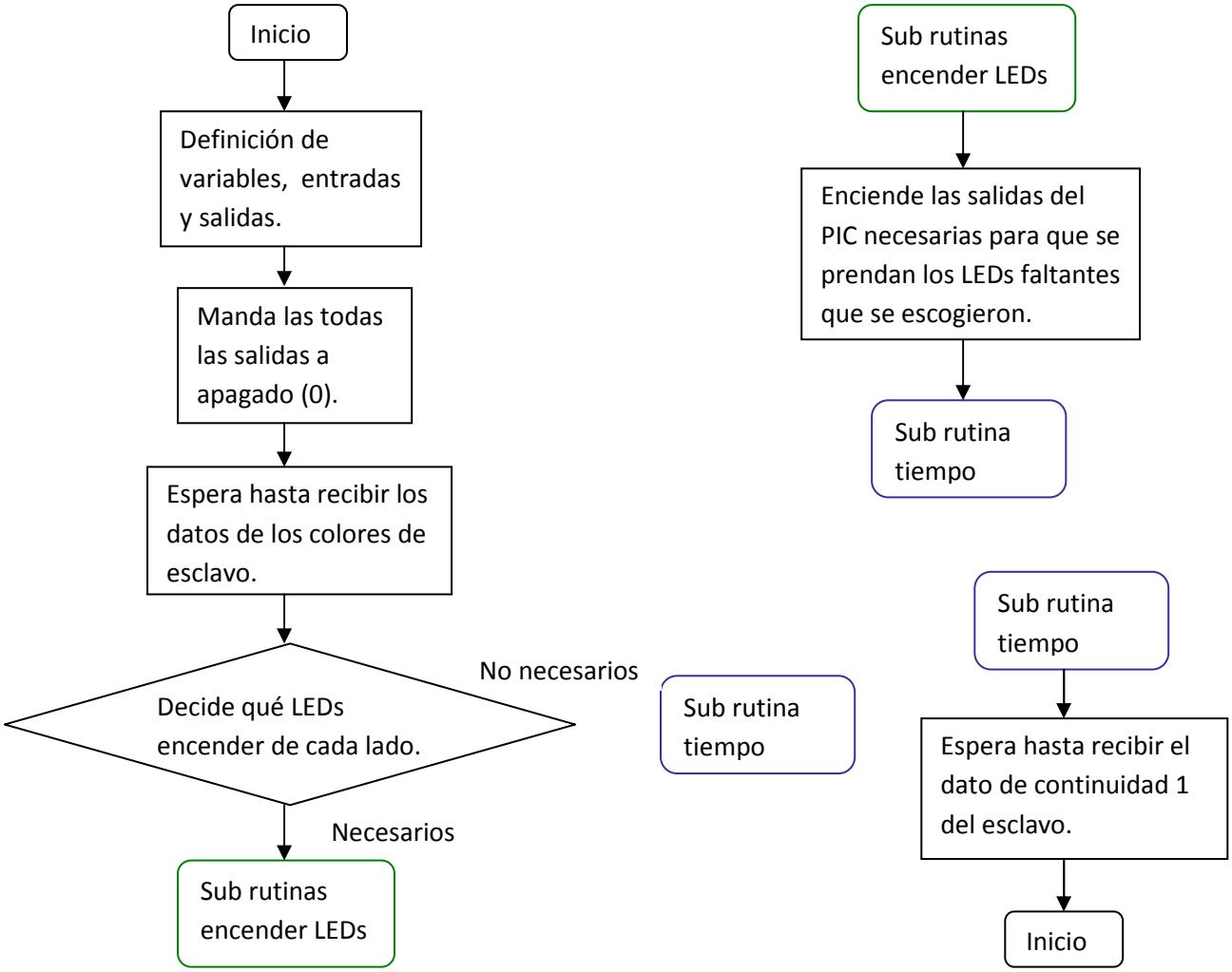

Sub rutina incrementarizq2
Incrementa el pot. los pasos indicados por el usuario en la intensidad (0 a 30).
Sub rutina subirder1
Sub rutina subirizq2
Sube un nivel el valor del Pot. digital.
Verifica que el valor del pot. =minpot2
sí
Sub rutina incrementarizq2
no
Verifica que el valor del pot.>minpot2
sí
Sub rutina bajarizq2
no
Baja un nivel el valor del Pot. digital
Sub rutina subirder1
Sube un nivel el valor del Pot. digital.
Verifica que el valor del pot. =minpot3
sí
Sub rutina incrementarder1
no
Verifica que el valor del pot.>minpot3
sí
Sub rutina bajarder1
no
Sub rutina incrementarizq2
sí
Verifica que el valor del pot. =minpot2
no
Sub rutina subirizq2
sí
Verifica que el valor del pot.<minpot2
no
Baja un nivel el valor del Pot. digital.
Sub rutina incrementarder1
sí
Verifica que el valor del pot. =minpot3
no
Sub rutina subirder1
sí
Verifica que el valor del pot.<minpot3
no
Sub rutina incrementarder1
Incrementa el pot. los pasos indicados por el usuario en la intensidad (0 a 30).
Sub rutina subirder2


Sub rutina incrementarder1
Incrementa el pot. los pasos indicados por el usuario en la intensidad (0 a 30).
Sub rutina espera
Sub rutina espera
Sub rutina tiempoobs
Manda dato de continuidad 0 a esclavo.
Sub rutina tiempo
Sub rutina tiempoobs2
Inicio
Sub rutina subirder2
Sube un nivel el valor del Potenciometro
Verifica que el valor del pot. =minpot4
sí
Sub rutina incrementarder2
no
Verifica que el valor del pot.>minpot4
sí
Sub rutina bajarder2
Baja un nivel el valor del Pot. digital.
Verifica que el valor del pot. =minpot4
sí
Sub rutina incrementarder2
no
Verifica que el valor del pot.<minpot4
sí
Sub rutina subirizq2
no
Sub rutina tiempoobs
Lee los optoacopladores y envía esa información a la computadora mientras transcurre el tiempo de obscuridad inicial.
Espera a recibir los datos de los optoacopladores faltantes.
return
Sub rutina tiempo
Lee los optoacopladores y envía esa información a la computadora mientras transcurre el tiempo del experimento.
Espera a recibir los datos de los optoacopladores faltantes.
return
Sub rutina tiempoobs2
Lee los optoacopladores y envía esa información a la computadora mientras transcurre el tiempo de obscuridad final.
Espera a recibir los datos de los optoacopladores faltantes.
return

ESCLAVO


Inicio
Definición de variables, entradas y salidas.
Manda las todas las salidas a apagado (0).
Espera hasta recibir los colores del maestro.
Sub rutina tiempoobs
Sub rutinas encender LEDs
Enciende las salidas del PIC necesarias para que se prendan los LEDs que se escogieron.
Sub rutina tiempo
Sub rutina tiempo
Lee los optoacopladores faltantes y los envía al maestro mientras transcurre el tiempo de experimento.
Sub rutina tiempoobs2
Sub rutina tiempoobs
Lee los optoacopladores faltantes y los envía al maestro mientras transcurre el tiempo de obscuridad inicial.
Sub rutina colores
Sub rutina colores
Espera hasta recibir el dato de continuidad 0 del maestro.
Envía los datos de los colores al esclavo2.
Decide qué LEDs encender de cada lado.
Sub rutinas encender LEDs
Sub rutina tiempoobs2
Lee los optoacopladores faltantes y los envía al maestro mientras transcurre el tiempo de obscuridad final.
Apaga todas las salidas del PIC para obtener obscuridad total y manda dato de continuidad 1 al esclavo2.
Inicio

ESCLAVO 2

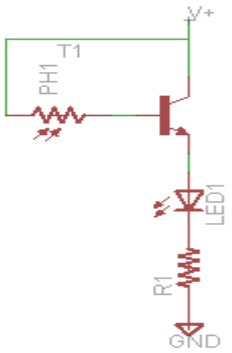


3.2.5. Alarma de fallo

Esta alarma fue diseñada e implementada por la experiencia del dispositivo anterior. Para ver resultados había que esperar a que terminara todo el proceso; es decir, tiempo de obscuridad inicial, tiempo de experimento y tiempo de obscuridad final. Dado que en las pruebas de tiempo prolongado, durante el tiempo de funcionamiento se podía presentar algún problema con los LED's o sensores y no se detectaba sino hasta el final del mismo. De esta forma la alarma permitirá dar aviso en caso de que deje de funcionar el dispositivo de forma óptima y poder terminar el experimento. La alarma consiste de dos sistemas independientes, el del funcionamiento de los sensores y el de comprobación de luz dentro de la cámara de registro.

El sistema de indicador de luz en el interior de la cámara consiste de un foto resistor en cada extremo del interior de la cámara de registro. Cada uno de los foto resistores está conectado en un extremo al voltaje y en el otro a la base de un transistor,

el colector se conecta a la fuente de voltaje y el emisor a un LED. Cuando hay luz en el foto resistor, su resistencia disminuye y así hay mayor paso de corriente, por lo que el LED que está conectado al emisor de cada transistor se enciende según la intensidad de luz que hay en extremo correspondiente dentro de la cámara de registro. Si no hay luz en el interior, los LED's se encontrarán totalmente apagados.



El sistema de alarma de los sensores consiste de compuertas de diseño digital tipo AND y NOT. Cada uno de los receptores de infrarrojos está conectado a una entrada de las compuertas tipo AND, esto hace que en la salida siempre haya potencial eléctrico siempre y cuando todos los receptores de infrarrojos estén recibiendo algún tipo de señal. A la salida de estas compuertas AND está conectado un LED de color verde. El LED se enciende cuando la salida de las compuertas es positiva y los receptores funcionan correctamente. A la salida de las compuertas AND también se conecta una compuerta NOT, la cual invierte el valor que hay en la salida de las compuertas AND. Así que si la salida es negativa, el LED verde no se encenderá, sin embargo la compuerta NOT invertirá la señal y a su salida se tendrá positivo. Entonces a la salida de la compuerta NOT se conecta otro LED, pero ahora de color rojo, el cual se encenderá cuando algún receptor de infrarrojo no está funcionando correctamente. También se tendrá una bocina conectada a la salida de la compuerta NOT, para que la alarma además de ser visual, sea sonora.

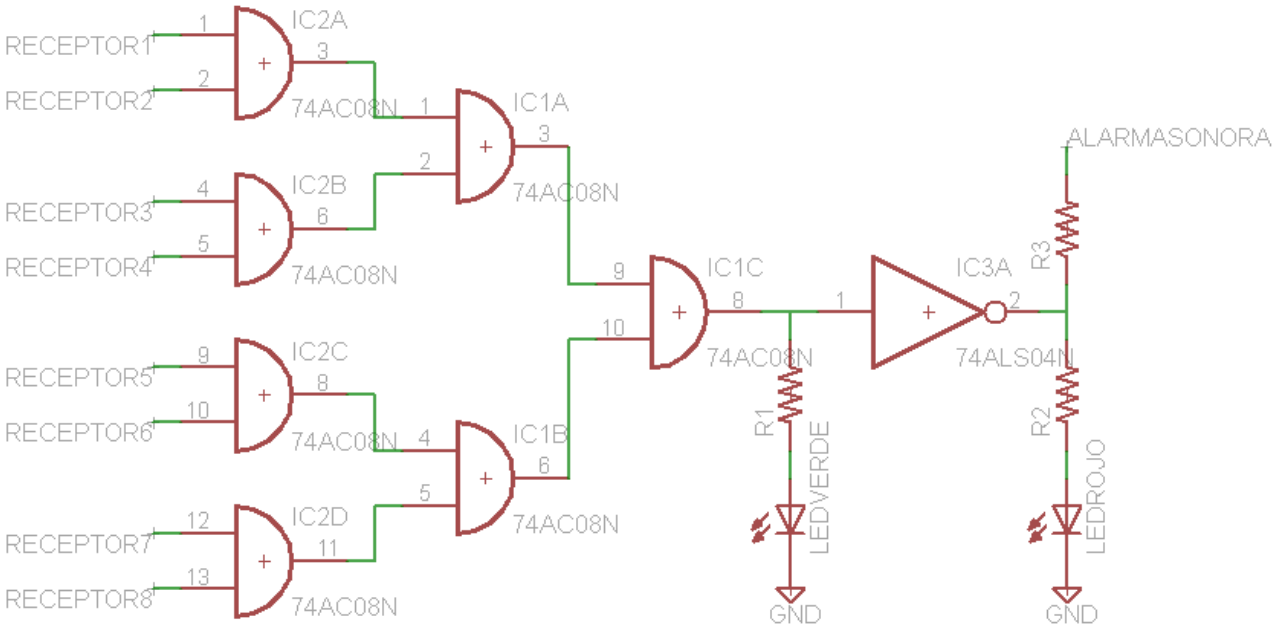


Figura 15: Conexión de componentes lógicos para la alarma visual y sonora

3.3.Desarrollo del software

La interfaz fue realizada con Matlab con la función de “guide”. Esta función permite hacer una ventana donde se introducen los datos que pide el programa para enviarlos a los microcontroladores, y al final proporcionar las gráficas donde estarán los resultados del experimento.

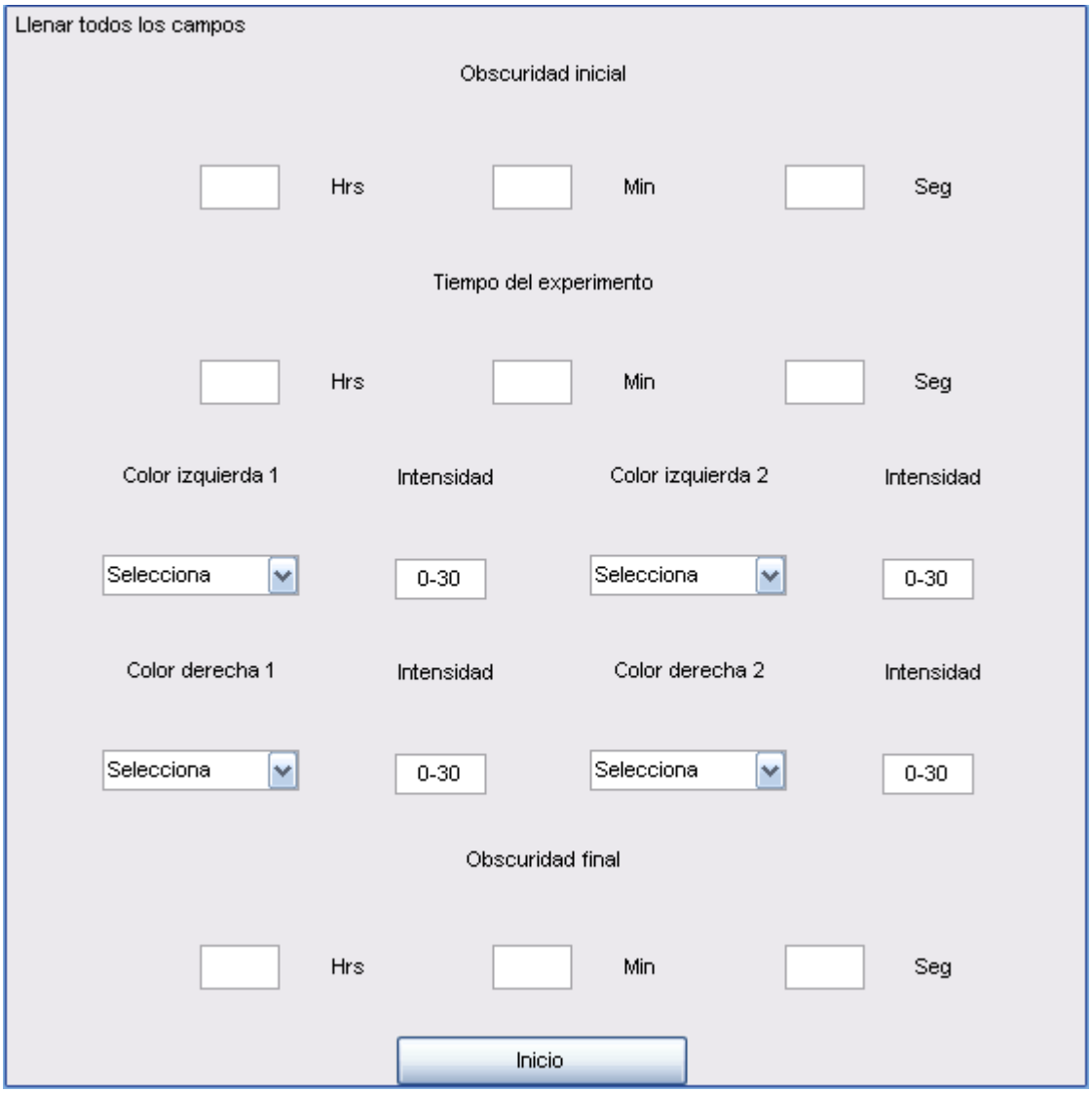


Figura 16: Ventana de selección de datos para enviar al dispositivo.

Se cuenta con entradas para cada dígito de los tiempos de obscuridad inicial, experimento y obscuridad final: horas, minutos y segundos. También hay espacios para cada intensidad de los LED's: izquierda 1, izquierda 2, derecha 1 y derecha 2; selección de color para cada LED: color izquierdo 1, color izquierdo 2, color derecho 1 y color derecho 2. Dando un total de diecisiete datos que el usuario debe ingresar en la interfaz para que funcione el programa de los microcontroladores.

La interfaz tiene programados los límites dentro los cuales debe estar cada dato y si no se respetan esos límites saldrá una ventana de error. Los límites para las horas es de 0 a 4, los minutos y los segundos entre 0 y 59, las intensidades entre 0 y 30, y por último las barras de selección de los colores disponibles de LED's.

El paso siguiente, ya que se introducen los datos en Matlab, es mandar los datos de Matlab al microcontrolador maestro para que comience a operar el sistema. Se debe tomar en cuenta que es diferente la forma de transmitir los datos que son números a los que son caracteres.

El paso siguiente es que el programa reciba los datos que el microcontrolador maestro envía. Para esto es necesario hacer una función la cual reciba los datos en el tiempo necesario y que soporte el tamaño necesario para no perder todos los datos que envía el microcontrolador maestro. Esta función se realiza tres veces, una para la recepción de los datos de obscuridad inicial, otra para los datos del experimento y otra para los datos de obscuridad final.

Después de cada recepción de datos, se organizan y se dibujan las gráficas de los datos. Los datos se reciben en columnas como sigue: el primer dato corresponde al primer sensor, el segundo, al segundo sensor, y así sucesivamente hasta terminar con los ocho sensores. Teniendo los datos en este orden, se normalizan los valores de cada columna para que el dato mayor sea uno. Después se multiplican los datos por cinco para que la variación del valor entre datos sea mayor, y se le suma o resta un valor según el sensor correspondiente. Por ejemplo, los datos del primer sensor ya multiplicados todos por cinco, se le restarán cuatro unidades, para que el dato mayor sea uno y en la gráfica la línea del primer sensor aparezca en el uno del eje "y". Esto se hace porque si le sumamos el valor directamente del sensor que le corresponde, la gráfica obtenida no sería muy explícita, en cambio multiplicando previamente los datos por cinco, se escala la salida para obtener una gráfica más explícita. Ya teniendo los datos de esta manera se pueden dibujar todos en gráficas de forma simultánea, así que los datos da cada sensor se ubican en líneas diferentes,.

De esta forma cuando exista una perturbación en los sensores, en la gráfica aparecerá como una perturbación en la línea correspondiente a cada sensor.

Otra característica adicional es que el programa desplegará una tabla con un contador de cada sensor, contabilizando las veces que pasa la mosca por cada sensor. Esto se implementó comparando los datos de ruido y de cuándo pasa la mosca, los cuales son diferentes. Sabiendo los datos de cuándo pasa la mosca, una variable se sumará y contará las veces que pasa por cada sensor.

3.4.Caracterización del dispositivo

Este proceso consistió en medir la intensidad luminosa en lúmenes para cada intensidad de 0 a 30 unidades de los diferentes colores de LED's. Estas mediciones se realizaron a diferentes distancias, a 1 cm, 2 cm, 3 cm y 4 cm de separación entre el LED y el medidor. Para esto se usó la **Casio Data Analyzer EA-200,** la cual utiliza un fotorresistor el cual tiene la siguiente respuesta relativa:

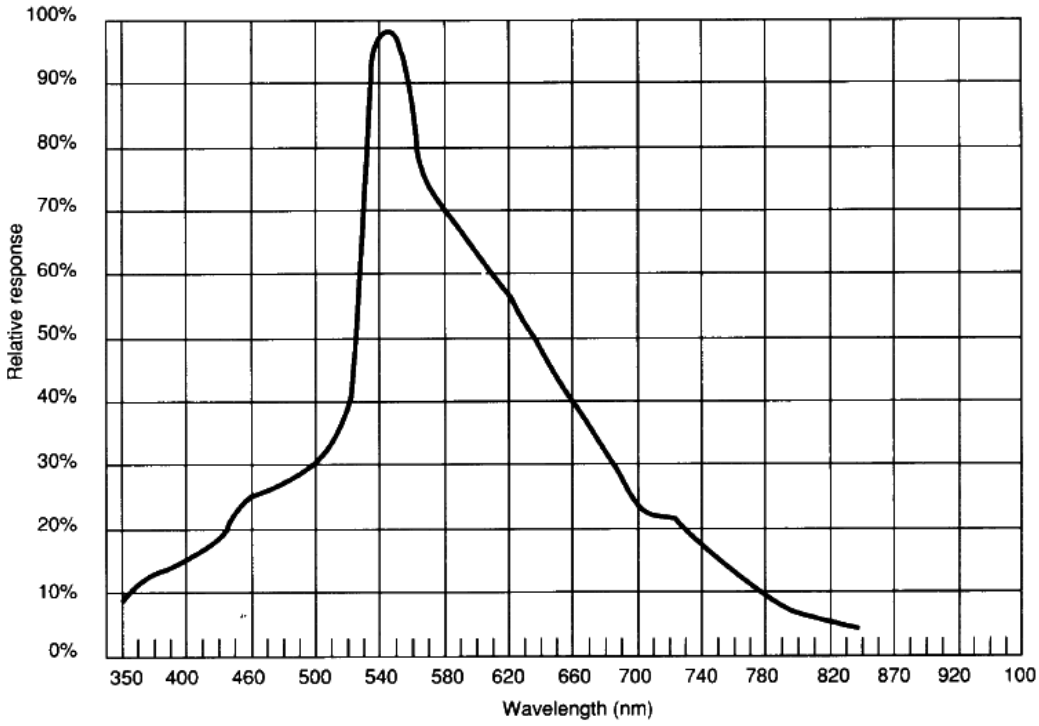


Figura 16: La gráfica muestra la respuesta relativa de la fotorresistencia dependiente de la longitud de onda de la señal a cuantificar.

