



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES**  
**Y REDES**

**“DISEÑO E IMPLEMENTACIÓN DE DOS SISTEMAS DE MONITOREO**  
**PARA LA EVALUACIÓN DE LA ACTIVIDAD ANSIOLÍTICA IN VIVO EN**  
**RATAS (RATTUS NORVEGICUS)”**

**TRABAJO DE TITULACIÓN: PROPUESTA TECNOLÓGICA**

Trabajo de titulación presentado para optar al grado académico de:

**INGENIERA EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES**

**AUTORA: ERICA CAROLINA ADRIANO CAMPOVERDE**

**TUTOR: ING. PAÚL DAVID MORENO AVILÉS. PhD**

**Riobamba-Ecuador**

**2018**

**@2018, Erica Carolina Adriano Campoverde**

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA ELECTRÓNICA**  
**EN TELECOMUNICACIONES Y REDES**

El Tribunal del Trabajo de Titulación certifica que el Trabajo de Titulación: “DISEÑO E IMPLEMENTACIÓN DE DOS SISTEMAS DE MONITOREO PARA LA EVALUACIÓN DE LA ACTIVIDAD ANSIOLÍTICA IN VIVO EN RATAS (*RATTUS NORVEGICUS*)”, de responsabilidad de Erica Carolina Adriano Campoverde ha sido minuciosamente revisado por los miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

<b>NOMBRE</b>	<b>FIRMA</b>	<b>FECHA</b>
Dr. Julio Roberto Santillán Castillo		
<b>VICEDECANO FACULTAD DE INFORMÁTICA Y ELECTRÓNICA</b>	_____	_____
Ing. Patricio Romero		
<b>DIRECTOR DE ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES</b>	_____	_____
Ing. Paúl David Moreno Avilés		
<b>DIRECTOR DEL TRABAJO DE TITULACIÓN</b>	_____	_____
Ing. Oswaldo Martínez Guashima		
<b>MIEMBRO DEL TRIBUNAL</b>	_____	_____

Yo, Erica Carolina Adriano Campoverde declaro ser la autora del presente trabajo de titulación: “DISEÑO E IMPLEMENTACIÓN DE DOS SISTEMAS DE MONITOREO PARA LA EVALUACIÓN DE LA ACTIVIDAD ANSIOLÍTICA IN VIVO EN RATAS (RATTUS NORVEGICUS)”, que fue elaborado en su totalidad por mí, bajo la dirección del Ingeniero David Moreno, haciéndome totalmente responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio de la misma pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.

Erica Carolina Adriano Campoverde

## **DEDICATORIA**

Este trabajo está dedicado en primer lugar a Dios por ser mi guía en cada instante de mi vida, por siempre brindarme sus bendiciones ayudándome a ser mejor persona cada día, a mis padres Wilson y Martha quienes con su ejemplo han sabido educarme y brindarme su apoyo no solo económico sino también emocional, demostrándome que todo es posible si lucha por conseguirlo, a mis hermanos Mercedes, Kevin y Catherine por ser esos amigos incondicionales que con su sonrisa y consejos me animan a seguir luchando por mis metas, y sobre todo a mi sobrina Odette, pequeña mía eres el ángel que Dios me envió para recordarme que no estoy sola, eres mi motivación para seguir perseverando por alcanzar cada una de mis metas. Los amo con todo mi corazón.

Erica Carolina.

## **AGRADECIMIENTO**

Agradezco a Dios por brindarme salud y sobre todo por permitirme tener una familia excepcional, a mis padres por siempre estar apoyándome y no permitir que me dé por vencida, a mi hermano y hermanas por ser siempre mis amigos, consejeros y hasta mi paño de lágrimas, a mi bella Odette por ser esa luz que ilumina mi vida, desde que llegaste mi vida dio un giro de 180 grados por ti hermosa seré la mejor, a mis amigos, amigas tanto a los de antaño como a los que tuve la dicha de conocer a lo largo de mi vida politécnica, quienes con su abrazo fraterno y sonrisa sincera me animaban a seguir adelante, quienes me acompañaron en mis tristezas, alegrías, derrotas y fracasos, dicen que los amigos son la familia que uno decide tener y yo tuve la fortuna de formar una gran familia, gracias amigos, amigas por ser cómplices en esta travesía.

De igual forma agradezco a mi templo del saber mí querida Escuela Superior Politécnica de Chimborazo, en particular a la Escuela de Ingeniería Electrónica en Telecomunicaciones y Redes, el lugar que me permitió formarme para ser una gran profesional.

A todos los docentes que formaron parte de mi vida estudiantil, transmitiéndome todos sus conocimientos que de una u otra forma supieron motivarme para seguir aprendiendo aún más de este extenso mundo de las Redes y Telecomunicaciones. En particular al Ingeniero David Moreno quien