Las intensidades siguientes son a un centímetro de distancia

Intensidad Relativa	Azul	Rojo	Verde	Blanco	Ámbar
0	0,26	0,26	0,26	0,26	0,26
1	0,26	0,26	0,26	0,26	0,26
2	0,26	0,26	0,26	0,26	0,26
3	0,26	0,26	0,26	0,26	0,26
4	0,26	0,26	0,26	0,26	0,26
5	0,26	0,26	0,26	0,26	0,26
6	0,26	0,26	0,26	0,26	0,26
7	0,26	0,26	0,26	0,26	0,26
8	0,26	0,26	0,26	0,26	0,26
9	0,26	0,26	0,26	0,26	0,26
10	0,26	0,26	0,26	0,26	0,26
11	0,26	0,36	0,26	0,26	0,26
12	0,26	0,42	0,26	0,26	0,36
13	0,26	0,46	0,26	0,26	0,4
14	0,26	0,46	0,26	0,26	0,79
15	0,26	0,79	0,26	0,26	1,46
16	0,26	1,01	0,36	0,26	1,6
17	0,26	1,21	0,51	0,26	4,66
18	0,26	1,74	1,31	0,26	4,89
19	0,26	2,03	2,16	0,26	7,49
20	0,33	2,25	4,78	0,26	11,25
21	0,33	2,81	4,87	0,35	11,49
22	0,67	3,45	5,16	2,95	19,76
23	2,03	4,38	10,66	21,28	20,04
24	4,66	5,53	15,19	61,27	32,64
25	8,64	8,44	21,49	113,9	41,25
26	13,71	10,67	22,46	182,29	51,21
27	20,53	13,03	30	263,48	64,41
28	29,69	16,4	43,3	365,29	79,68
29	39,39	16,88	62,04	494,23	99,35
30	54,23	21,64	136,59	608,25	134,97

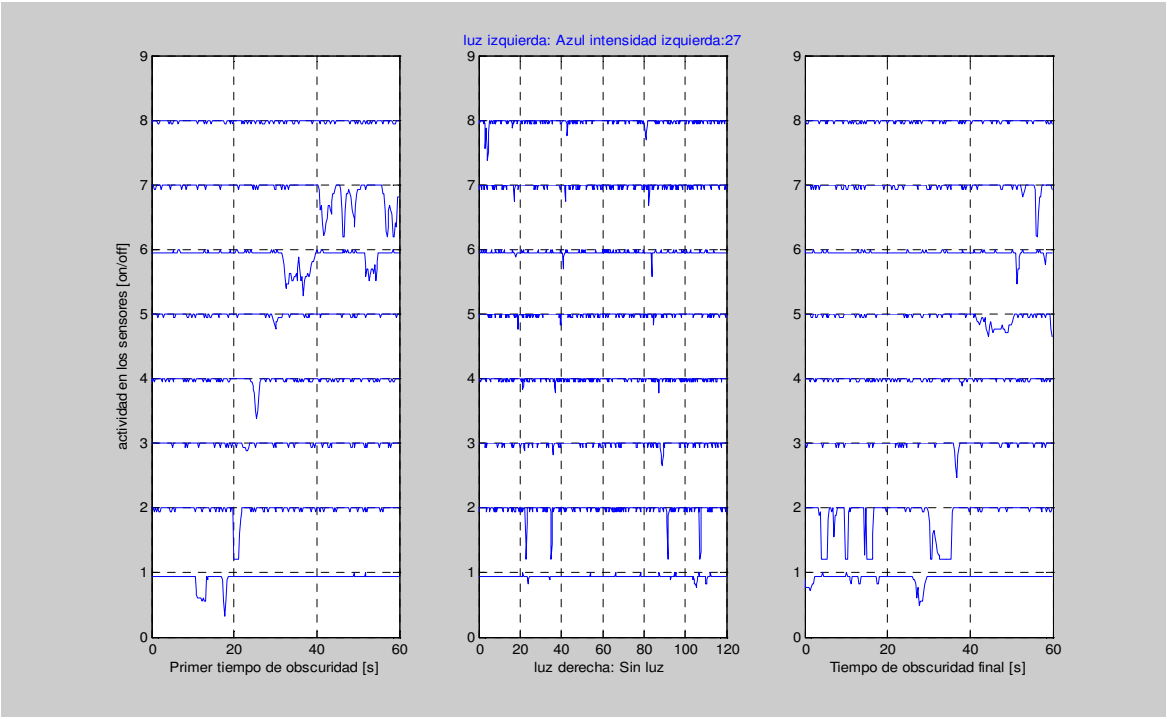
Tabla 4: Lúmenes según el valor dado al dispositivo para cada LED de diferente color.

Los valores que están en un recuadro son cuando el LED comienza a tener efecto en el fotorresistor del analizador de datos. Los valores que están sombreados son cuando comienzan a verse a simple vista. Es por esto que se decidió tener un rango de intensidades de 0 a 30, siendo 0 cuatro unidades menor al valor de cuando comienza a verse el primer LED, el rojo, a simple vista, ya que realmente lo que se quiere estudiar es el efecto de la luz en la mosca. Se pueden ver las intensidades de las diferentes distancias en el apéndice G.

4. Resultados

Se realizaron experimentos con cinco moscas diferentes con un minuto de tiempo de obscuridad inicial, dos minutos de experimento y un minuto de tiempo de obscuridad final. Se hicieron pruebas con las mismas intensidades dadas en la caracterización de cada LED y con algunas combinaciones de colores, así como con pura obscuridad.

Mosca 1, azul izquierda, intensidad 27 (20.53 Lm)

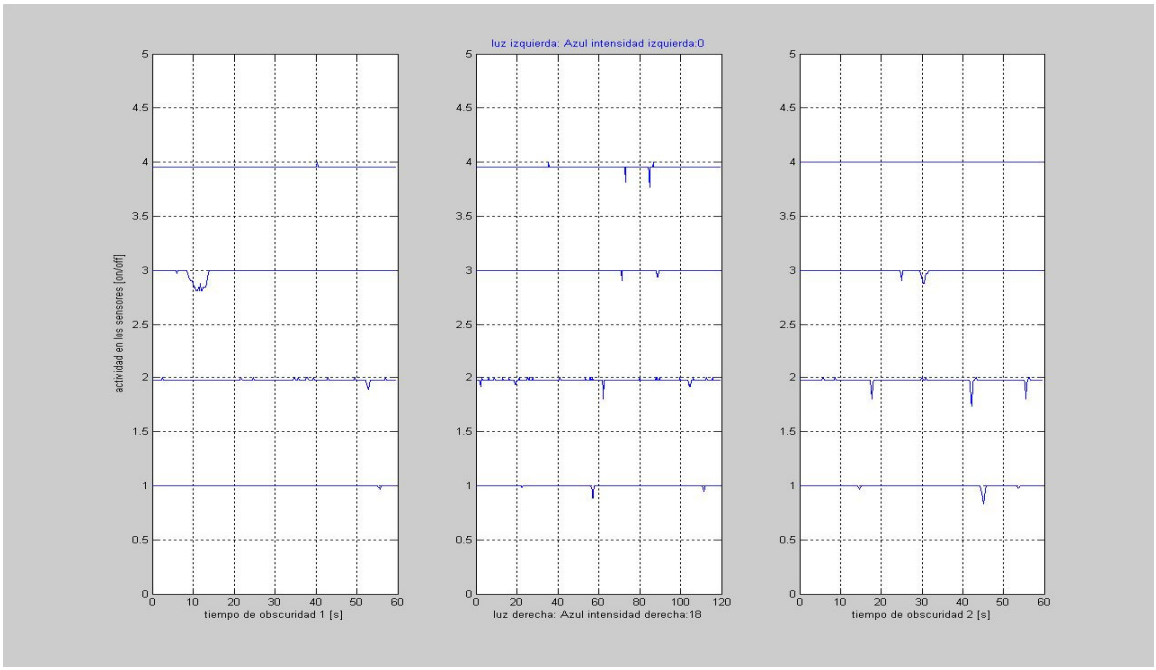


Contador:

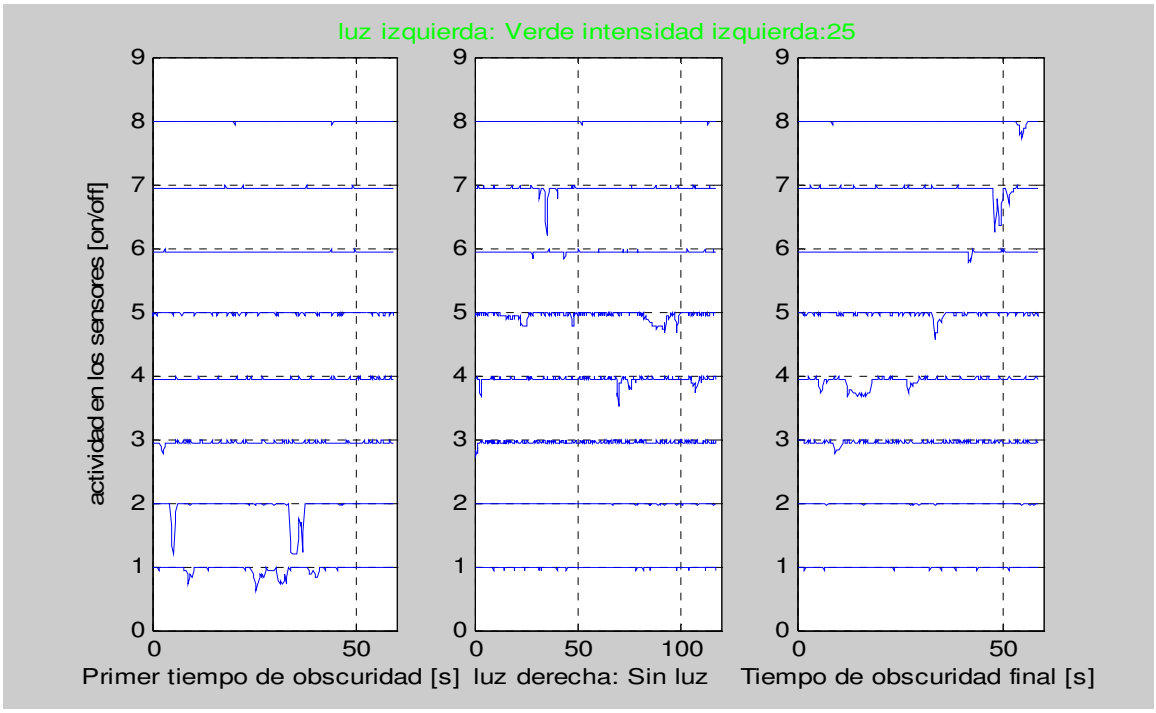
	azul izq		
mosca 1	obsc inicial	experimento	obsc final
sensor 1	3	2	5
sensor 2	1	6	11
sensor 3	0	3	3
sensor 4	2	4	0
sensor 5	0	3	4
sensor 6	12	2	2
sensor 7	12	3	2
sensor 8	0	6	0
promedio	3,75	3,625	3,375

En este experimento se aprecia que la mosca se aleja, se acerca y se vuelve a alejar del LED, se puede comparar con un experimento parecido realizado con la tesis anterior donde el comportamiento de la mosca es similar.

Experimento en dispositivo de tesis anterior con azul derecha intensidad luminosa de 18.66 Lm.



Mosca 2, verde izquierda, intensidad 25 (21.49 Lm)



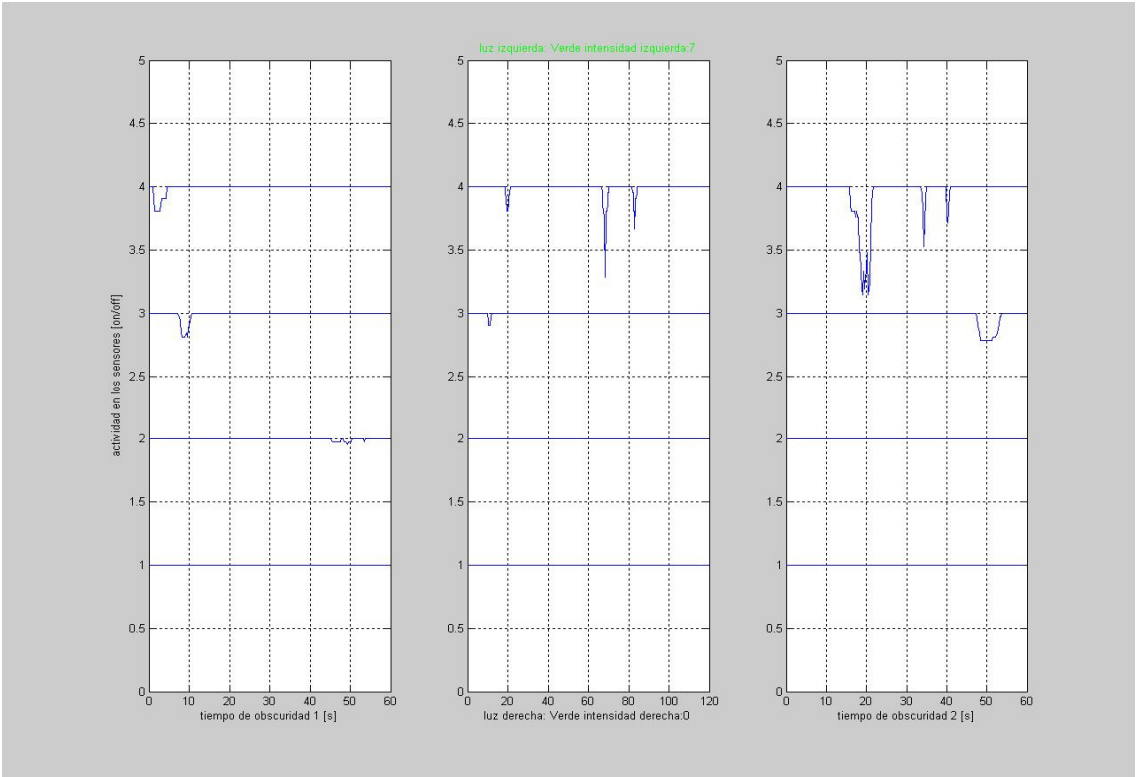
Contador:

mosca 2	verde izq obsc inicial	experimento	obsc final
sensor 1	5	0	0
sensor 2	7	0	0
sensor 3	0	0	1
sensor 4	0	7	5
sensor 5	0	5	2
sensor 6	0	0	1
sensor 7	0	4	6

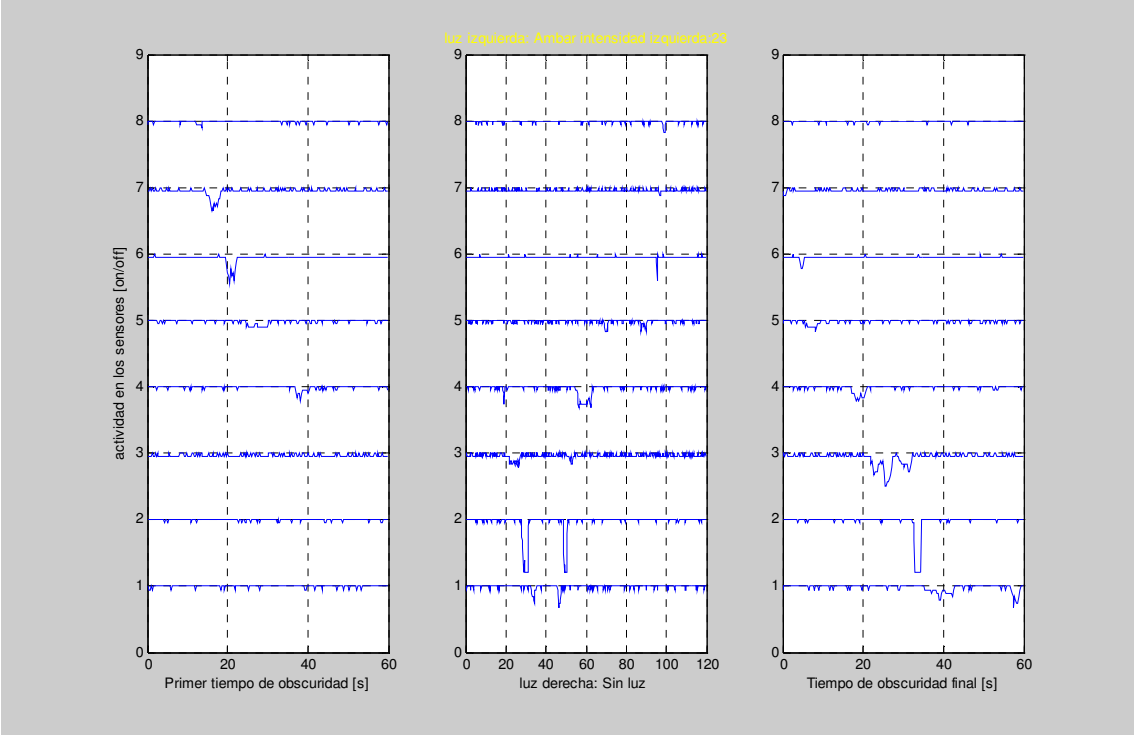
sensor 8	0	0	2
promedio	1,5	2	2,125

Se aprecia que la mosca se acerca al LED y luego se aleja, pero no totalmente, se mantiene a una intensidad luminosa entre 0.33 Lm y 0.46 Lm. Se puede comparar con el experimento realizado previamente donde se observa que la mosca se mantiene a una intensidad luminosa de aproximadamente 0.66 Lm.

Experimento en dispositivo de tesis anterior con verde izquierda, intensidad luminosa de .66 Lm.



Mosca3, ámbar izquierdo, intensidad 23 (20.04 Lm)

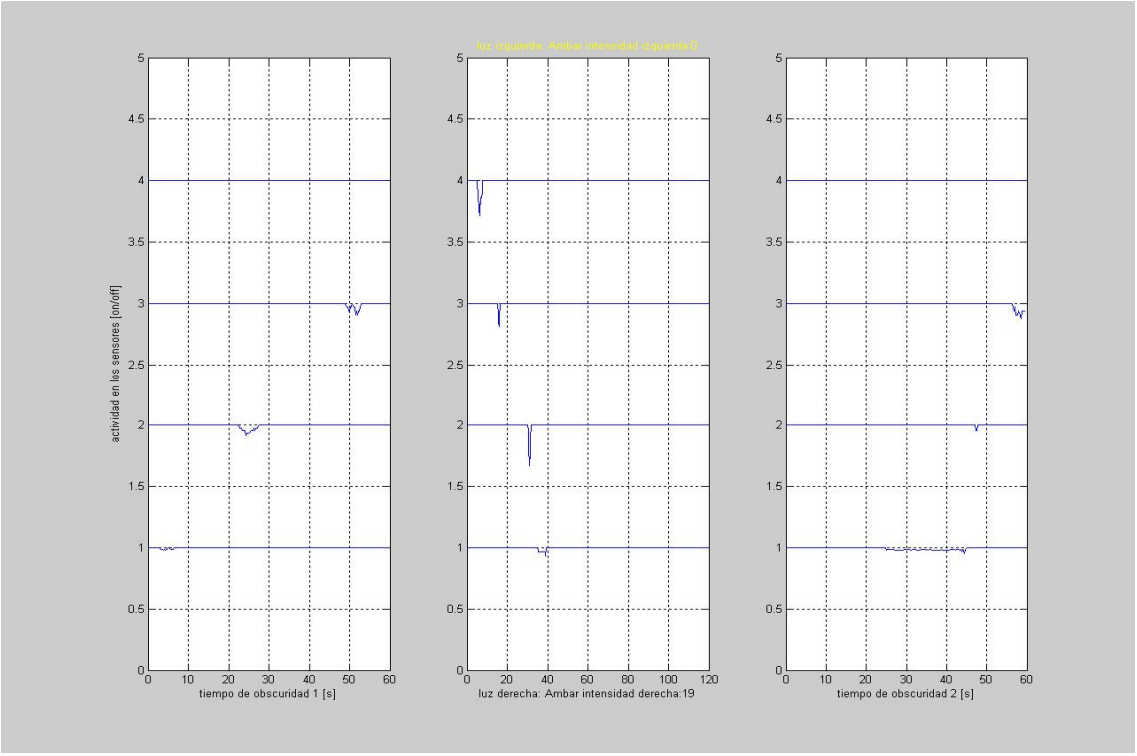


Contador:

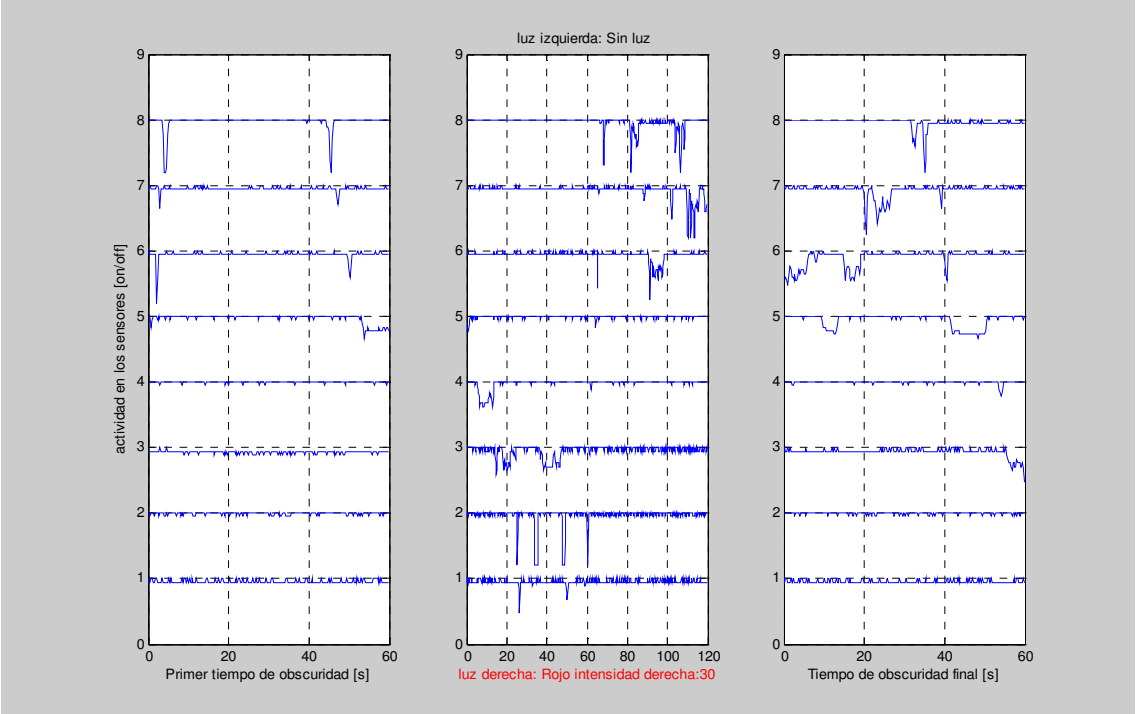
mosca 3	ámbar izq obsc inicial	experimento	obsc final
sensor 1	0	1	2
sensor 2	0	7	2
sensor 3	0	1	6
sensor 4	2	3	1
sensor 5	0	0	1
sensor 6	5	1	1
sensor 7	1	0	0
sensor 8	0	0	0
promedio	1	1,625	1,625

Se aprecia que la mosca se aleja y luego lentamente se acerca a la luz. Comparando con el experimento de la tesis anterior que la mosca se acercó a la luz y se mantuvo ahí, el comportamiento fue un poco distinto.

Experimento en dispositivo de tesis anterior con ámbar derecha, intensidad luminosa de 11 Lm.



Mosca 4, rojo derecha, intensidad 30 (21.64 Lm)



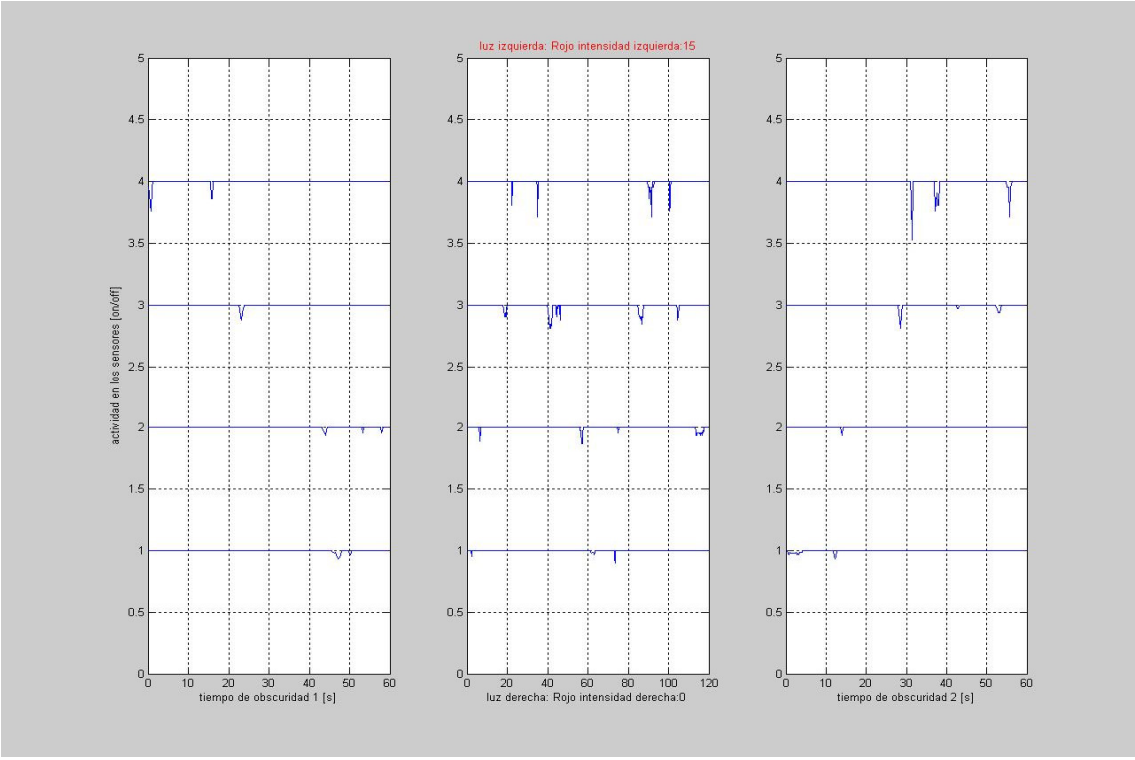
Contador:

	rojo der		
mosca 4	obsc inicial	experimento	obsc final

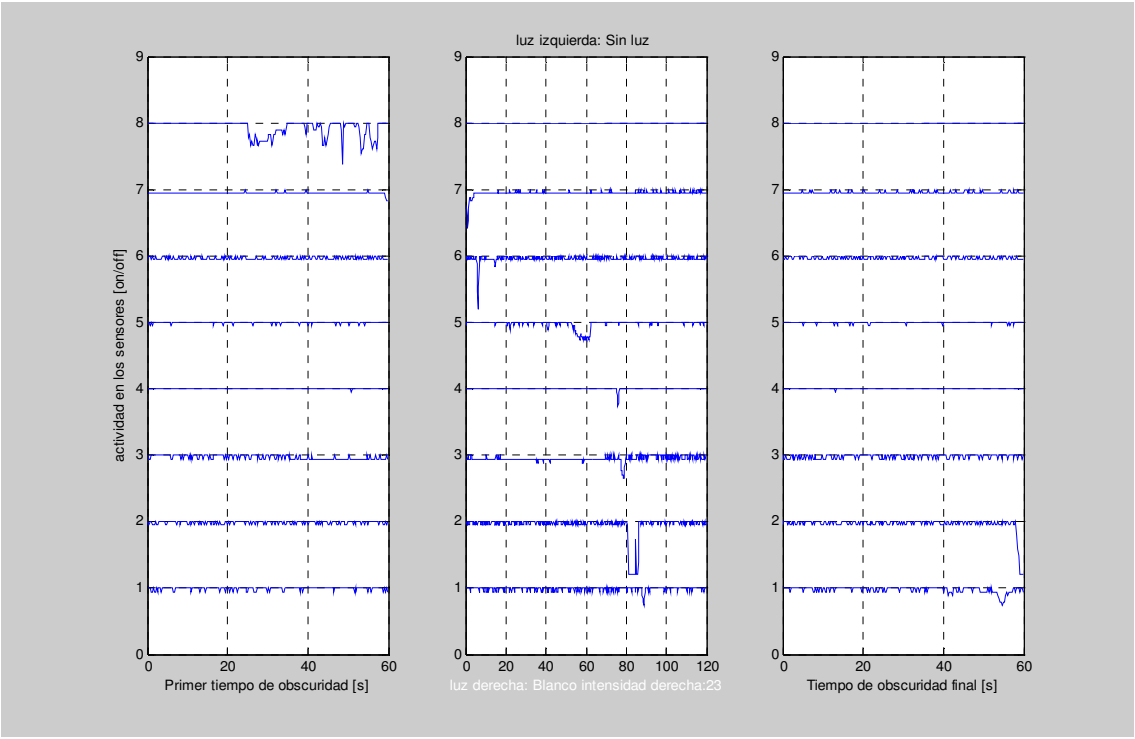
sensor 1	0	4	0
sensor 2	0	6	0
sensor 3	0	7	2
sensor 4	0	4	1
sensor 5	3	1	3
sensor 6	3	7	10
sensor 7	3	11	8
sensor 8	5	11	5
promedio	1,75	6,375	3,625

El comportamiento de esta mosca fue un poco extraño, se acercó a la luz y se mantuvo deambulando en el brazo de la cámara donde hay luz y después se alejó y se mantuvo en el lado del brazo donde hay menos luz. Comparando con la tesis anterior, la mosca fue más constante, primero se acercó, luego se alejó y se volvió a acercar a la luz.

Experimento en dispositivo de tesis anterior con rojo izquierda, intensidad luminosa de 11.33 Lm.



Mosca 5, blanco derecha, intensidad 23 (21.28 Lm)

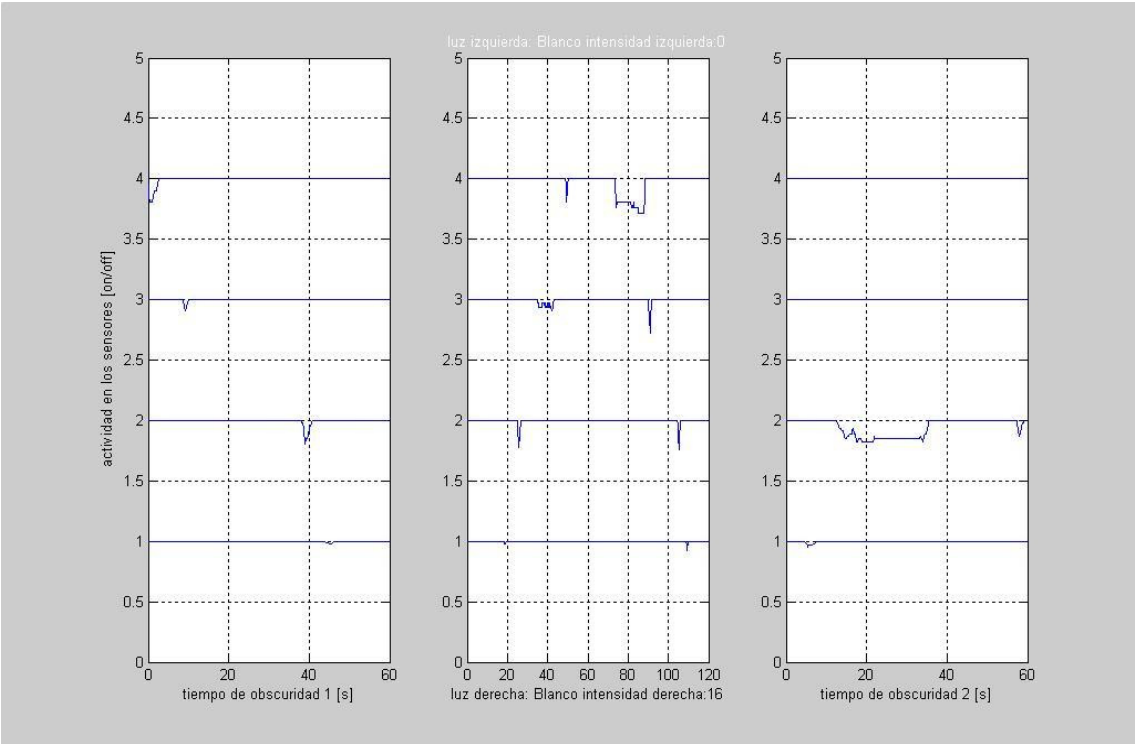


Contador:

mosca 5	blanco der obsc inicial	experimento	obsc final
sensor 1	0	1	1
sensor 2	0	3	4
sensor 3	0	1	0
sensor 4	0	1	0
sensor 5	0	0	0
sensor 6	0	2	0
sensor 7	0	2	0
sensor 8	15	0	0
promedio	1,875	1,25	0,625

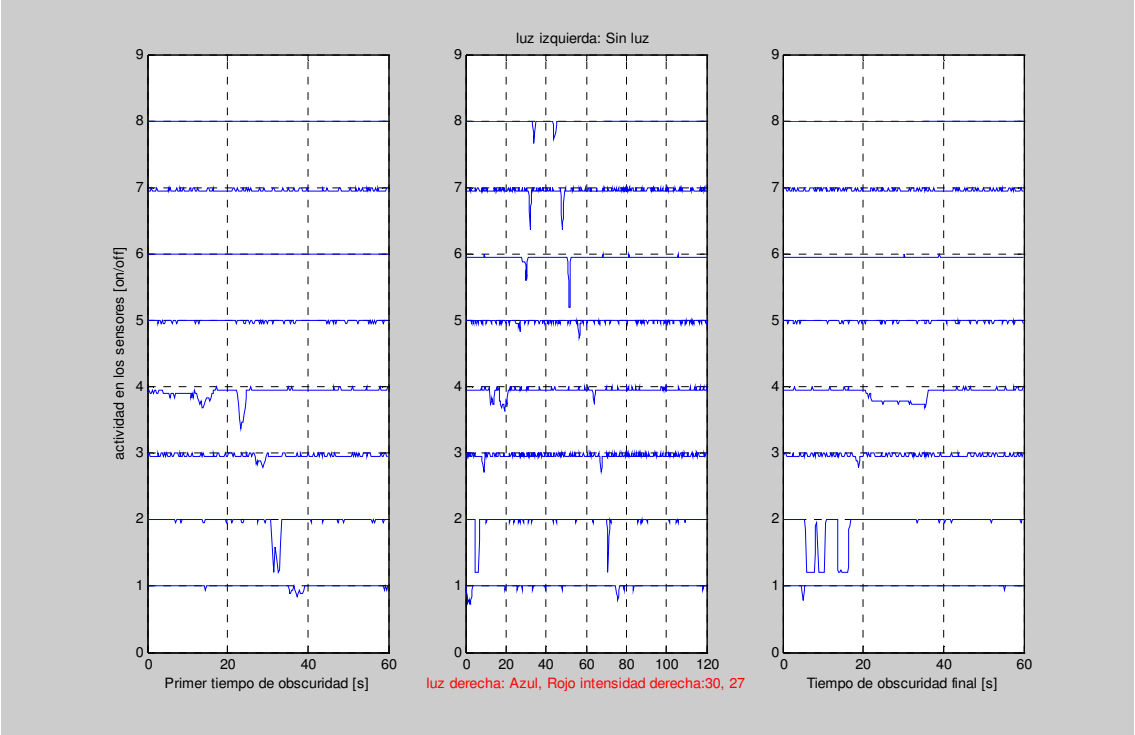
En este caso se aprecia que la mosca se acercó lentamente a la luz y ahí se mantuvo. Comparando con el experimento anterior que se alejó lentamente y después se volvió a acercar lentamente.

Experimento en dispositivo de tesis anterior con blanco izquierda, intensidad luminosa de 44.66 Lm.



Estos experimentos son los comparables con los experimentos realizados con el dispositivo original, claro que se puede observar que entran en acción los nuevos sensores implementados. Los siguientes experimentos son con combinaciones de colores al azar con intensidades iguales

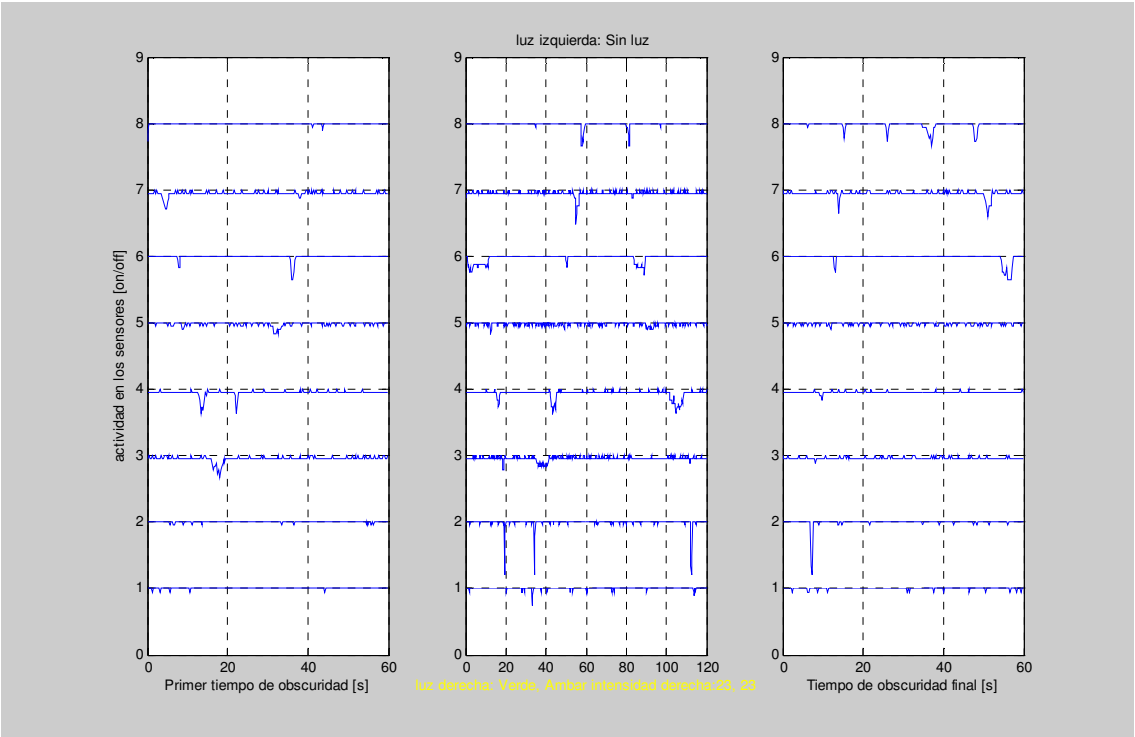
Mosca 1, azul-rojo, intensidades de 27 (20.53 Lm) y 30 (21.64 Lm) respectivamente.



Contador:	rojo-azul der		
mosca 1	obsc inicial	experimento	obsc final
sensor 1	0	3	2
sensor 2	6	3	5
sensor 3	1	2	1
sensor 4	4	3	0
sensor 5	0	2	0
sensor 6	0	4	0
sensor 7	0	5	0
sensor 8	0	3	0
promedio	1,375	3,125	1

Se aprecia que la mosca primero se aleja y después regresa y se mantiene donde está la luz.

Mosca 2, verde-ámbar, intensidades de 23 (21.49 Lm) y 23 (20.04 Lm) respectivamente.



Contador:	verde-am izq		
mosca 2	obsc inicial	experimento	obsc final
sensor 1	0	1	0
sensor 2	0	4	2
sensor 3	3	10	0
sensor 4	2	7	0
sensor 5	2	2	0
sensor 6	2	3	3
sensor 7	0	2	3
sensor 8	0	4	4
promedio	1,125	4,125	1,5

Se observa que la mosca se acerca a la luz, se aleja y después se vuelve a acercar.

5. Conclusiones

5.1. Conclusiones generales

El estudio de los estímulos de la mosca *Drosophila* es muy usado; sin embargo sus estudios en general son a simple vista, con este dispositivo es posible cuantificar su comportamiento al estimularla con luz. Éste es un principio para la cuantificación del comportamiento de las moscas evocado por un estímulo, sin embargo será posible estudiar el comportamiento con otro tipo de estímulos tales como temperatura, olor y sonido. Con este tipo de dispositivos será más preciso el estudio de éstas con fin educativo y científico. El propósito original de diseñar el presente dispositivo era como una herramienta didáctica a utilizarse con alumnos de bachillerato, sin embargo su alcance es mucho mayor.

El dispositivo es de precio medio, sin embargo podría disminuir su precio considerablemente al hacerlo en serie y si de alguna manera se pudiera realizar un microcontrolador con las especificaciones necesarias del dispositivo donde no sea necesario tener más de un microcontrolador realizando todas las operaciones.

5.2. Recomendaciones y posibles trabajos futuros

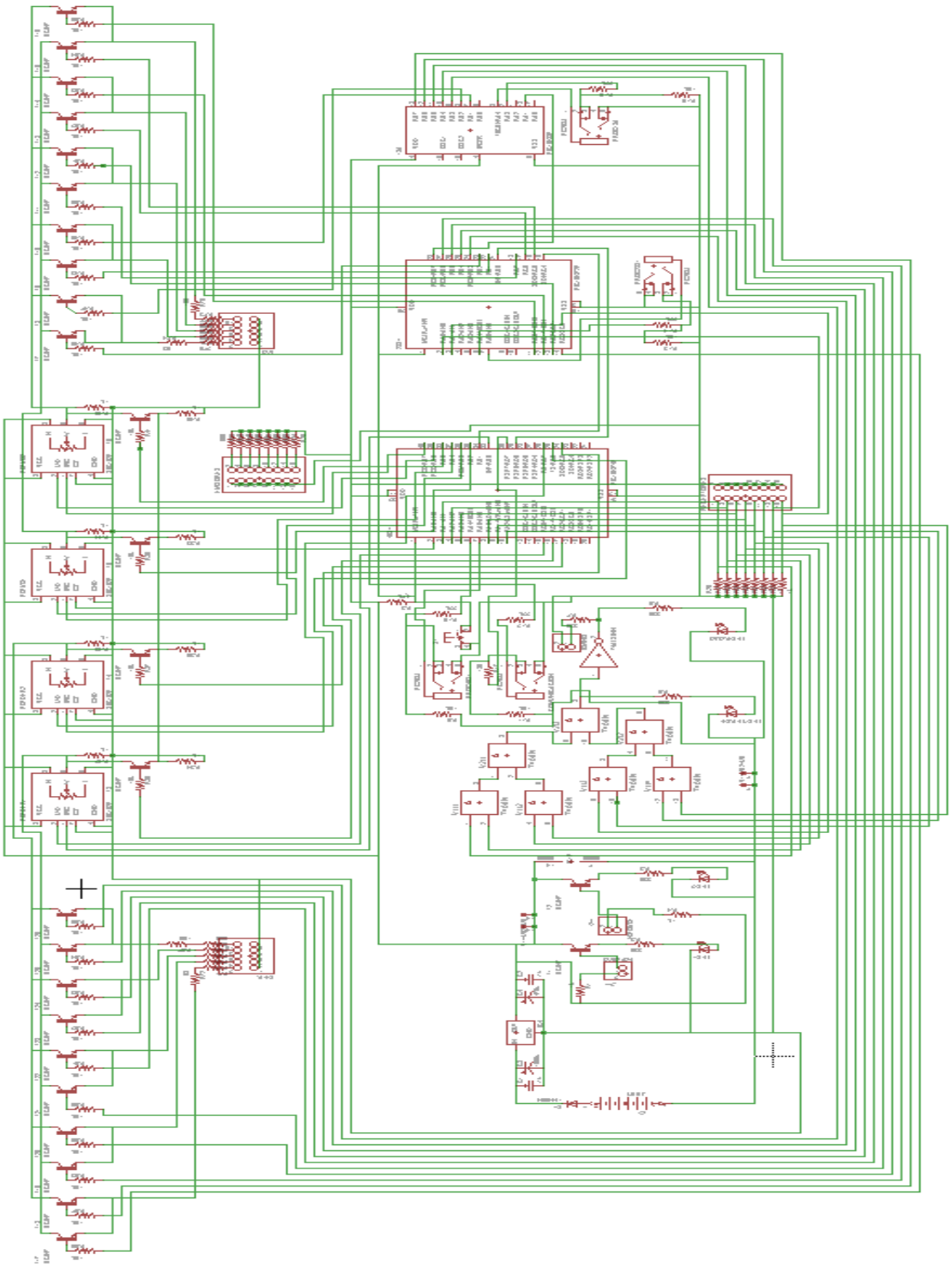
El rango de visión de la mosca, comprende del rojo al ultravioleta, así que una posible mejora sería adaptar un LED de tipo ultravioleta de cada lado de la cámara con un potenciómetro digital para poder variar la intensidad también. Una mejora también sería cambiar los sensores de infrarrojos por otro tipo, ya que la emisión de rayos infrarrojos es un tipo de ruido para la mosca. No sabemos si la mosca puede llegar a responder a estímulos infrarrojos y es por esto que es recomendable cambiar el tipo de sensores, podría ser de sonido, movimiento, etc.

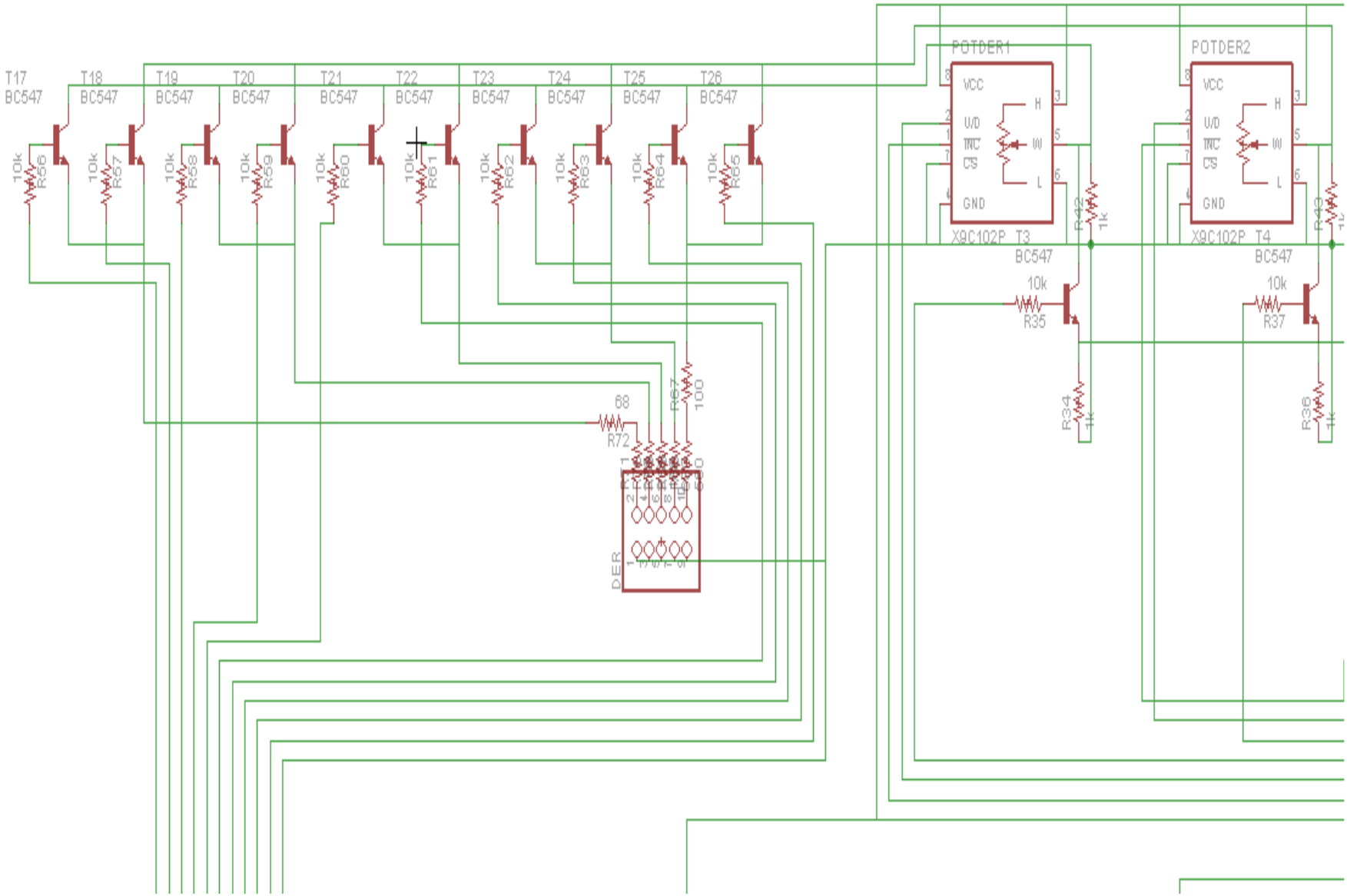
Como podemos observar, la interfaz con el usuario se realizó con un programa ya hecho como lo es Matlab. Una interfaz gráfica mejorada en diseño y realizada en otra plataforma de programación, es una mejora que debería realizarse. También sería un implemento necesario que el programa realice un estudio estadístico que sea desplegado en gráficas para saber en el instante el comportamiento de la mosca comparado con diferentes frecuencias y con diferentes moscas. También que se pueda observar el comportamiento de la mosca en el instante y no al final del experimento.

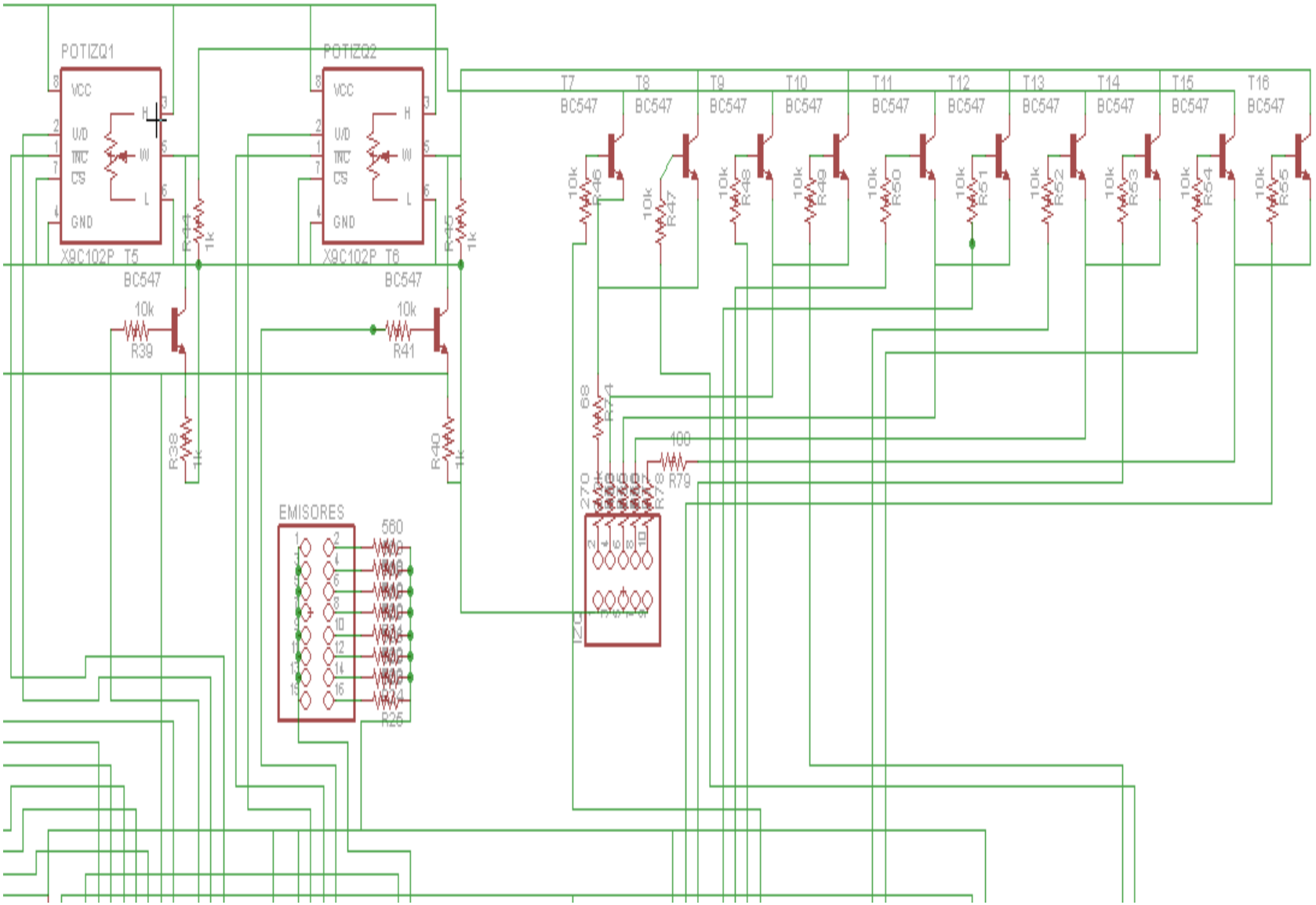
6. Apéndices

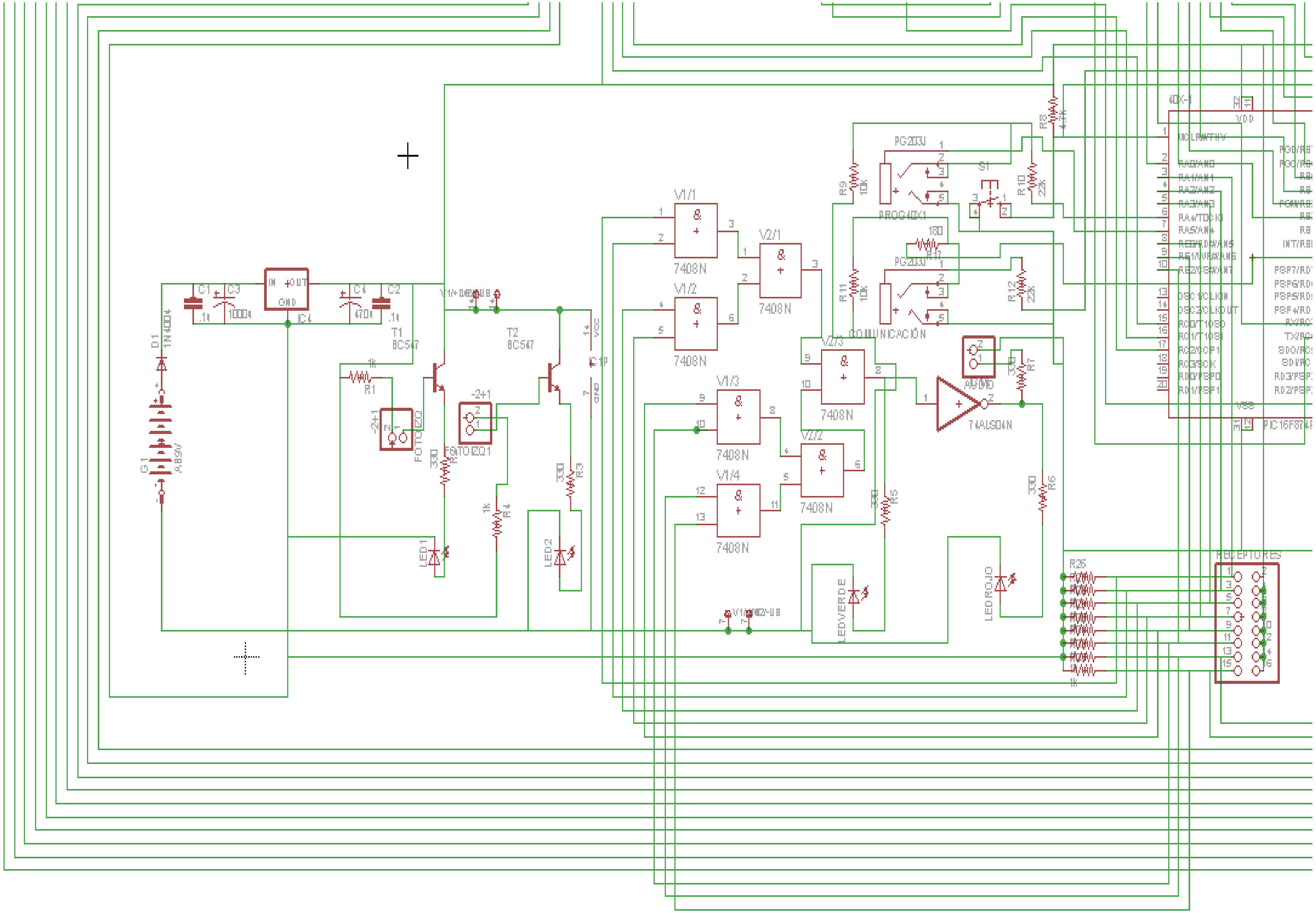
A. Arreglo Óptico

La primera imagen es el circuito completo, las siguientes cuatro son por partes.

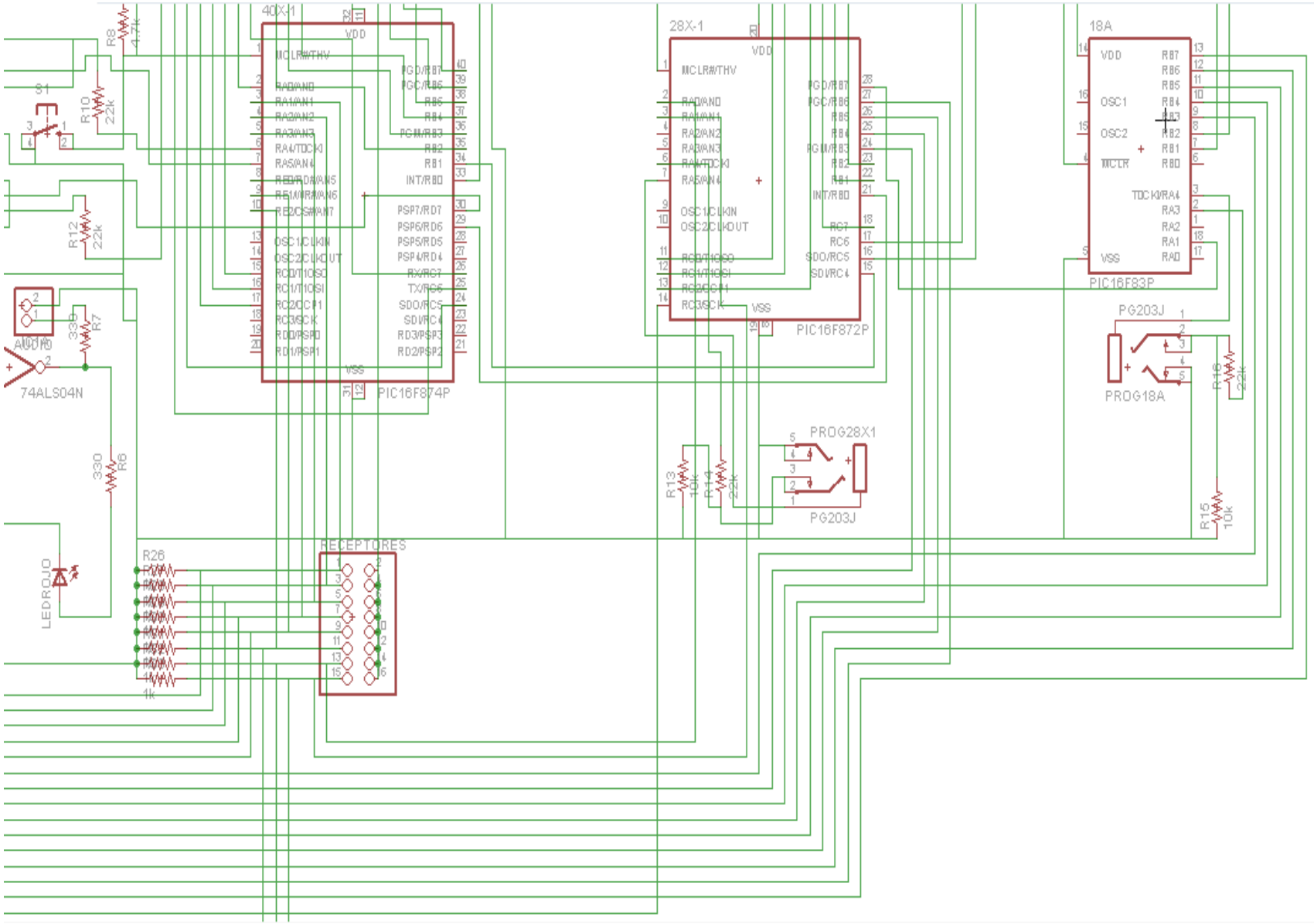



T17
BC547
T18
BC547
T19
BC547
T20
BC547
T21
BC547
T22
BC547
T23
BC547
T24
BC547
T25
BC547
T26
BC547
10k
R56
10k
R57
10k
R58
10k
R59
10k
R60
10k
R61
10k
R62
10k
R63
10k
R64
10k
R65
R67
100
68
R72
DER
POTDER1
VCC
U/D
INC
CS
GND
H
W
L
X9C102P T3
R42
1k
BC547
10k
R35
R34
1k
POTDER2
VCC
U/D
INC
CS
GND
H
W
L
X9C102P T4
R43
BC547
10k
R37
R36
1k


POTIZQ1
POTIZQ2
VCC
U/D
INC
CS
GND
H
W
L
X9C102P T5
X9C102P T6
R44 1k
R45 1k
BC547
10k
R39
R41
R38 1k
R40 1k
EMISORES
560
R25
T7 BC547
T8 BC547
T9 BC547
T10 BC547
T11 BC547
T12 BC547
T13 BC547
T14 BC547
T15 BC547
T16 BC547
R46
R47
R48
R49
R50
R51
R52
R53
R54
R55
68
R74
100
R79
270
R78
IZQ



7408N
74ALS04N
BC547
LED1
LED2
LEDVERDE
LEDROJO
1N4004
RECEPTORES
COMUNICACIÓN
PIC16F874


40X-1
PIC16F874P
28X-1
PIC16F872P
18A
PIC16F83P
74ALS04N
PROG18A
PROG28X1
PG203J
RECEPTORES
LEDROJO
R8
4.7k
R10
22k
R12
22k
R7
330
R6
330
R26
1k
R13
10k
R14
22k
R15
10k
R16
22k

B. Requerimientos y especificaciones del dispositivo

Requerimientos

Equipo de cómputo:

Sistema operativo: Win98/ME/2000/XP/Vista

Hardware:

Conexión puerto serie

Software:

MATLAB versión 6.5 o superiores

PICAXE® Programming Editor 5.2.0

Especificaciones

Las especificaciones del dispositivo son:

Intensidad relativa[1]

Rango dinámico: 0 – 30

Resolución: 1

Tiempo para obscuridad

Rango dinámico: 1-14400 [s] (4hrs)

Resolución: 1[s]

Tiempo para experimento

Rango dinámico: 1-14400 [s] (4hrs)

Resolución: 1[s]

Voltaje de alimentación[2]**:** 9V

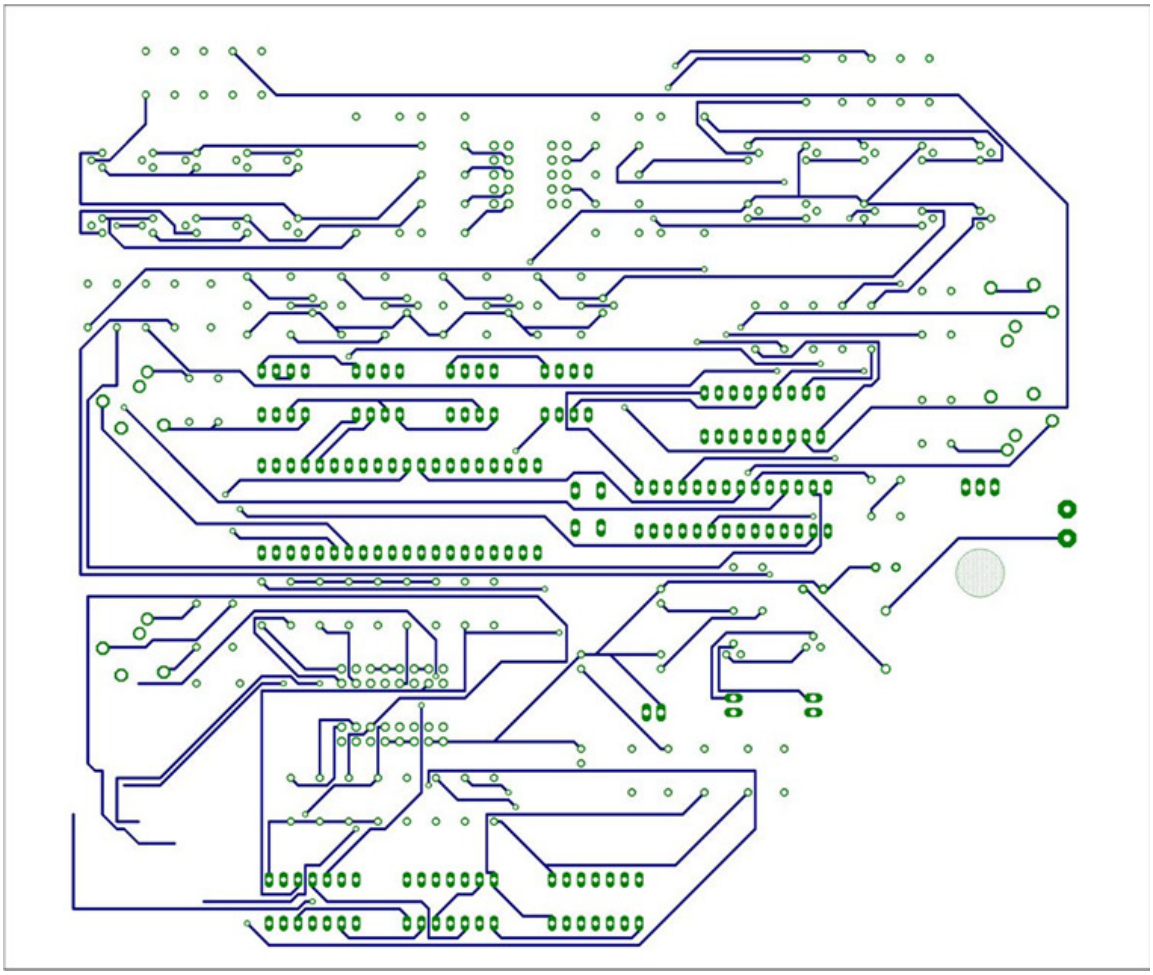
Consumo de corriente: 320 mA

[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 1h, en caso contrario se requiere conectar el dispositivo a un eliminador de voltaje fijo a 9V o bien una fuente de poder.

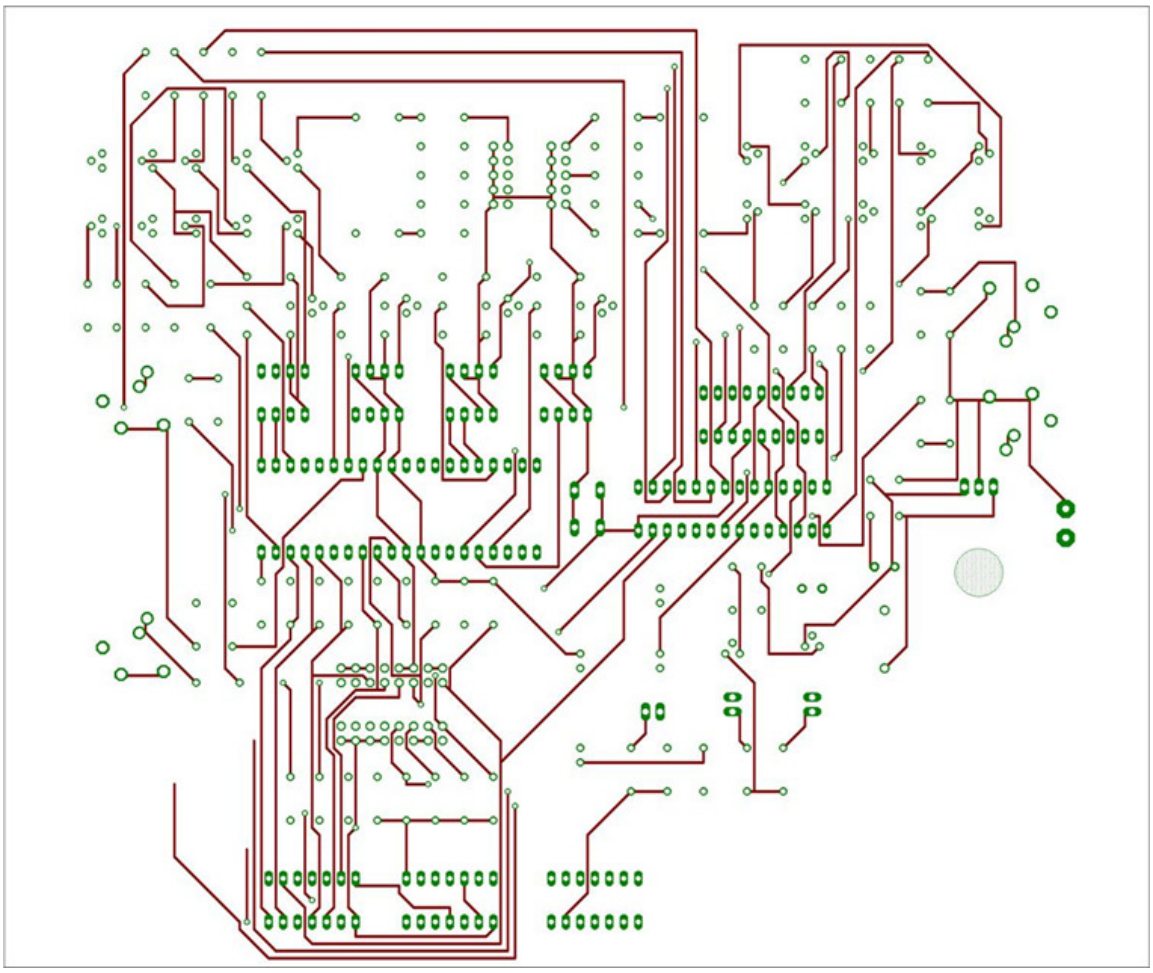
C. PCB’s

Se presentan los PCB’s para el circuito impreso en bicapa

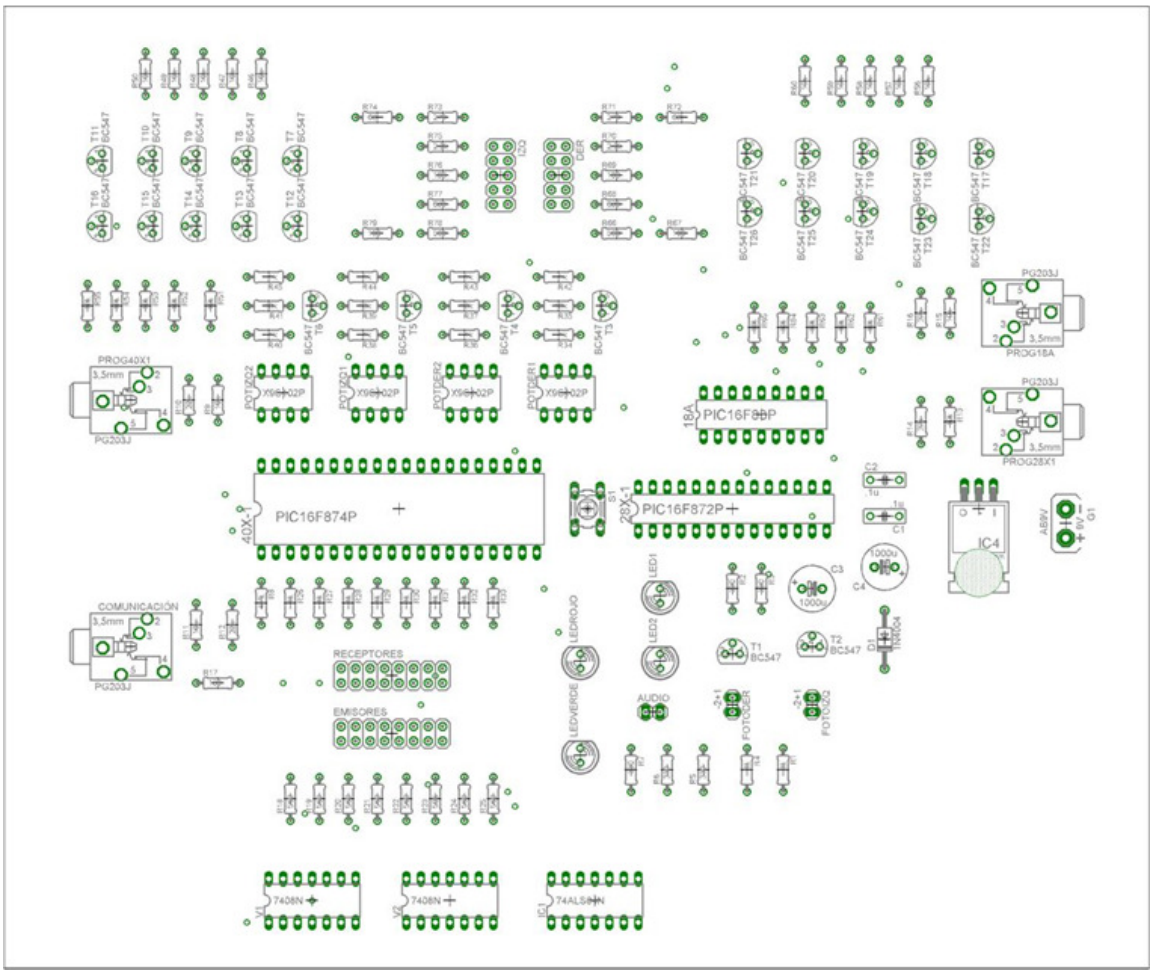
Capa inferior



Capa superior



Capa superior de componentes



D. Programa de los PIC

A continuación se muestran los programas desarrollados en lenguaje Basic, con los que fueron programados los PICs.

MAESTRO

```
let dirsc =%11111111
symbol valorder1=b2
symbol valorizq1=b3
symbol valorder2=b4
symbol valorizq2=b5

symbol colorder1=b6
symbol colorizq1=b7
```

```
symbol colorder2=b16
symbol colorizq2=b17
symbol duracion=w6
symbol minpot1=23
symbol minpot2=23
symbol minpot3=23
symbol minpot4=23

setfreq m8
let pins=0
low portc 0
low portc 1
low portc 2
low portc 3
low portc 4
low portc 5
low portc 6
low portc 7

main:

        let dirsc =%11111111
        low portc 6
        low portc 5
        serin 7,N2400,b9,b10,b11     'Adquiere el tiempo de 1a obscuridad

        let w0=b10*60
        let w7=b11*3600
        let w0=w0+b9+w7
        let w0=w0*4
        serin 7,N2400,b9,b10,b11     'Adquiere el tiempo de 2a obscuridad

        let w10=b10*60
        let w7=b11*3600
        let w10=w10+b9+w7
        let w10=w10*4

'Etapa de Selección de color
```

```
        serin 7,N2400,valorizq1,valorizq2,valorder1,valorder2,colorizq1,colorizq2,b14,colorder1,colorder2,b15 ,b9,b10,b11
        serout 1,N2400,(colorizq1,colorizq2,colorder1,colorder2)

        let w6=0
        let w6=b10*60
        let w11=b11*3600
        let w6=w6+b9+w11
        let w6=w6*4

'Etapa de control de intensidad

subirizq1:
        high 2
        high 3
        pulsout 4,500
        readadc 0, b9  'Mide valor del ADC0 del PIC correspondiente al número de pasos en el potenciómetro digital izquierdo 1

        if b9=minpot1 then incrementari1
        if b9<minpot1 then subirizq1
        if b9>minpot1 then bajarizq1
        goto subirizq1

subirizq2:
        low 2
        high 5
        high 6
        pulsout 7,500
        readadc 0, b9  'Mide valor del ADC0 del PIC correspondiente al número de pasos en el potenciómetro digital izquierdo 2

        if b9=minpot2 then incrementari2
        if b9<minpot2 then subirizq2
        if b9>minpot2 then bajarizq2
        goto subirizq2

bajarizq1:
        low 3
        pulsout 4,500
```

```
        readadc 0, b9  'Mide valor del ADC1 del PIC correspondiente al número de pasos en el potenciómetro digital izquierdo

        if b9=minpot1 then incrementari1
        if b9>minpot1 then bajarizq1
        if b9<minpot1 then subirizq1
        goto bajarizq1

bajarizq2:
        low 6
        pulsout 7,500
        readadc 0, b9  'Mide valor del ADC1 del PIC correspondiente al número de pasos en el potenciómetro digital izquierdo

        if b9=minpot2 then incrementari2
        'if b9>b14 and b9<b15 then incrementari2
        if b9>minpot2 then bajarizq2
        if b9<minpot2 then subirizq2
        goto bajarizq2

subirder1:
        low 5
        high portc 2
        high portc 1
        high portc 0
        pauseus 500
        low portc 0
        pauseus 500
        readadc 0, b9  'Mide valor del ADC0 del PIC correspondiente al número de pasos en el potenciómetro digital derecho

        if b9=minpot3 then incrementard1
        if b9<minpot3 then subirder1
        if b9>minpot3 then bajarder1
        goto subirder1

subirder2:
        low portc 2
        high portc 5
        high portc 6
        high portc 7
```

```
        pauseus 500
        low portc 7
        pauseus 500
        readadc 0, b9  'Mide valor del ADC0 del PIC correspondiente al número de pasos en el potenciómetro digital derecho

        if b9=minpot4 then incrementard2
        if b9<minpot4 then subirder2
        if b9>minpot4 then bajarder2
        goto subirder2

bajarder1:
        low portc 1
        high portc 0
        pauseus 500
        low portc 0
        pauseus 500
        readadc 0, b9  'Mide valor del ADC0 del PIC correspondiente al número de pasos en el potenciómetro digital derecho

        if b9=minpot3 then incrementard1
        if b9>minpot3 then bajarder1
        if b9<minpot3 then subirder1
        goto bajarder1

bajarder2:
        low portc 6
        high portc 7
        pauseus 500
        low portc 7
        pauseus 500
        readadc 0, b9  'Mide valor del ADC0 del PIC correspondiente al número de pasos en el potenciómetro digital derecho

        if b9=minpot4 then incrementard2
        'if b9>b14 and b9<b15 then incrementard2
        if b9>minpot4 then bajarder2
        if b9<minpot4 then subirder2
        goto bajarder2

incrementari1:
```

```
        let b9=0
        for b9=0 to valorizq1
                high 3
                pulsout 4,500
                readadc 0,b10

        next b9
        gosub subirizq2

incrementari2:
        let b9=0
        for b9=0 to valorizq2
                high 6
                pulsout 7,500
                readadc 0,b10

        next b9
        gosub subirder1

incrementard1:
        let b9=0
        for b9 =0 to valorder1
                high portc 1
                high portc 0
                pauseus 500
                low portc 0
                pauseus 500
                readadc 0,b10

        next b9
        gosub subirder2

incrementard2:
        let b9=0
        for b9=0 to valorder2
                high portc 6
                high portc 7
                pauseus 500
                low portc 7
                pauseus 500
                readadc 0,b10
```

```
        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:
        low portc 5
        gosub tiempoobs                 'Realiza la subrutina tiempo de obscuridad
        serout 1,N2400,(0)
        gosub tiempo
        gosub tiempoobs2
        goto main
```

'Monitoreo de la Mosca.

```
tiempoobs:
        serout 1,N2400,(w0)
        let w11=0
        for w11=0 to w0
                pause 390
                readadc 1, b19
                let b19=b19-128
                readadc 2, b9
                let b9=b9-128
                readadc 3, b10
                let b10=b10-128
                readadc 5, b11
                let b11=b11-128
                readadc 6,b24
                let b24=b24-128
                readadc 7,b25
                let b25=b25-128
                serout 1,N2400,(1)
                serin 6,N2400,b26
                'let b26=b26-128
                serin 6,N2400,b27
                'let b27=b27-128
```

```
            serout 0, N2400,(b19,b9,b10,b11,b24,b25,b26,b27) ‘envía el valor a la
computadora
        next w11
        return

tiempo:
        serin 6,N2400,(4)
        serout 1,N2400,(duracion)
        let w11=0
        for w11=0 to duracion
                pause 410
                readadc 1, b19
                let b19=b19-128
                readadc 2, b9
                let b9=b9-128
                readadc 3,b10
                let b10=b10-128
                readadc 5, b11
                let b11=b11-128
                readadc 6,b24
                let b24=b24-128
                readadc 7,b25
                let b25=b25-128
                serout 1,N2400,(2)
                serin 6,N2400,b26
                'let b26=b26-128
                serin 6,N2400,b27
                'let b27=b27-128
                serout 0, N2400,(b19,b9,b10,b11,b24,b25,b26,b27) ‘envía el valor a la
computadora
        next w11
        return

tiempoobs2:
        serout 1,N2400,(w10)
        let w11=0
        for w11=0 to w10
                pause 420
                readadc 1, b19
                let b19=b19-128
                readadc 2, b9
```

```
            let b9=b9-128
            readadc 3, b10
            let b10=b10-128
            readadc 5, b11
            let b11=b11-128
            readadc 6,b24
            let b24=b24-128
            readadc 7,b25
            let b25=b25-128
            serout 1,N2400,(3)
            serin 6,N2400,b26
            'let b26=b26-128
            serin 6,N2400,b27
            'let b27=b27-128
            serout 0, N2400,(b19,b9,b10,b11,b24,b25,b26,b27) ‘envía el valor a la computadora
        next w11
        return
```

ESCLAVO 1

```
let dirsc=%11101111
symbol colorder28x1=b0
symbol colorder28x2=b1
symbol colorizq28x1=b2
symbol colorizq28x2=b3
symbol yd=6
symbol gd=5
symbol bd=4
symbol wd=3
symbol ri=2
symbol yi=1

setfreq m8

main:
        let pins=%00000000
        low portc 7
        low portc 6
        low portc 5
        low portc 3
```

```
        low portc 2
        low portc 1
        low portc 0

        serin 4,N2400,colorizq28x1,colorizq28x2,colorder28x1,colorder28x2
tiempoobs:
        serin 4,N2400,w3
        let w2=0
        for w2=0 to w3
                serin 4,N2400,(1)
                readadc 0,b8
                b8=b8-128
                serout 0,N2400,(b8)
                readadc 1,b9
                b9=b9-128
                serout 0,N2400,(b9)
        next w2
        goto colores

colores:

        serin 4,N2400,(0)
        serout 7,N2400,(colorizq28x1,colorizq28x2,colorder28x1,colorder28x2)

        if colorizq28x1="b" and colorizq28x2="b" and colorder28x1="b" and
colorder28x2="b" then azulazul
        if colorizq28x1="b" and colorizq28x2="b" and colorder28x1="r" and
colorder28x2="r" then azulrojo
        if colorizq28x1="b" and colorizq28x2="b" and colorder28x1="g" and
colorder28x2="g" then azulverde
        if colorizq28x1="b" and colorizq28x2="b" and colorder28x1="w" and
colorder28x2="w" then azulblanco
        if colorizq28x1="b" and colorizq28x2="b" and colorder28x1="y" and
colorder28x2="y" then azulambar
        if colorizq28x1="r" and colorizq28x2="r" and colorder28x1="b" and
colorder28x2="b" then rojoazul
        if colorizq28x1="r" and colorizq28x2="r" and colorder28x1="r" and
colorder28x2="r" then rojorojo
        if colorizq28x1="r" and colorizq28x2="r" and colorder28x1="g" and
colorder28x2="g" then rojoverde
```

if colorizq28x1="r" and colorizq28x2="r" and colorder28x1="w" and colorder28x2="w" then rojoblanco
if colorizq28x1="r" and colorizq28x2="r" and colorder28x1="y" and colorder28x2="y" then rojoambar
if colorizq28x1="g" and colorizq28x2="g" and colorder28x1="b" and colorder28x2="b" then verdeazul
if colorizq28x1="g" and colorizq28x2="g" and colorder28x1="r" and colorder28x2="r" then verderojo
if colorizq28x1="g" and colorizq28x2="g" and colorder28x1="g" and colorder28x2="g" then verdeverde
if colorizq28x1="g" and colorizq28x2="g" and colorder28x1="w" and colorder28x2="w" then verdeblanco
if colorizq28x1="g" and colorizq28x2="g" and colorder28x1="y" and colorder28x2="y" then verdeambar
if colorizq28x1="w" and colorizq28x2="w" and colorder28x1="b" and colorder28x2="b" then blancoazul
if colorizq28x1="w" and colorizq28x2="w" and colorder28x1="r" and colorder28x2="r" then blancorojo
if colorizq28x1="w" and colorizq28x2="w" and colorder28x1="g" and colorder28x2="g" then blancoverde
if colorizq28x1="w" and colorizq28x2="w" and colorder28x1="w" and colorder28x2="w" then blancoblanco
if colorizq28x1="w" and colorizq28x2="w" and colorder28x1="y" and colorder28x2="y" then blancoambar
if colorizq28x1="y" and colorizq28x2="y" and colorder28x1="b" and colorder28x2="b" then ambarazul
if colorizq28x1="y" and colorizq28x2="y" and colorder28x1="r" and colorder28x2="r" then ambarrojo
if colorizq28x1="y" and colorizq28x2="y" and colorder28x1="g" and colorder28x2="g" then ambarverde
if colorizq28x1="y" and colorizq28x2="y" and colorder28x1="w" and colorder28x2="w" then ambarblanco
if colorizq28x1="y" and colorizq28x2="y" and colorder28x1="y" and colorder28x2="y" then ambarambar

if colorizq28x1="b" and colorizq28x2="r" and colorder28x1="b" and colorder28x2="r" then brbr
if colorizq28x1="b" and colorizq28x2="g" and colorder28x1="b" and colorder28x2="r" then bgbr
if colorizq28x1="b" and colorizq28x2="w" and colorder28x1="b" and colorder28x2="r" then bwbr

if colorizq28x1="b" and colorizq28x2="y" and colorder28x1="b" and colorder28x2="r" then bybr
if colorizq28x1="b" and colorizq28x2="r" and colorder28x1="b" and colorder28x2="g" then brbg
if colorizq28x1="b" and colorizq28x2="g" and colorder28x1="b" and colorder28x2="g" then bgbg
if colorizq28x1="b" and colorizq28x2="w" and colorder28x1="b" and colorder28x2="g" then bwbg
if colorizq28x1="b" and colorizq28x2="y" and colorder28x1="b" and colorder28x2="g" then bybg
if colorizq28x1="b" and colorizq28x2="r" and colorder28x1="b" and colorder28x2="w" then brbw
if colorizq28x1="b" and colorizq28x2="g" and colorder28x1="b" and colorder28x2="w" then bgbw
if colorizq28x1="b" and colorizq28x2="w" and colorder28x1="b" and colorder28x2="w" then bwbw
if colorizq28x1="b" and colorizq28x2="y" and colorder28x1="b" and colorder28x2="w" then bybw
if colorizq28x1="b" and colorizq28x2="r" and colorder28x1="b" and colorder28x2="y" then brby
if colorizq28x1="b" and colorizq28x2="g" and colorder28x1="b" and colorder28x2="y" then bgby
if colorizq28x1="b" and colorizq28x2="w" and colorder28x1="b" and colorder28x2="y" then bwby
if colorizq28x1="b" and colorizq28x2="y" and colorder28x1="b" and colorder28x2="y" then byby

if colorizq28x1="b" and colorizq28x2="r" and colorder28x1="r" and colorder28x2="g" then brrg
if colorizq28x1="b" and colorizq28x2="g" and colorder28x1="r" and colorder28x2="g" then bgrg
if colorizq28x1="b" and colorizq28x2="w" and colorder28x1="r" and colorder28x2="g" then bwrg
if colorizq28x1="b" and colorizq28x2="y" and colorder28x1="r" and colorder28x2="g" then byrg
if colorizq28x1="b" and colorizq28x2="r" and colorder28x1="r" and colorder28x2="w" then brrw
if colorizq28x1="b" and colorizq28x2="g" and colorder28x1="r" and colorder28x2="w" then bgrw
if colorizq28x1="b" and colorizq28x2="w" and colorder28x1="r" and colorder28x2="w" then bwrw

if colorizq28x1="b" and colorizq28x2="y" and colorder28x1="r" and colorder28x2="w" then byrw
if colorizq28x1="b" and colorizq28x2="r" and colorder28x1="r" and colorder28x2="y" then brry
if colorizq28x1="b" and colorizq28x2="g" and colorder28x1="r" and colorder28x2="y" then bgry
if colorizq28x1="b" and colorizq28x2="w" and colorder28x1="r" and colorder28x2="y" then bwry
if colorizq28x1="b" and colorizq28x2="y" and colorder28x1="r" and colorder28x2="y" then byry

if colorizq28x1="b" and colorizq28x2="r" and colorder28x1="g" and colorder28x2="w" then brgw
if colorizq28x1="b" and colorizq28x2="g" and colorder28x1="g" and colorder28x2="w" then bggw
if colorizq28x1="b" and colorizq28x2="w" and colorder28x1="g" and colorder28x2="w" then bwgw
if colorizq28x1="b" and colorizq28x2="y" and colorder28x1="g" and colorder28x2="w" then bygw

if colorizq28x1="b" and colorizq28x2="r" and colorder28x1="g" and colorder28x2="y" then brgy
if colorizq28x1="b" and colorizq28x2="g" and colorder28x1="g" and colorder28x2="y" then bggy
if colorizq28x1="b" and colorizq28x2="w" and colorder28x1="g" and colorder28x2="y" then bwgy
if colorizq28x1="b" and colorizq28x2="y" and colorder28x1="g" and colorder28x2="y" then bygy

if colorizq28x1="b" and colorizq28x2="r" and colorder28x1="w" and colorder28x2="y" then brwy
if colorizq28x1="b" and colorizq28x2="g" and colorder28x1="w" and colorder28x2="y" then bgwy
if colorizq28x1="b" and colorizq28x2="w" and colorder28x1="w" and colorder28x2="y" then bwwy
if colorizq28x1="b" and colorizq28x2="y" and colorder28x1="w" and colorder28x2="y" then bywy

if colorizq28x1="r" and colorizq28x2="g" and colorder28x1="b" and colorder28x2="r" then rgbr

```
        if colorizq28x1="r" and colorizq28x2="w" and colorder28x1="b" and
colorder28x2="r" then rwbr
        if colorizq28x1="r" and colorizq28x2="y" and colorder28x1="b" and
colorder28x2="r" then rybr
        if colorizq28x1="r" and colorizq28x2="g" and colorder28x1="b" and
colorder28x2="g" then rgbg
        if colorizq28x1="r" and colorizq28x2="w" and colorder28x1="b" and
colorder28x2="g" then rwbg
        if colorizq28x1="r" and colorizq28x2="y" and colorder28x1="b" and
colorder28x2="g" then rybg
        if colorizq28x1="r" and colorizq28x2="g" and colorder28x1="b" and
colorder28x2="w" then rgbw
        if colorizq28x1="r" and colorizq28x2="w" and colorder28x1="b" and
colorder28x2="w" then rwbw
        if colorizq28x1="r" and colorizq28x2="y" and colorder28x1="b" and
colorder28x2="w" then rybw
        if colorizq28x1="r" and colorizq28x2="g" and colorder28x1="b" and
colorder28x2="y" then rgby
        if colorizq28x1="r" and colorizq28x2="w" and colorder28x1="b" and
colorder28x2="y" then rwby
        if colorizq28x1="r" and colorizq28x2="y" and colorder28x1="b" and
colorder28x2="y" then ryby

        if colorizq28x1="r" and colorizq28x2="g" and colorder28x1="r" and
colorder28x2="g" then rgrg
        if colorizq28x1="r" and colorizq28x2="w" and colorder28x1="r" and
colorder28x2="g" then rwrg
        if colorizq28x1="r" and colorizq28x2="y" and colorder28x1="r" and
colorder28x2="g" then ryrg
        if colorizq28x1="r" and colorizq28x2="g" and colorder28x1="r" and
colorder28x2="w" then rgrw
        if colorizq28x1="r" and colorizq28x2="w" and colorder28x1="r" and
colorder28x2="w" then rwrw
        if colorizq28x1="r" and colorizq28x2="y" and colorder28x1="r" and
colorder28x2="w" then ryrw
        if colorizq28x1="r" and colorizq28x2="g" and colorder28x1="r" and
colorder28x2="y" then rgry
        if colorizq28x1="r" and colorizq28x2="w" and colorder28x1="r" and
colorder28x2="y" then rwry
        if colorizq28x1="r" and colorizq28x2="y" and colorder28x1="r" and
colorder28x2="y" then ryry
```

if colorizq28x1="r" and colorizq28x2="g" and colorder28x1="g" and colorder28x2="w" then rggw
if colorizq28x1="r" and colorizq28x2="w" and colorder28x1="g" and colorder28x2="w" then rwgw
if colorizq28x1="r" and colorizq28x2="y" and colorder28x1="g" and colorder28x2="w" then rygw
if colorizq28x1="r" and colorizq28x2="g" and colorder28x1="g" and colorder28x2="y" then rggy
if colorizq28x1="r" and colorizq28x2="w" and colorder28x1="g" and colorder28x2="y" then rwgy
if colorizq28x1="r" and colorizq28x2="y" and colorder28x1="g" and colorder28x2="y" then rygy

if colorizq28x1="r" and colorizq28x2="g" and colorder28x1="w" and colorder28x2="y" then rgwy
if colorizq28x1="r" and colorizq28x2="w" and colorder28x1="w" and colorder28x2="y" then rwwy
if colorizq28x1="r" and colorizq28x2="y" and colorder28x1="w" and colorder28x2="y" then rywy

if colorizq28x1="g" and colorizq28x2="w" and colorder28x1="b" and colorder28x2="r" then gwbr
if colorizq28x1="g" and colorizq28x2="y" and colorder28x1="b" and colorder28x2="r" then gybr
if colorizq28x1="g" and colorizq28x2="w" and colorder28x1="b" and colorder28x2="g" then gwbg
if colorizq28x1="g" and colorizq28x2="y" and colorder28x1="b" and colorder28x2="g" then gybg
if colorizq28x1="g" and colorizq28x2="w" and colorder28x1="b" and colorder28x2="w" then gwbw
if colorizq28x1="g" and colorizq28x2="y" and colorder28x1="b" and colorder28x2="w" then gybw
if colorizq28x1="g" and colorizq28x2="w" and colorder28x1="b" and colorder28x2="y" then gwby
if colorizq28x1="g" and colorizq28x2="y" and colorder28x1="b" and colorder28x2="y" then gyby

if colorizq28x1="g" and colorizq28x2="w" and colorder28x1="r" and colorder28x2="g" then gwrg

if colorizq28x1="g" and colorizq28x2="y" and colorder28x1="r" and colorder28x2="g" then gyrg
if colorizq28x1="g" and colorizq28x2="w" and colorder28x1="r" and colorder28x2="w" then gwrw
if colorizq28x1="g" and colorizq28x2="y" and colorder28x1="r" and colorder28x2="w" then gyrw
if colorizq28x1="g" and colorizq28x2="w" and colorder28x1="r" and colorder28x2="y" then gwry
if colorizq28x1="g" and colorizq28x2="y" and colorder28x1="r" and colorder28x2="y" then gyry

if colorizq28x1="g" and colorizq28x2="w" and colorder28x1="g" and colorder28x2="w" then gwgw
if colorizq28x1="g" and colorizq28x2="y" and colorder28x1="g" and colorder28x2="w" then gygw
if colorizq28x1="g" and colorizq28x2="w" and colorder28x1="g" and colorder28x2="y" then gwgy
if colorizq28x1="g" and colorizq28x2="y" and colorder28x1="g" and colorder28x2="y" then gygy

if colorizq28x1="g" and colorizq28x2="w" and colorder28x1="w" and colorder28x2="y" then gwwy
if colorizq28x1="g" and colorizq28x2="y" and colorder28x1="w" and colorder28x2="y" then gywy

if colorizq28x1="w" and colorizq28x2="y" and colorder28x1="b" and colorder28x2="r" then wybr
if colorizq28x1="w" and colorizq28x2="y" and colorder28x1="b" and colorder28x2="g" then wybg
if colorizq28x1="w" and colorizq28x2="y" and colorder28x1="b" and colorder28x2="w" then wybw
if colorizq28x1="w" and colorizq28x2="y" and colorder28x1="b" and colorder28x2="y" then wyby

if colorizq28x1="w" and colorizq28x2="y" and colorder28x1="r" and colorder28x2="g" then wyrg
if colorizq28x1="w" and colorizq28x2="y" and colorder28x1="r" and colorder28x2="w" then wyrw
if colorizq28x1="w" and colorizq28x2="y" and colorder28x1="r" and colorder28x2="y" then wyry

```
        if colorizq28x1="w" and colorizq28x2="y" and colorder28x1="g" and
colorder28x2="w" then wygw
        if colorizq28x1="w" and colorizq28x2="y" and colorder28x1="g" and
colorder28x2="y" then wygy

        if colorizq28x1="w" and colorizq28x2="y" and colorder28x1="w" and
colorder28x2="y" then wywy
azulazul:
        high portc 6
        high bd
        goto tiempo

azulrojo:
        high portc 6
        high portc 3
        goto tiempo

azulverde:
        high portc 6
        high gd
        goto tiempo

azulblanco:
        high portc 6
        high wd
        goto tiempo

azulambar:
        high portc 6
        high yd
        goto tiempo


rojoazul:
        high ri
        high bd
        goto tiempo

rojorojo:
        high ri
        high portc 3
        goto tiempo

rojoverde:
        high ri
        high gd
        goto tiempo

rojoblanco:
        high ri
        high wd
        goto tiempo

rojoambar:
        high ri
        high yd
        goto tiempo

verdeazul:
        high portc 7
        high bd
        goto tiempo

verderojo:
        high portc 7
        high portc 3
        goto tiempo

verdeverde:
        high portc 7
        high gd
        goto tiempo

verdeblanco:
        high portc 7
```

```
        high wd
        goto tiempo

verdeambar:
        high portc 7
        high yd
        goto tiempo

blancoazul:
        high portc 5
        high bd
        goto tiempo


blancorojo:
        high portc 5
        high portc 3
        goto tiempo

blancoverde:
        high portc 5
        high gd
        goto tiempo

blancoblanco:
        high portc 5
        high wd
        goto tiempo

blancoambar:
        high portc 5
        high yd
        goto tiempo

ambarazul:
        high yi
        high bd
        goto tiempo

ambarrojo:
        high yi
        high portc 3
        goto tiempo

ambarverde:
        high yi
        high gd
        goto tiempo

ambarblanco:
        high yi
        high wd
        goto tiempo

ambarambar:
        high yi
        high yd
        goto tiempo
brbr:
        high portc 6
        high bd
        goto tiempo
bgbr:
        high portc 6
        high portc 2
        high bd
        goto tiempo
bwbr:
        high portc 6
        high portc 0
        high bd
        goto tiempo
bybr:
        high portc 6
        high bd
        goto tiempo
brbg:
        high portc 6
        high bd
        high gd
        goto tiempo
bgbg:
```

```
        high portc 6
        high portc 2
        high bd
        goto tiempo
bwbg:
        high portc 6
        high portc 0
        high bd
        goto tiempo
bybg:
        high portc 6
        high bd
        goto tiempo
brbw:
        high portc 6

        high bd
        goto tiempo
bgbw:
        high portc 6
        high portc 2
        high bd
        goto tiempo
bwbw:
        high portc 6
        high portc 0
        high bd
        goto tiempo
bybw:
        high portc 6
        high bd
        goto tiempo
brby:
        high portc 6
        high bd
        goto tiempo
bgby:
        high portc 6
        high portc 2
        high bd
        goto tiempo
bwby:
        high portc 6
        high portc 0
        high bd
        goto tiempo
byby:
        high portc 6
        high bd
        goto tiempo
brrg:
        high portc 6
        high portc 3
        goto tiempo
bgrg:
        high portc 6
        high portc 2
        high portc 3
        goto tiempo
bwrg:
        high portc 6
        high portc 0
        high portc 3
        goto tiempo
byrg:
        high portc 6
        high portc 3
        goto tiempo
brrw:
        high portc 6
        high portc 3
        goto tiempo
bgrw:
        high portc 6
        high portc 2
        high portc 3
        goto tiempo
bwrw:
        high portc 6
        high portc 0
        high portc 3
        goto tiempo
```

```
byrw:
        high portc 6
        high portc 3
        goto tiempo
brry:
        high portc 6
        high portc 3
        goto tiempo
bgry:
        high portc 6
        high portc 2
        high portc 3
        goto tiempo
bwry:
        high portc 6
        high portc 0
        high portc 3

        goto tiempo
byry:
        high portc 6
        high portc 3
        goto tiempo
brgw:
        high portc 6
        high gd
        goto tiempo
bggw:
        high portc 6
        high portc 2
        high gd
        goto tiempo
bwgw:
        high portc 6
        high portc 0
        high gd
        goto tiempo
bygw:
        high portc 6
        high gd
        goto tiempo
brgy:
        high portc 6
        high gd
        goto tiempo
bggy:
        high portc 6
        high portc 2
        high gd
        goto tiempo
bwgy:
        high portc 6
        high portc 0
        high gd
        goto tiempo
bygy:
        high portc 6
        high gd
        goto tiempo
brwy:
        high portc 6
        high wd
        goto tiempo
bgwy:
        high portc 6
        high portc 2
        high wd
        goto tiempo
bwwy:
        high portc 6
        high portc 0
        high wd
        goto tiempo
bywy:
        high portc 6
        high wd
        goto tiempo
rgbr:
        high ri
        high portc 2
        high bd
        goto tiempo
```

```
rwbr:
        high ri
        high portc 0
        high bd
        goto tiempo
rybr:
        high ri
        high bd
        goto tiempo
rgbg:
        high ri
        high portc 2
        high bd
        goto tiempo
rwbg:
        high ri
        high portc 0
        high bd
        goto tiempo
rybg:
        high ri
        high bd
        goto tiempo
rgbw:
        high ri
        high portc 2
        high bd
        goto tiempo
rwbw:
        high ri
        high portc 0
        high bd
        goto tiempo
rybw:
        high ri
        high bd
        goto tiempo
rgby:
        high ri
        high portc 2
        high bd
        goto tiempo
rwby:
        high ri
        high portc 0
        high bd
        goto tiempo
ryby:
        high ri
        high bd
        goto tiempo
rgrg:
        high ri
        high portc 2
        high portc 3

        goto tiempo
rwrg:
        high ri
        high portc 0
        high portc 3
        goto tiempo
ryrg:
        high ri
        high portc 3
        goto tiempo
rgrw:
        high ri
        high portc 2
        high portc 3
        goto tiempo
rwrw:
        high ri
        high portc 0
        high portc 3
        goto tiempo
ryrw:
        high ri
        high portc 3
        goto tiempo
rgry:
        high ri
```

```
        high portc 2
        high portc 3
        goto tiempo
rwry:
        high ri
        high portc 0
        high portc 3
        goto tiempo
ryry:
        high ri
        high portc 3
        goto tiempo
rggw:
        high ri
        high portc 2
        high gd
        goto tiempo
rwgw:
        high ri
        high portc 0
        high gd
        goto tiempo
rygw:
        high ri
        high gd
        goto tiempo
rggy:
        high ri
        high portc 2
        high gd
        goto tiempo
rwgy:
        high ri
        high portc 0
        high gd
        goto tiempo
rygy:
        high ri
        high gd
        goto tiempo
rgwy:
        high ri
        high portc 2
        high wd
        goto tiempo
rwwy:
        high ri
        high portc 0
        high wd
        goto tiempo
rywy:
        high ri
        high wd
        goto tiempo
gwbr:
        high portc 7
        high portc 0
        high bd
        goto tiempo
gybr:
        high portc 7
        high bd
        goto tiempo
gwbg:
        high portc 7
        high portc 0
        high bd
        goto tiempo
gybg:
        high portc 7
        high bd
        goto tiempo
gwbw:
        high portc 7
        high portc 0
        high bd
        goto tiempo
gybw:
        high portc 7
        high bd
        goto tiempo
gwby:
```

```
        high portc 7
        high portc 0
        high bd
        goto tiempo
gyby:
        high portc 7
        high bd
        goto tiempo
gwrg:
        high portc 7
        high portc 0
        high portc 3
        goto tiempo
gyrg:
        high portc 7

        high portc 3
        goto tiempo
gwrw:
        high portc 7
        high portc 0
        high portc 3
        goto tiempo
gyrw:
        high portc 7
        high portc 3
        goto tiempo
gwry:
        high portc 7
        high portc 0
        high portc 3
        goto tiempo
gyry:
        high portc 7
        high portc 3
        goto tiempo
gwgw:
        high portc 7
        high portc 0
        high gd
        goto tiempo
gygw:
        high portc 7
        high gd
        goto tiempo
gwgy:
        high portc 7
        high portc 0
        high gd
        goto tiempo
gygy:
        high portc 7
        high gd
        goto tiempo
gwwy:
        high portc 7
        high portc 0
        high wd
        goto tiempo
gywy:
        high portc 7
        high wd
        goto tiempo
wybr:
        high portc 5
        high bd
        goto tiempo
wybg:
        high portc 5
        high bd
        goto tiempo
wybw:
        high portc 5
        high bd
        goto tiempo
wyby:
        high portc 5
        high bd
        goto tiempo
wyrg:
        high portc 5
        high portc 3
```

```
        goto tiempo
wyrw:
        high portc 5
        high portc 3
        goto tiempo
wyry:
        high portc 5
        high portc 3
        goto tiempo
wygw:
        high portc 5

tiempo:
        serout 0,N2400,(4)
        serin 4,N2400,w5
        let w2=0
        for w2=0 to w5
                serin 4,N2400,(2)
                readadc 0,b8
                b8=b8-128
                serout 0,N2400,(b8)
                readadc 1,b9
                b9=b9-128
                serout 0,N2400,(b9)
                next w2
        goto tiempoobs2
tiempoobs2:
        let pins=%00000000
        low portc 7
        low portc 6
        low portc 5
        low portc 3
        low portc 2
        low portc 1
        low portc 0
        serout 7,N2400,(1)
        serin 4,N2400,w6
        let w2=0
        for w2=0 to w6
                serin 4,N2400,(3)
                readadc 0,b8
                high gd
                goto tiempo
wygy:
        high portc 5
        high gd
        goto tiempo
wywy:
        high portc 5
        high wd
        goto tiempo
```

```
            b8=b8-128
            serout 0,N2400,(b8)
            readadc 1,b9
            b9=b9-128
            serout 0,N2400,(b9)
            next w2
        goto main
```

ESCLAVO 2

```
setfreq m8
symbol ri2=2
symbol wd2=3
symbol yi2=1
symbol bd2=4
symbol rd2=7
symbol yd2=6
symbol gd2=5

main:
let pins=0
serin 1,N2400,b0,b1,b2,b3

if b0="b" and b1="r"  then br
if b0!="y" and b1="y" then nyy
if b0!="r" and b1="r" then redizq
if b0!="y" and b1="y" then yellowizq
if b3="b" and b2!="b" then blue
if b3="r" and b2!="r" then red
if b3="g" and b2!="g" then green
if b3="w" and b2!="w" then white
if b3="y" and b2!="y" then yellow

goto tiempo
br:
if b2="b" and b3="r" then rr
if b2!="g" and b3="g" then rg
if b2!="w" and b3="w" then rw
if b2!="y" and b3="y" then ry
goto tiempo
```

```
nyy:
if b2!="b" and b3="b" then yb
if b2!="r" and b3="r" then yr
if b2!="g" and b3="g" then yg
if b2!="w" and b3="w" then yw
if b3="y" then yy
goto tiempo

blue:
        high bd2
        goto tiempo
red:
        high rd2
        goto tiempo
green:
        high gd2
        goto tiempo
white:
        high wd2
        goto tiempo
yellow:
        high yd2
        goto tiempo
rr:
        high ri2
        high rd2
        goto tiempo
rg:
        high ri2
        high gd2
        goto tiempo
rw:
        high ri2
        high wd2
        goto tiempo
ry:
        high ri2
        high yd2
        goto tiempo
yb:
        high yi2
        high bd2
        goto tiempo
yr:
        high yi2
        high rd2
        goto tiempo
yg:
        high yi2
        high gd2
        goto tiempo
yw:
        high yi2
        high wd2
        goto tiempo
yy:
        high yi2
        high yd2
        goto tiempo
redizq:
        high ri2
        goto tiempo
yellowizq:
        high yi2
        goto tiempo
```

```
tiempo:
        serin 1,N2400,(1)
        goto main
```

E. Hojas de datos

Transistor bipolar npn

FAIRCHILD
SEMICONDUCTOR®

BC546/547/548/549/550

Switching and Applications

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

TO-92
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	80	V
	: BC547/550	50	V
	: BC548/549	30	V
V_{CEO}	Collector-Emitter Voltage : BC546	65	V
	: BC547/550	45	V
	: BC548/549	30	V
V_{EBO}	Emitter-Base Voltage : BC546/547	6	V
	: BC548/549/550	5	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		90	250	mV
		I_C=100mA, I_B=5mA		200	600	mV
V_{BE} (sat)	Base-Emitter Saturation Voltage	I_C=10mA, I_B=0.5mA		700		mV
		I_C=100mA, I_B=5mA		900		mV
V_{BE} (on)	Base-Emitter On Voltage	V_{CE}=5V, I_C=2mA	580	660	700	mV
		V_{CE}=5V, I_C=10mA			720	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	V_{CE}=5V, I_C=200μA		2	10	dB
	: BC549/550	f=1KHz, R_G=2KΩ		1.2	4	dB
	: BC549	V_{CE}=5V, I_C=200μA		1.4	4	dB
	: BC550	R_G=2KΩ, f=30~15000MHz		1.4	3	dB

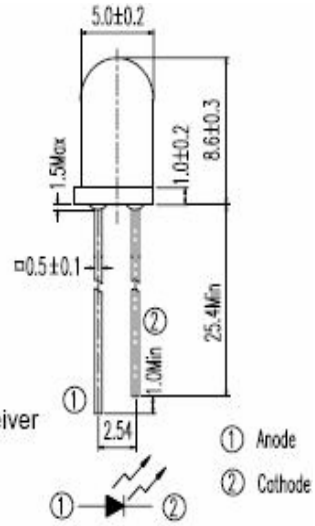
h_{FE} Classification

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

Rev. A2, August 2002

LED emisor de infrarrojos

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

Fototransistor (Receptor de infrarrojos)

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

Terminal Voltages ±5V, 100 Taps

XICOR

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

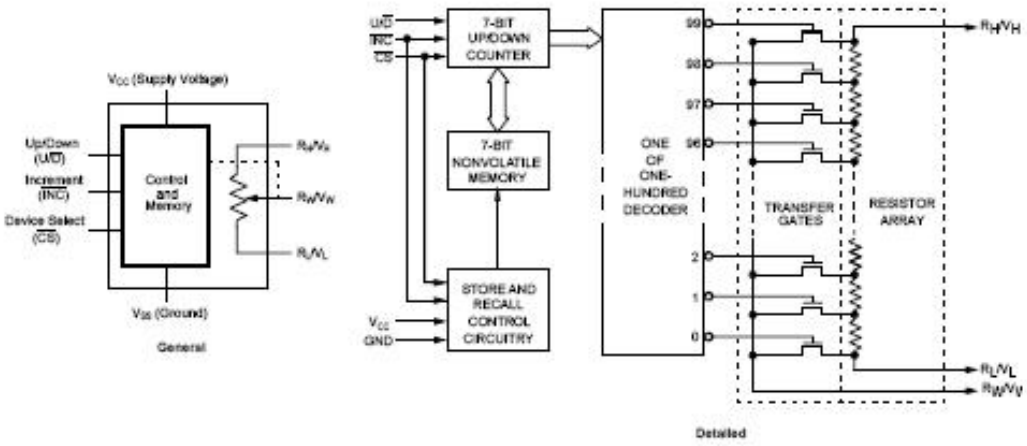
The X9Cxxx are Xicor digitally-controlled (XDCP) potentiometers. The device consists of a resistor array, wiper switches, a control section, and nonvolatile memory. The wiper position is controlled by a three-wire interface.

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

3883-2.4 2/12/99 T2/C0/D0 SH

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.

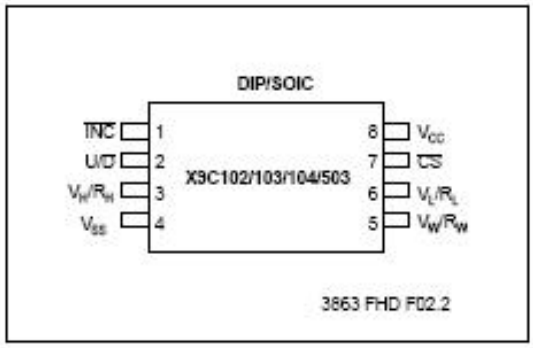
Increment ($\overline{INC}$)

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



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	MI(3)	$V_{W(n)(actual)} - V_{W(n)(expected)}$
	Relative Linearity(2)	–0.2		+0.2	MI(3)	$V_{W(n+1)(actual)} - [V_{W(n)} + MI]$
	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 = $[V_{W(n)(actual)} - V_{W(n)(expected)}] = \pm 1$ MI Maximum.
(2) Relative Linearity is a measure of the error in step size between taps = $V_{W(n+1)} - [V_{W(n)} + MI] = +0.2$ MI.
(3) 1 MI = 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.

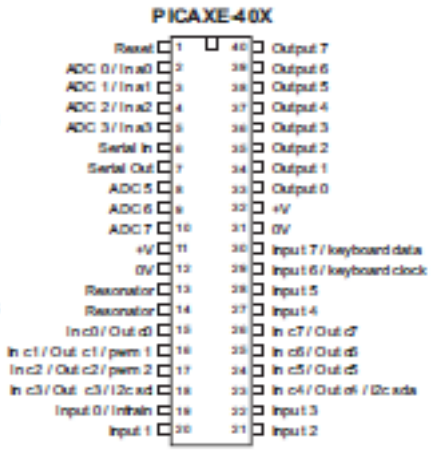
Patigrama PICAXE 40X-1

Appendix F – Configuring PICAXE-40X /40X1 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

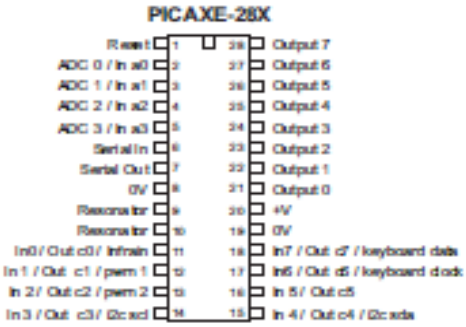
Leg	Default Function	Special Function
19	input 0	infrared (input)
20	input 1	
21	input 2	
22	input 3	
27	input 4	
28	input 5	
29	input 6	keyboard clock (input)
30	input 7	keyboard data (input)

Patigrama PICAXE 28X-1

Appendix E – Configuring PICAXE-28X / 28X1 Input-Output Pins

To provide greater flexibility, the input / output pin configiration of the PICAXE-28X can be varied by the user.
The default power up settings are the same as the other PICAXE-28 parts (8 in, 8 out, 4 analogue).

PORTA (legs 2 to 5) provide 4 analogue inputs (default) or up to 4 digital inputs.
PORTB (leg 21 to 28) provide 8 fixed outputs.
PORTC (leg 11 to 18) provide 8 digital inputs (default) or up to 8 outputs.



This gives a maximum of 12 digital inputs, 16 outputs and 4 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 Functions

PORTB pins are fixed as outputs and cannot be altered.

PORTC Functions

Leg	Default	Second Function	Special Function
11	input 0	output portc 0	infrared (input)
12	input 1	output portc 1	pwm 1 (output)
13	input 2	output portc 2	pwm 2 (output)
14	input 3	output portc 3	i2c scl clock (input)
15	input 4	output portc 4	i2c sda data (input)
16	input 5	output portc 5	
17	input 6	output portc 6	keyboard clock (input)
18	input 7	output portc 7	keyboard data (input)

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

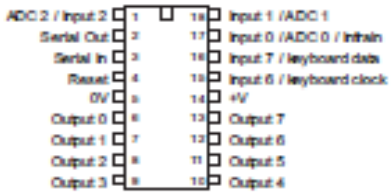
Patigrama PICAXE 18A

PICAXE-18/18A/18X Pinout and Circuit

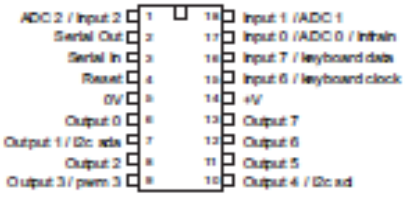
The pinout diagrams for the 18 pin devices are as follows:

PICAXE-18

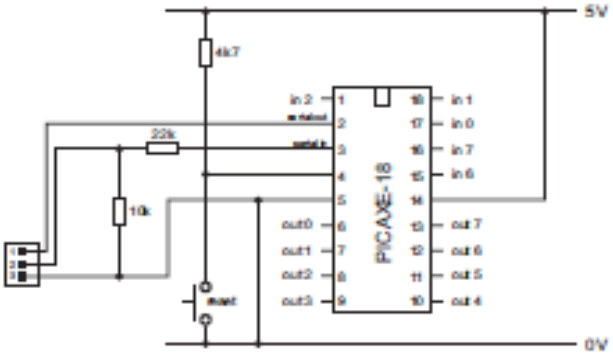
PICAXE-18A



PICAXE-18X



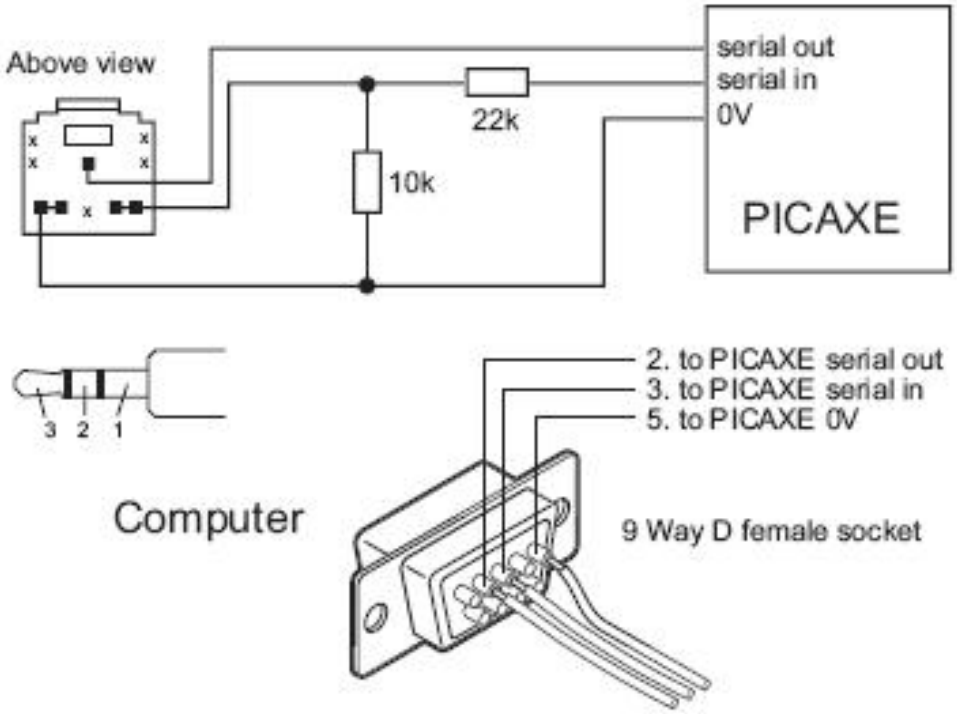
The minimum operating circuit for the 18 pin devices is:



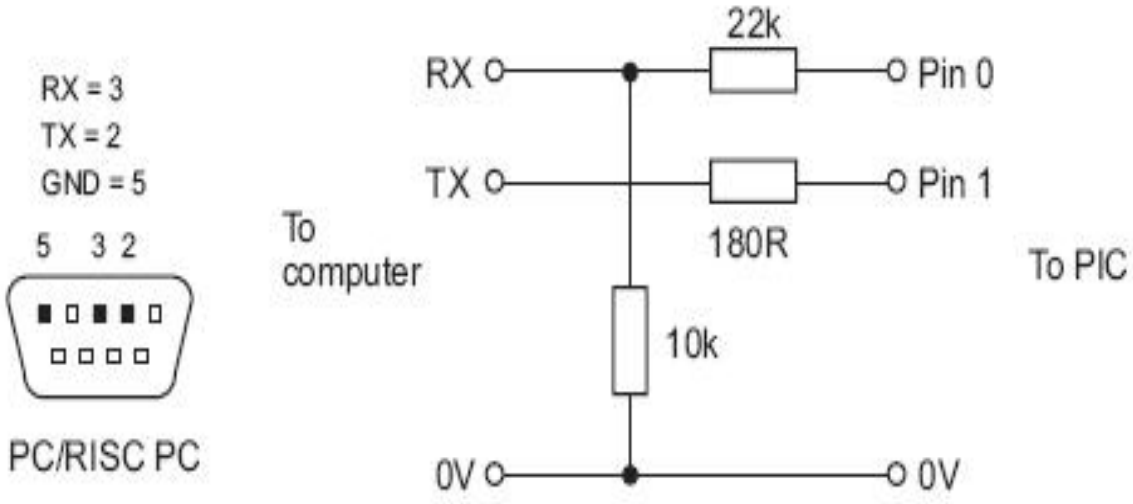
See the USB / Serial Download Circuit section of this manual for more details about the download circuit.

Notes:
1) The 10k/22k resistors must be included for reliable operation.
DO NOT leave the serial in pin floating as THE PROGRAM WILL NOT RUN!
2) The reset pin must be tied high with the 4k7 resistor to operate.
3) No external resonator is required as the chips have an internal resonator.

Circuito para programación



Circuito para comunicación serie de datos



F. Programa de la GUI en Matlab

INTERFAZ

```
function varargout = interfase_12marzo5(varargin)
% INTERFASE_12MARZO5 M-file for interfase_12marzo5.fig
%      INTERFASE_12MARZO5, by itself, creates a new INTERFASE_12MARZO5
or raises the existing
%      singleton*.
%
%      H = INTERFASE_12MARZO5 returns the handle to a new
INTERFASE_12MARZO5 or the handle to
%      the existing singleton*.
%
%      INTERFASE_12MARZO5('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in INTERFASE_12MARZO5.M with the given
input arguments.
%
%      INTERFASE_12MARZO5('Property','Value',...) creates a new
INTERFASE_12MARZO5 or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before interfase_12marzo5_OpeningFunction
gets calLED.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to interfase_12marzo5_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
interfase_12marzo5

% Last Modified by GUIDE v2.5 13-Mar-2008 18:24:12

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @interfase_12marzo5_OpeningFcn,
...
                   'gui_OutputFcn',  @interfase_12marzo5_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 interfase_12marzo5 is made visible.
function interfase_12marzo5_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 interfase_12marzo5 (see
VARARGIN)

% Choose default command line output for interfase_12marzo5
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes interfase_12marzo5 wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = interfase_12marzo5_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



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 str1
```

```
str1=str2double(get(handles.edit1,'String'));
if str1>60
    errordlg('el valor debe ser menor a 60!', '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



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 str2
str2=str2double(get(handles.edit2,'String'));
if str2>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



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 str3
str3=str2double(get(handles.edit3,'String'));
if str3>5
    errordlg('el valor debe ser menor a 5!', '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



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 str4
str4=str2double(get(handles.edit4,'String'));
if str4>60
    errordlg('el valor debe ser menor a 60!', '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



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 str5
str5=str2double(get(handles.edit5,'String'));
if str5>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



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 str6
str6=str2double(get(handles.edit6,'String'));
```

```
if str6>5
    errordlg('el valor debe ser menor a 5!', '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 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



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 str7
str7=str2double(get(handles.edit7,'String'));
if str7>60
    errordlg('el valor debe ser menor a 60!', '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



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 str8
str8=str2double(get(handles.edit8,'String'));
if str8>60
    errordlg('el valor debe ser menor a 60!', '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 during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (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



function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global str9
str9=str2double(get(handles.edit9,'String'));
if str9>5
    errordlg('el valor debe ser menor a 5!', 'Error Dialog Box',
'modal');
end

% Hints: get(hObject,'String') returns contents of edit9 as text
%        str2double(get(hObject,'String')) returns contents of edit9
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


% --- 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 colorizq1
    colorizq1 = 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


% --- 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 colorizq2
    colorizq2 = 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 popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (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


% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global colorder1
    colorder1 = get(hObject,'Value');
   return

% Hints: contents = get(hObject,'String') returns popupmenu3 contents
as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu3


% --- Executes during object creation, after setting all properties.
function popupmenu4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (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


% --- Executes on selection change in popupmenu4.
function popupmenu4_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
global colorder2
    colorder2 = get(hObject,'Value');
   return

% Hints: contents = get(hObject,'String') returns popupmenu4 contents
as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu4


% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (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



function edit10_Callback(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global str10
str10=str2double(get(handles.edit10,'String'));
if str10>31
    errordlg('el valor debe ser menor a 31!', 'Error Dialog Box',
'modal');
end

% Hints: get(hObject,'String') returns contents of edit10 as text
%        str2double(get(hObject,'String')) returns contents of edit10
as a double


% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (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



function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global str11
str11=str2double(get(handles.edit11,'String'));
if str11>31
    errordlg('el valor debe ser menor a 31!', 'Error Dialog Box',
'modal');
end

% Hints: get(hObject,'String') returns contents of edit11 as text
%        str2double(get(hObject,'String')) returns contents of edit11
as a double


% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit12 (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



function edit12_Callback(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global str12
str12=str2double(get(handles.edit12,'String'));
if str12>31
    errordlg('el valor debe ser menor a 31!', 'Error Dialog Box',
'modal');
end

% Hints: get(hObject,'String') returns contents of edit12 as text
%        str2double(get(hObject,'String')) returns contents of edit12
as a double


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

```
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (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



function edit13_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global str13
str13=str2double(get(handles.edit13,'String'));
if str13>31
    errordlg('el valor debe ser menor a 31!', 'Error Dialog Box',
'modal');
end

% Hints: get(hObject,'String') returns contents of edit13 as text
%        str2double(get(hObject,'String')) returns contents of edit13
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)

global colorizq1
global colorizq2
global colorder1
global colorder2
global str1
global str2
global str3
global str4
global str5
global str6
global str7
global str8
global str9
global str10
global str11
global str12
global str13
global str14
global str15
global str16
global str17
```

```
global str20
global str21

global rdata
str14=(3600*str3)+(60*str2)+str1;
str15=str14*4;
str20=(3600*str9)+(60*str8)+str7;
str21=str20*4;
enviard(str1);
enviard(str2);
enviard(str3);
enviard(str7);
enviard(str8);
enviard(str9);

if colorizq1==1 & colorizq2==1
    enviard(0);
    enviard(0);

elseif colorizq2==1
    enviard(str10);
    enviard(str10);
elseif colorizq1==1
    enviard(str11);
    enviard(str11);
elseif colorizq1<=colorizq2
    enviard(str10);
    enviard(str11);
elseif colorizq1>colorizq2
    enviard(str11);
    enviard(str10);
end

if colorder1==1 & colorder2==1
    enviard(0);
    enviard(0);

elseif colorder2==1
    enviard(str12);
    enviard(str12);
elseif colorder1==1
    enviard(str13);
    enviard(str13);
elseif colorder1<=colorder2
    enviard(str12);
    enviard(str13);
elseif colorder1>colorder2
    enviard(str13);
    enviard(str12);
end

if (colorizq1==1 & colorizq2==1)
    enviarc('bb');

elseif (colorizq1==2 & colorizq2==1) | (colorizq1==2 & colorizq2==2) |
(colorizq1==1 & colorizq2==2)
    enviarc('bb');
    %enviarc('b');
elseif (colorizq1==3 & colorizq2==1) | (colorizq1==3 & colorizq2==3) |
(colorizq1==1 & colorizq2==3)
    enviarc('rr');
```

```
    %enviarc('r');
elseif (colorizq1==4 & colorizq2==1) | (colorizq1==4 & colorizq2==4) |
(colorizq1==1 & colorizq2==4)
    enviarc('gg');
    %enviarc('g');
elseif (colorizq1==5 & colorizq2==1) | (colorizq1==5 & colorizq2==5) |
(colorizq1==1 & colorizq2==5)
    enviarc('ww');
    %enviarc('w');
elseif (colorizq1==6 & colorizq2==1) | (colorizq1==6 & colorizq2==6) |
(colorizq1==1 & colorizq2==6)
    enviarc('yy');
    %enviarc('y');
elseif (colorizq1==2 & colorizq2==3) | (colorizq1==3 & colorizq2==2)
    enviarc('br');
    %enviarc('r')
elseif (colorizq1==2 & colorizq2==4) | (colorizq1==4 & colorizq2==2)
    enviarc('bg');
    %enviarc('g')
elseif (colorizq1==2 & colorizq2==5) | (colorizq1==5 & colorizq2==2)
    enviarc('bw');
    %enviarc('w')
elseif (colorizq1==2 & colorizq2==6) | (colorizq1==6 & colorizq2==2)
    enviarc('by');
    %enviarc('y')
elseif (colorizq1==3 & colorizq2==4) | (colorizq1==4 & colorizq2==3)
    enviarc('rg');
    %enviarc('g')
elseif (colorizq1==3 & colorizq2==5) | (colorizq1==5 & colorizq2==3)
    enviarc('rw');
    %enviarc('w')
elseif (colorizq1==3 & colorizq2==6) | (colorizq1==6 & colorizq2==3)
    enviarc('ry');
    %enviarc('y')
elseif (colorizq1==4 & colorizq2==5) | (colorizq1==5 & colorizq2==4)
    enviarc('gw');
    %enviarc('w')
elseif (colorizq1==4 & colorizq2==6) | (colorizq1==6 & colorizq2==4)
    enviarc('gy');
    %enviarc('y')
elseif (colorizq1==5 & colorizq2==6) | (colorizq1==6 & colorizq2==5)
    enviarc('wy');
    %enviarc('y')
end

if (colorder1==1 & colorder2==1)
    enviarc('yy');

elseif (colorder1==2 & colorder2==1) | (colorder1==2 & colorder2==2) |
(colorder1==1 & colorder2==2)
    enviarc('bb');
    %enviarc('b');
elseif (colorder1==3 & colorder2==1) | (colorder1==3 & colorder2==3) |
(colorder1==1 & colorder2==3)
    enviarc('rr');
    %enviarc('r');
elseif (colorder1==4 & colorder2==1) | (colorder1==4 & colorder2==4) |
(colorder1==1 & colorder2==4)
    enviarc('gg');
    %enviarc('g');
```

```
elseif (colorder1==5 & colorder2==1) | (colorder1==5 & colorder2==5) |
(colorder1==1 & colorder2==5)
    enviarc('ww');
    %enviarc('w');
elseif (colorder1==6 & colorder2==1) | (colorder1==6 & colorder2==6) |
(colorder1==1 & colorder2==6)
    enviarc('yy');
    %enviarc('y');
elseif (colorder1==2 & colorder2==3) | (colorder1==3 & colorder2==2)
    enviarc('br');
    %enviarc('r')
elseif (colorder1==2 & colorder2==4) | (colorder1==4 & colorder2==2)
    enviarc('bg');
    %enviarc('g')
elseif (colorder1==2 & colorder2==5) | (colorder1==5 & colorder2==2)
    enviarc('bw');
    %enviarc('w')
elseif (colorder1==2 & colorder2==6) | (colorder1==6 & colorder2==2)
    enviarc('by');
    %enviarc('y')
elseif (colorder1==3 & colorder2==4) | (colorder1==4 & colorder2==3)
    enviarc('rg');
    %enviarc('g')
elseif (colorder1==3 & colorder2==5) | (colorder1==5 & colorder2==3)
    enviarc('rw');
    %enviarc('w')
elseif (colorder1==3 & colorder2==6) | (colorder1==6 & colorder2==3)
    enviarc('ry');
    %enviarc('y')
elseif (colorder1==4 & colorder2==5) | (colorder1==5 & colorder2==4)
    enviarc('gw');
    %enviarc('w')
elseif (colorder1==4 & colorder2==6) | (colorder1==6 & colorder2==4)
    enviarc('gy');
    %enviarc('y')
elseif (colorder1==5 & colorder2==6) | (colorder1==6 & colorder2==5)
    enviarc('wy');
    %enviarc('y')
end

str16=(3600*str6)+(60*str5)+str4;
str17=str16*4;
enviard(str4);
enviard(str5);
enviard(str6);

global colorizq1
global colorizq2
global colorder1
global colorder2
global str1
global str2
global str3
global str4
global str5
global str6
global str7
global str8
global str9
global str10
global str11
```

```
global str12
global str13
global str14
global str15
global str16
global str17
global rdata
pause(5)
enviard(0)
recibir1;
matrizando1(rdata)

global colorizq1
global colorizq2
global colorder1
global colorder2
global str1
global str2
global str3
global str4
global str5
global str6
global str7
global str8
global str9
global str10
global str11
global str12
global str13
global str14
global str15
global str16
global str17
global rdata

enviard(1)

recibir2;
matrizando2(rdata)

global colorizq1
global colorizq2
global colorder1
global colorder2
global str1
global str2
global str3
global str4
global str5
global str6
global str7
global str8
global str9
global str10
global str11
global str12
global str13
global str14
global str15
global str16
global str17
```

```
global contador_sensores
global rdata

enviard(2)

recibir3;
matrizando3(rdata)

%str1=str10,11
%str2=str12,13
%str9=str15
%str19=str14
%str10=str17
%str110=str16



function enviarc(msj)
% el argumento de la funcion enviar debe ser escrito entre apostrofes
dado
% que utiliza el comando fprintf el cual es solo para enviar
caracteres.
s=serial('com1','baudrate',4800);
fopen(s);
fprintf(s, msj);
fclose(s);
return

function enviard(msj)
% el argumento de la funcion enviar debe ser escrito entre apostrofes
dado
% que utiliza el comando fprintf el cual es solo para enviar
caracteres.
s=serial('com1','baudrate',4800);
fopen(s)
fwrite(s,msj)
fclose(s)
return

function rdata=recibir1
global rdata
global str15
global str18
global str14
str18=(8*str15)+8;
s=serial('com1','baudrate',4800,'Timeout',str14,'InputBufferSize',str1
8);
fopen(s)
rdata=fread(s);
fclose(s)
return

function rdata=recibir2
global rdata
global str17
global str16
global str19
str19=(8*str17)+8;
s=serial('com1','baudrate',4800,'Timeout',str16,'InputBufferSize',str1
9);
fopen(s)
```

```
rdata=fread(s);
fclose(s)
return

function rdata=recibir3
global rdata
global str20
global str21
global str22
str22=(8*str21)+8;
s=serial('com1','baudrate',4800,'Timeout',str20,'InputBufferSize',str2
2);
fopen(s)
rdata=fread(s);
fclose(s)
return

function datos=matrizando1(A)

global str14
global contador_sensores
global t
[filas columnas]=size(A);
filas2=filas/8;
B=zeros(filas2,8);
ra=0;
for rb=1:filas2,
    for j=1:8,
        ra=ra+1;
        B(rb,j)=A(ra,1);
    end
end

for i=1:filas2
    for j=1:8
        if B(i,j)>100
            B(i,j)=0;
        end
    end
end

filas4=filas2/4;

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;
C1=(5.*c1./Z1)-4;

for i=1:filas2
    if C1(i,1)<0.2
```

```
        C1(i,1)=0.2;
    end
end

contador_sensores=zeros(8,3);
for i=1:filas2-1
    if C1(i+1,1)<=(C1(i,1)-.07)
        contador_sensores(1,1)=contador_sensores(1,1)+1;
    end
end

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;
C2=(5.*c2./Z2)-3;

for i=1:filas2
    if C2(i,1)<1.2
        C2(i,1)=1.2;
    end
end

for i=1:filas2-1
    if C2(i+1,1)<=(C2(i,1)-.07)
        contador_sensores(2,1)=contador_sensores(2,1)+1;
    end
end

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;
C3=(5.*c3./Z3)-2;

for i=1:filas2
    if C3(i,1)<2.2
        C3(i,1)=2.2;
    end
end

for i=1:filas2-1
    if C3(i+1,1)<=(C3(i,1)-.07)
        contador_sensores(3,1)=contador_sensores(3,1)+1;
    end
end

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;
C4=(5.*c4./Z4)-1;

for i=1:filas2
    if C4(i,1)<3.2
        C4(i,1)=3.2;
    end
end

for i=1:filas2-1
    if C4(i+1,1)<=(C4(i,1)-.07)
        contador_sensores(4,1)=contador_sensores(4,1)+1;
    end
end

c5=zeros(filas2,1);
for i=1:filas2
    c5(i,1)=B(i,5);
end
C5=zeros(filas2,1);
Z5=max(c5);

%C5=5.*c5./Z5;
C5=(5.*c5./Z5);

for i=1:filas2
    if C5(i,1)<4.2
        C5(i,1)=4.2;
    end
end

for i=1:filas2-1
    if C5(i+1,1)<=(C5(i,1)-.07)
        contador_sensores(5,1)=contador_sensores(5,1)+1;
    end
end
c6=zeros(filas2,1);
for i=1:filas2
    c6(i,1)=B(i,6);
end
C6=zeros(filas2,1);
Z6=max(c6);

%C6=6.*c6./Z6;
C6=(5.*c6./Z6)+1;

for i=1:filas2
    if C6(i,1)<5.2
        C6(i,1)=5.2;
    end
end

for i=1:filas2-1
    if C6(i+1,1)<=(C6(i,1)-.07)
        contador_sensores(6,1)=contador_sensores(6,1)+1;
    end
end

c7=zeros(filas2,1);
for i=1:filas2
```

```
    c7(i,1)=B(i,7);
end
C7=zeros(filas2,1);
Z7=max(c7);

%C7=7.*c7./Z7;
C7=(5.*c7./Z7)+2;

for i=1:filas2
    if C7(i,1)<6.2
        C7(i,1)=6.2;
    end
end

for i=1:filas2-1
    if C7(i+1,1)<=(C7(i,1)-.07)
        contador_sensores(7,1)=contador_sensores(7,1)+1;
    end
end
c8=zeros(filas2,1);
for i=1:filas2
    c8(i,1)=B(i,8);
end
C8=zeros(filas2,1);
Z8=max(c8);

%C8=8.*c8./Z8;
C8=(5.*c8./Z8)+3;

for i=1:filas2
    if C8(i,1)<7.2
        C8(i,1)=7.2;
    end
end

for i=1:filas2-1
    if C8(i+1,1)<=(C8(i,1)-.07)
        contador_sensores(8,1)=contador_sensores(8,1)+1;
    end
end

figure
subplot(131),plot(t,C1,'b',t,C2,'b',t,C3,'b',t,C4,'b',t,C5,'b',t,C6,'b
',t,C7,'b',t,C8,'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
%legend('derecho ext','derecho int','izquierdo int','izquierdo ext')
%text(str1,str2,'string')
grid
axis([0 str14 0 9])
xlabel('Primer tiempo de obscuridad [s]')
ylabel('actividad en los sensores [on/off]')
return


function datos=matrizando2(A)

global str16
global colorizq1
global colorizq2
global colorder1
global colorder2
global contador_sensores
global t
[filas columnas]=size(A);
filas2=filas/8;
B=zeros(filas2,8);
ra=0;
for rb=1:filas2,
    for j=1:8,
        ra=ra+1;
        B(rb,j)=A(ra,1);
    end
end

for i=1:filas2
    for j=1:8
        if B(i,j)>100
            B(i,j)=0;
        end
    end
end

%filas4=filas2/4;
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;
C1=(5.*c1./Z1)-4;

for i=1:filas2
    if C1(i,1)<0.2
        C1(i,1)=0.2;
    end
end
```

```
for i=1:filas2-1
    if C1(i+1,1)<=(C1(i,1)-.07)
        contador_sensores(1,2)=contador_sensores(1,2)+1;
    end
end

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;
C2=(5.*c2./Z2)-3;

for i=1:filas2
    if C2(i,1)<1.2
        C2(i,1)=1.2;
    end
end

for i=1:filas2-1
    if C2(i+1,1)<=(C2(i,1)-.07)
        contador_sensores(2,2)=contador_sensores(2,2)+1;
    end
end

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;
C3=(5.*c3./Z3)-2;

for i=1:filas2
    if C3(i,1)<2.2
        C3(i,1)=2.2;
    end
end

for i=1:filas2-1
    if C3(i+1,1)<=(C3(i,1)-.07)
        contador_sensores(3,2)=contador_sensores(3,2)+1;
    end
end

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;
C4=(5.*c4./Z4)-1;

for i=1:filas2
```

```
    if C4(i,1)<3.2
        C4(i,1)=3.2;
    end
end

for i=1:filas2-1
    if C4(i+1,1)<=(C4(i,1)-.07)
        contador_sensores(4,2)=contador_sensores(4,2)+1;
    end
end

c5=zeros(filas2,1);
for i=1:filas2
    c5(i,1)=B(i,5);
end
C5=zeros(filas2,1);
Z5=max(c5);

%C5=5.*c5./Z5;
C5=(5.*c5./Z5);

for i=1:filas2
    if C5(i,1)<4.2
        C5(i,1)=4.2;
    end
end

for i=1:filas2-1
    if C5(i+1,1)<=(C5(i,1)-.07)
        contador_sensores(5,2)=contador_sensores(5,2)+1;
    end
end
c6=zeros(filas2,1);
for i=1:filas2
    c6(i,1)=B(i,6);
end
C6=zeros(filas2,1);
Z6=max(c6);

%C6=6.*c6./Z6;
C6=(5.*c6./Z6)+1;

for i=1:filas2
    if C6(i,1)<5.2
        C6(i,1)=5.2;
    end
end

for i=1:filas2-1
    if C6(i+1,1)<=(C6(i,1)-.07)
        contador_sensores(6,2)=contador_sensores(6,2)+1;
    end
end

c7=zeros(filas2,1);
for i=1:filas2
    c7(i,1)=B(i,7);
end
C7=zeros(filas2,1);
Z7=max(c7);
```

```
%C7=7.*c7./Z7;
C7=(5.*c7./Z7)+2;

for i=1:filas2
    if C7(i,1)<6.2
        C7(i,1)=6.2;
    end
end

for i=1:filas2-1
    if C7(i+1,1)<=(C7(i,1)-.07)
        contador_sensores(7,2)=contador_sensores(7,2)+1;
    end
end

c8=zeros(filas2,1);
for i=1:filas2
    c8(i,1)=B(i,8);
end
C8=zeros(filas2,1);
Z8=max(c8);

%C8=8.*c8./Z8;
C8=(5.*c8./Z8)+3;

for i=1:filas2
    if C8(i,1)<7.2
        C8(i,1)=7.2;
    end
end

for i=1:filas2-1
    if C8(i+1,1)<=(C8(i,1)-.07)
        contador_sensores(8,2)=contador_sensores(8,2)+1;
    end
end

subplot(132),plot(t,C1,'b',t,C2,'b',t,C3,'b',t,C4,'b',t,C5,'b',t,C6,'b
',t,C7,'b',t,C8,'b')

global colorizq1
global colorizq2
global str10
global str11

if colorizq1==1 & colorizq2==1
title('luz izquierda: Sin luz ')

elseif colorizq1==2 & colorizq2==2 | colorizq1==2 &colorizq2==1 |
colorizq1==1 & colorizq2==2
title(['luz izquierda: Azul ','intensidad
izquierda:',num2str(str10)],'Color','b')
elseif (colorizq1==2 & colorizq2==3) | (colorizq1==3 & colorizq2==2)
title(['luz izquierda: Azul, Rojo ','intensidad
izquierda:',num2str(str10),', ', num2str(str11)],'Color','b')
elseif (colorizq1==2 & colorizq2==4) | (colorizq1==4 & colorizq2==2)
title(['luz izquierda: Azul, Verde ','intensidad
izquierda:',num2str(str10),', ', num2str(str11)],'Color','b')
elseif (colorizq1==2 & colorizq2==5) | (colorizq1==5 & colorizq2==2)
title(['luz izquierda: Azul, Blanco ','intensidad
izquierda:',num2str(str10),', ', num2str(str11)],'Color','b')
```

```
elseif (colorizq1==2 & colorizq2==6) | (colorizq1==6 & colorizq2==2)
title(['luz izquierda: Azul, Ambar ','intensidad
izquierda:',num2str(str10),', ', num2str(str11)],'Color','b')
elseif colorizq1==3 & colorizq2==3 | colorizq1==3 &colorizq2==1 |
colorizq1==1 & colorizq2==3
title(['luz izquierda: Rojo ','intensidad
izquierda:',num2str(str10)],'Color','r')
elseif (colorizq1==3 & colorizq2==4) | (colorizq1==4 & colorizq2==3)
title(['luz izquierda: Rojo, Verde ','intensidad
izquierda:',num2str(str10),', ', num2str(str11)],'Color','r')
elseif (colorizq1==3 & colorizq2==5) | (colorizq1==5 & colorizq2==3)
title(['luz izquierda: Rojo, Blanco ','intensidad
izquierda:',num2str(str10),', ', num2str(str11)],'Color','r')
elseif (colorizq1==3 & colorizq2==6) | (colorizq1==6 & colorizq2==3)
title(['luz izquierda: Rojo, Ambar ','intensidad
izquierda:',num2str(str10),', ', num2str(str11)],'Color','r')
elseif colorizq1==4 & colorizq2==4 | colorizq1==4 &colorizq2==1 |
colorizq1==1 & colorizq2==4
title(['luz izquierda: Verde ','intensidad
izquierda:',num2str(str10)],'Color','g')
elseif (colorizq1==4 & colorizq2==5) | (colorizq1==5 & colorizq2==4)
title(['luz izquierda: Verde, Blanco ','intensidad
izquierda:',num2str(str10),', ', num2str(str11)],'Color','g')
elseif (colorizq1==4 & colorizq2==6) | (colorizq1==6 & colorizq2==4)
title(['luz izquierda: Verde, Ambar ','intensidad
izquierda:',num2str(str10),', ', num2str(str11)],'Color','g')
elseif colorizq1== 5 & colorizq2==5 | colorizq1==5 &colorizq2==1 |
colorizq1==1 & colorizq2==5
title(['luz izquierda: Blanco ','intensidad
izquierda:',num2str(str10)],'Color','w')
elseif (colorizq1==5 & colorizq2==6) | (colorizq1==6 & colorizq2==5)
title(['luz izquierda: Blanco, Ambar ','intensidad
izquierda:',num2str(str10),', ', num2str(str11)],'Color','w')
elseif colorizq1==6 & colorizq2==6 | colorizq1==6 &colorizq2==1 |
colorizq1==1 & colorizq2==6
title(['luz izquierda: Ambar ','intensidad
izquierda:',num2str(str10)],'Color','y')
end

global colorder1
global colorder2
global str12
global str13

if colorder1==1 & colorder2==1
xlabel('luz derecha: Sin luz ')

elseif colorder1==2 & colorder2==2 | colorder1==2 &colorder2==1 |
colorder1==1 & colorder2==2
xlabel(['luz derecha: Azul ','intensidad
derecha:',num2str(str12)],'Color','b')
elseif (colorder1==2 & colorder2==3) | (colorder1==3 & colorder2==2)
xlabel(['luz derecha: Azul, Rojo ','intensidad
derecha:',num2str(str12),', ', num2str(str13)],'Color','r')
elseif (colorder1==2 & colorder2==4) | (colorder1==4 & colorder2==2)
xlabel(['luz derecha: Azul, Verde ','intensidad
derecha:',num2str(str12),', ', num2str(str13)],'Color','g')
elseif (colorder1==2 & colorder2==5) | (colorder1==5 & colorder2==2)
xlabel(['luz derecha: Azul, Blanco ','intensidad
derecha:',num2str(str12),', ', num2str(str13)],'Color','w')
elseif (colorder1==2 & colorder2==6) | (colorder1==6 & colorder2==2)
```

```
xlabel(['luz derecha: Azul, Ambar ','intensidad
derecha:',num2str(str12),', ', num2str(str13)],'Color','y')
elseif (colorder1==3 & colorder2==3) | colorder1==3 &colorder2==1 |
colorder1==1 & colorder2==3
xlabel(['luz derecha: Rojo ','intensidad
derecha:',num2str(str12)],'Color','r')
elseif (colorder1==3 & colorder2==4) | (colorder1==4 & colorder2==3)
xlabel(['luz derecha: Rojo, Verde ','intensidad
derecha:',num2str(str12),', ', num2str(str13)],'Color','g')
elseif (colorder1==3 & colorder2==5) | (colorder1==5 & colorder2==3)
xlabel(['luz derecha: Rojo, Blanco ','intensidad
derecha:',num2str(str12),', ', num2str(str13)],'Color','w')
elseif (colorder1==3 & colorder2==6) | (colorder1==6 & colorder2==3)
xlabel(['luz derecha: Rojo, Ambar ','intensidad
derecha:',num2str(str12),', ', num2str(str13)],'Color','y')
elseif (colorder1==4 & colorder2==4) | colorder1==4 &colorder2==1 |
colorder1==1 & colorder2==4
xlabel(['luz derecha: Verde ','intensidad
derecha:',num2str(str12)],'Color','g')
elseif (colorder1==4 & colorder2==5) | (colorder1==5 & colorder2==4)
xlabel(['luz derecha: Verde, Blanco ','intensidad
derecha:',num2str(str12),', ', num2str(str13)],'Color','w')
elseif (colorder1==4 & colorder2==6) | (colorder1==6 & colorder2==4)
xlabel(['luz derecha: Verde, Ambar ','intensidad
derecha:',num2str(str12),', ', num2str(str13)],'Color','y')
elseif (colorder1==5 & colorder2==5) | colorder1==5 &colorder2==1 |
colorder1==1 & colorder2==5
xlabel(['luz derecha: Blanco ','intensidad
derecha:',num2str(str12)],'Color','w')
elseif (colorder1==5 & colorder2==6) | (colorder1==6 & colorder2==5)
xlabel(['luz derecha: Blanco, Ambar ','intensidad
derecha:',num2str(str12),', ', num2str(str13)],'Color','y')
elseif (colorder1==6 & colorder2==6) | colorder1==6 &colorder2==1 |
colorder1==1 & colorder2==6
xlabel(['luz derecha: Ambar ','intensidad
derecha:',num2str(str12)],'Color','y')
end
%legend('derecho ext','derecho int','izquierdo int','izquierdo ext')
grid
axis([0 str16 0 9])
%xlabel('tiempo de experimento [s]')
%ylabel('actividad en los sensores [on/off]')
return

function datos=matrizando3(A)

global str20
global contador_sensores
global t
[filas columnas]=size(A);
filas2=filas/8;
B=zeros(filas2,8);
ra=0;
for rb=1:filas2,
    for j=1:8,
        ra=ra+1;
        B(rb,j)=A(ra,1);
    end
end

for i=1:filas2
```

```
    for j=1:8
        if B(i,j)>100
            B(i,j)=0;
        end
    end
end

B=5*B;
filas4=filas2/4;
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;
C1=(5.*c1./Z1)-4;

for i=1:filas2
    if C1(i,1)<0.2
        C1(i,1)=0.2;
    end
end

for i=1:filas2-1
    if C1(i+1,1)<=(C1(i,1)-.07)
        contador_sensores(1,3)=contador_sensores(1,3)+1;
    end
end
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;
C2=(5.*c2./Z2)-3;

for i=1:filas2
    if C2(i,1)<1.2
        C2(i,1)=1.2;
    end
end

for i=1:filas2-1
    if C2(i+1,1)<=(C2(i,1)-.07)
        contador_sensores(2,3)=contador_sensores(2,3)+1;
    end
end

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;
C3=(5.*c3./Z3)-2;

for i=1:filas2
    if C3(i,1)<2.2
        C3(i,1)=2.2;
    end
end

for i=1:filas2-1
    if C3(i+1,1)<=(C3(i,1)-.07)
        contador_sensores(3,3)=contador_sensores(3,3)+1;
    end
end

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;
C4=(5.*c4./Z4)-1;

for i=1:filas2
    if C4(i,1)<3.2
        C4(i,1)=3.2;
    end
end

for i=1:filas2-1
    if C4(i+1,1)<=(C4(i,1)-.07)
        contador_sensores(4,3)=contador_sensores(4,3)+1;
    end
end

c5=zeros(filas2,1);
for i=1:filas2
    c5(i,1)=B(i,5);
end
C5=zeros(filas2,1);
Z5=max(c5);

%C5=5.*c5./Z5;
C5=(5.*c5./Z5);

for i=1:filas2
    if C5(i,1)<4.2
        C5(i,1)=4.2;
    end
end

for i=1:filas2-1
    if C5(i+1,1)<=(C5(i,1)-.07)
        contador_sensores(5,3)=contador_sensores(5,3)+1;
    end
```

```
end

c6=zeros(filas2,1);
for i=1:filas2
    c6(i,1)=B(i,6);
end
C6=zeros(filas2,1);
Z6=max(c6);

%C6=6.*c6./Z6;
C6=(5.*c6./Z6)+1;

for i=1:filas2
    if C6(i,1)<5.2
        C6(i,1)=5.2;
    end
end

for i=1:filas2-1
    if C6(i+1,1)<=(C6(i,1)-.07)
        contador_sensores(6,3)=contador_sensores(6,3)+1;
    end
end

c7=zeros(filas2,1);
for i=1:filas2
    c7(i,1)=B(i,7);
end
C7=zeros(filas2,1);
Z7=max(c7);

%C7=7.*c7./Z7;
C7=(5.*c7./Z7)+2;

for i=1:filas2
    if C7(i,1)<6.2
        C7(i,1)=6.2;
    end
end

for i=1:filas2-1
    if C7(i+1,1)<=(C7(i,1)-.07)
        contador_sensores(7,3)=contador_sensores(7,3)+1;
    end
end

c8=zeros(filas2,1);
for i=1:filas2
    c8(i,1)=B(i,8);
end
C8=zeros(filas2,1);
Z8=max(c8);

%C8=8.*c8./Z8;
C8=(5.*c8./Z8)+3;

for i=1:filas2
    if C8(i,1)<7.2
        C8(i,1)=7.2;
    end
end
```

```
for i=1:filas2-1
    if C8(i+1,1)<=(C8(i,1)-.07)
        contador_sensores(8,3)=contador_sensores(8,3)+1;
    end
end

contador_sensores
subplot(133),plot(t,C1,'b',t,C2,'b',t,C3,'b',t,C4,'b',t,C5,'b',t,C6,'b
',t,C7,'b',t,C8,'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
%legend('derecho ext','derecho int','izquierdo int','izquierdo ext')
%text(str1,str2,'string')
grid
axis([0 str20 0 9])
xlabel('Tiempo de obscuridad final [s]')
%ylabel('actividad en los sensores [on/off]')
Return
```

G. Caracterización del dispositivo

	2 cm				
Intensidad Relativa	Azul	Rojo	Verde	Blanco	Ámbar
0	0,26	0,26	0,26	0,26	0,26
1	0,26	0,26	0,26	0,26	0,26
2	0,26	0,26	0,26	0,26	0,26
3	0,26	0,26	0,26	0,26	0,26
4	0,26	0,26	0,26	0,26	0,26
5	0,26	0,26	0,26	0,26	0,26
6	0,26	0,26	0,26	0,26	0,26
7	0,26	0,26	0,26	0,26	0,26
8	0,26	0,26	0,26	0,26	0,26
9	0,26	0,26	0,26	0,26	0,26
10	0,26	0,26	0,26	0,26	0,26
11	0,26	0,26	0,26	0,26	0,26
12	0,26	0,26	0,26	0,26	0,26
13	0,26	0,26	0,26	0,26	0,26
14	0,26	0,26	0,26	0,26	0,26
15	0,26	0,26	0,26	0,26	0,26
16	0,26	0,36	0,26	0,26	0,33

17	0,26	0,41	0,26	0,26	0,36
18	0,26	0,46	0,26	0,26	0,42
19	0,26	0,53	0,26	0,26	0,44
20	0,26	0,66	0,26	0,26	0,46
21	0,26	0,71	0,26	0,26	0,54
22	0,33	0,79	0,35	1,55	0,64
23	0,78	0,86	0,46	24,56	0,68
24	2,68	0,95	0,51	45,18	0,9
25	2,72	1,1	0,56	48,28	1,01
26	4,26	1,17	0,75	76,28	1,1
27	6,25	1,21	0,89	104,45	1,22
28	8,59	1,33	0,97	107,48	1,25
29	11,22	1,52	1,01	204,4	2,25
30	16,85	1,73	3,3	253,78	2,72

3 cm

Intensidad Relativa	Azul	Rojo	Verde	Blanco	Ámbar
0	0,26	0,26	0,26	0,26	0,26
1	0,26	0,26	0,26	0,26	0,26
2	0,26	0,26	0,26	0,26	0,26
3	0,26	0,26	0,26	0,26	0,26
4	0,26	0,26	0,26	0,26	0,26
5	0,26	0,26	0,26	0,26	0,26
6	0,26	0,26	0,26	0,26	0,26
7	0,26	0,26	0,26	0,26	0,26
8	0,26	0,26	0,26	0,26	0,26
9	0,26	0,26	0,26	0,26	0,26
10	0,26	0,26	0,26	0,26	0,26
11	0,26	0,26	0,26	0,26	0,26
12	0,26	0,26	0,26	0,26	0,26
13	0,26	0,26	0,26	0,26	0,26
14	0,26	0,26	0,26	0,26	0,26
15	0,26	0,26	0,26	0,26	0,26
16	0,26	0,33	0,26	0,26	0,26
17	0,26	0,36	0,26	0,26	0,33
18	0,26	0,39	0,26	0,26	0,36
19	0,26	0,4	0,26	0,26	0,37
20	0,26	0,44	0,26	0,26	0,42
21	0,26	0,47	0,26	0,26	0,44
22	0,33	0,52	0,26	0,26	0,46
23	0,7	0,57	0,26	3,41	0,54
24	1,21	0,62	0,26	3,79	0,57
25	2,42	0,69	0,26	6,94	0,69
26	3,77	0,74	0,36	11,25	0,77
27	5,62	0,77	0,56	13,75	0,79
28	8,01	0,81	0,63	16,84	1,09
29	11,09	0,93	0,78	34,72	1,25
30	16,62	1,02	1,54	35,64	1,59

4 cm

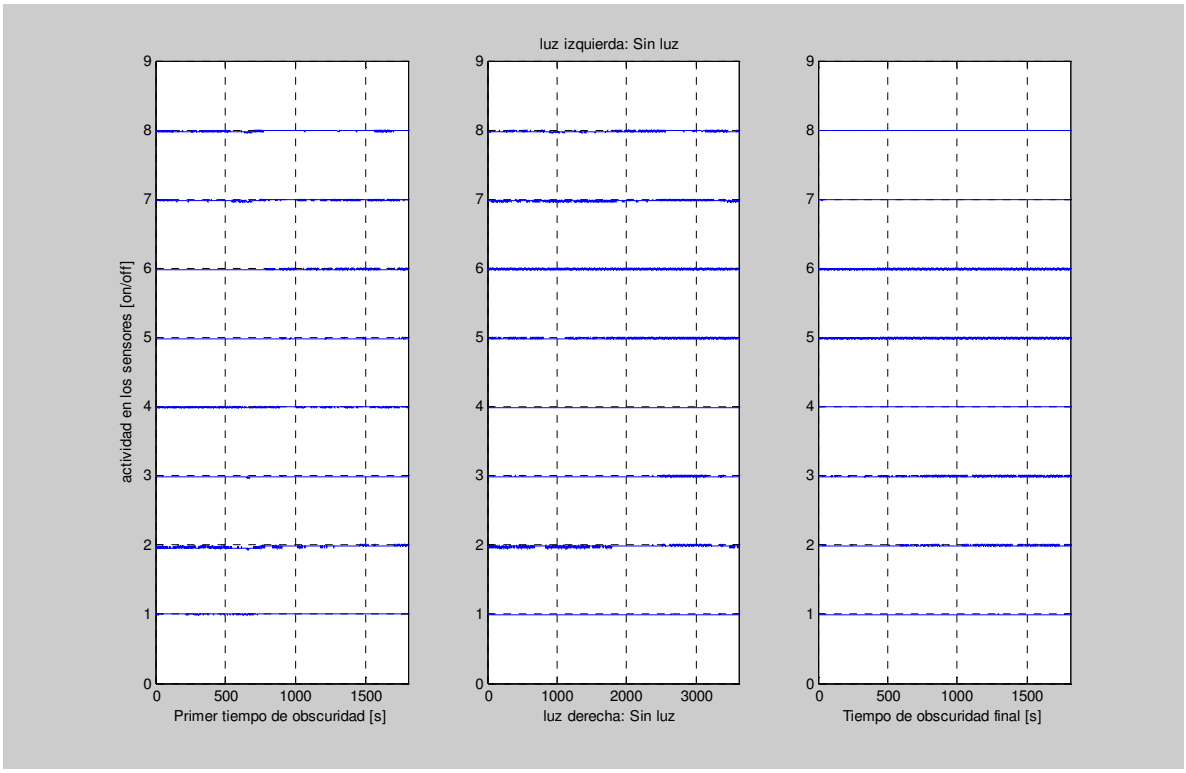
Intensidad Relativa	Azul	Rojo	Verde	Blanco	Ámbar
0	0,26	0,26	0,26	0,26	0,26

1	0,26	0,26	0,26	0,26	0,26
2	0,26	0,26	0,26	0,26	0,26
3	0,26	0,26	0,26	0,26	0,26
4	0,26	0,26	0,26	0,26	0,26
5	0,26	0,26	0,26	0,26	0,26
6	0,26	0,26	0,26	0,26	0,26
7	0,26	0,26	0,26	0,26	0,26
8	0,26	0,26	0,26	0,26	0,26
9	0,26	0,26	0,26	0,26	0,26
10	0,26	0,26	0,26	0,26	0,26
11	0,26	0,26	0,26	0,26	0,26
12	0,26	0,26	0,26	0,26	0,26
13	0,26	0,26	0,26	0,26	0,26
14	0,26	0,26	0,26	0,26	0,26
15	0,26	0,26	0,26	0,26	0,26
16	0,26	0,26	0,26	0,26	0,26
17	0,26	0,26	0,26	0,26	0,26
18	0,26	0,26	0,26	0,26	0,26
19	0,26	0,26	0,26	0,26	0,26
20	0,26	0,33	0,26	0,26	0,33
21	0,26	0,36	0,26	0,26	0,35
22	0,33	0,39	0,26	0,26	0,38
23	0,75	0,43	0,26	1,09	0,42
24	0,81	0,44	0,26	1,21	0,44
25	1,33	0,49	0,26	4,5	0,46
26	2,03	0,52	0,33	7,49	0,49
27	2,74	0,55	0,46	11,01	0,56
28	3,85	0,56	0,59	15,59	0,59
29	5,07	0,64	0,63	15,62	0,7
30	6,56	0,71	1,12	28,66	1,1

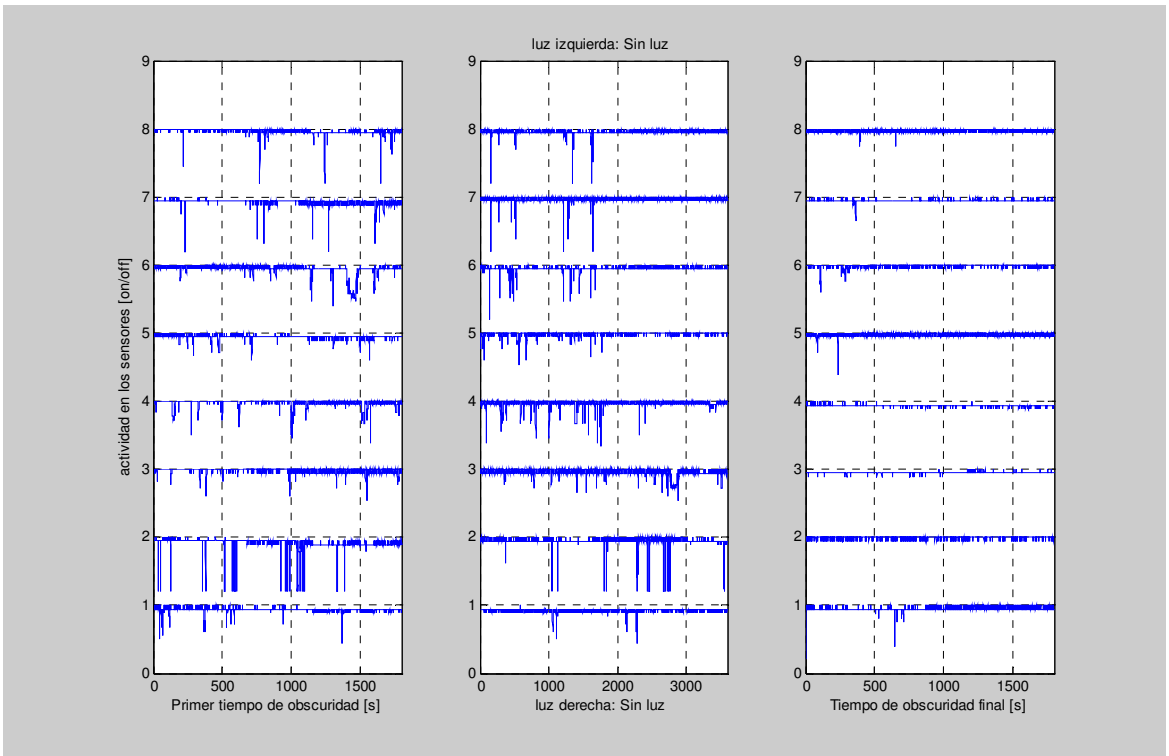
H. Gráficas del estado basal

Se hicieron pruebas con dos horas de obscuridad sin mosca y con mosca, y los resultados fueron los siguientes.

Sin mosca:



Con mosca:



Contador:

	obscuridad		
mosca 4	obsc inicial	experimento	obsc final
sensor 1	2	2	2
sensor 2	2	3	1
sensor 3	1	7	8
sensor 4	7	8	2
sensor 5	0	2	0
sensor 6	0	0	0
sensor 7	0	0	0
sensor 8	0	0	0
promedio	1,5	2,75	1,625

I. Cotización por dispositivo

Los precios son en pesos mexicanos.

Cantidad	Nombre	Precio por unidad	Precio total
1	Picaxe 40X-1	$280.00	$280.00
1	Picaxe 28X-1	$260.00	$260.00
1	Picaxe 18A	$220.00	$220.00
4	Potenciómetro digital	$42.61	$170.44
1	Regulador de voltaje	$7.00	$7.00
1	Micro switch push	$3.00	$3.00
4	Jack 3.5 mm estéreo	$5.00	$20.00
1	Plug 3.5 mm estéreo	$4.00	$4.00
1	Terminales con dos tornillos	$6.00	$6.00
2	Capacitor 470 uF	$7.00	$14.00
2	Capacitor 0.1 uF	$3.00	$6.00
2	Resistor 68 ohms	$0.61	$1.22
4	Resistor 100 ohms	$0.61	$2.44
2	Resistor 270 ohms	$0.61	$1.22
4	Resistor 330 ohms	$0.61	$2.44
10	Resistor 560 ohms	$0.61	$6.09
2	Resistor 680 ohms	$0.61	$1.22
18	Resistor 1 kohms	$0.61	$10.96
2	Resistor 2.2 kohms	$0.61	$1.22
2	Resistor 4.7 kohms	$0.61	$1.22
28	Resistor 10 kohms	$0.61	$17.05
4	Resistor 22 kohms	$0.61	$2.44
2	Resistor 100 kohms	$0.61	$1.22
26	Transistor bc547	$2.61	$67.83
2	LED azul súper brillante 5 mm	$6.00	$12.00
2	LED ámbar súper brillante 5 mm	$6.00	$12.00
2	LED verde súper brillante 5 mm	$6.00	$12.00
2	LED rojo súper brillante 5 mm	$6.00	$12.00
2	LED blanco súper brillante 5 mm	$10.00	$20.00

8	LED emisor de IR 5 mm	$5.00	$40.00
8	Fototransistor 5 mm	$6.00	$48.00
1	Base para c.i. 40 patas	$4.00	$4.00
1	Base para c.i. 28 patas	$4.00	$4.00
1	Base para c.i. 18 patas	$2.00	$2.00
4	Base para c.i. 8 patas	$2.00	$8.00
1	Broche porta pilas 9 V	$4.00	$4.00
1	Conector DB9-Hembra	$10.00	$10.00
2	Metro cable para micrófono estéreo	$15.00	$30.00
2	Lentes	$300.00	$600.00
1	Fenólica	$40.00	$40.00
6	Metro de cable bicolor calibre 22	$6.00	$36.00
1	Gabinete de plástico	$150.00	$150.00
	TOTAL		$2,150.99

7. Bibliografía

1. http://es.wikipedia.org/wiki/Drosophila

2. Tesis: "Diseño de un dispositivo para el estudio cuantitativo de conductas evocadas por luz en la mosca *Drosophila*" de Giovanni Fonseca Fonseca y Clara Susana Yáñez Martínez.

3. http://www.monografias.com/trabajos12/microco/microco.shtml

4. http://www.unicrom.com/Tut_diodo_LED.asp

5. Bate M. and Martínez A. *The development of Drosophila Melanogaster*, Cold Spring Harbor Laboratoy Press, 1993, Vol 3, 22(1,2) p1279-1281

6. Freeman Matthew (2005) Eye development: stable cell fate decision in insects colour vision. *Current Biology* 15, 924-926.

7. Ramos Patricia y colaboradores. *Manual de laboratorio de Genética para Drosophila melanogaster*,MacGraw-Hill México 1993, 1, p3

8. 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.

9. Zucker Charles S. (1996). The biology of vision in *Drosophila*. *PNAS* 93, 571-576.

10. Boylestad and Nashelsky, *Electrónica: Teoría de circuitos y dispositivos electrónicos*. Pearson Education, 8ª edición, México 2003, 1(16), p 40-41.

11. Malvino Albert Paul *Principios de electrónica* MacGraw-Hill, 3ª edición 1991, 4(3), p 115-116.

12. Manuales 1, 2 y 3 del microcontrolador PICAXE, disponible en www.picaxe.com.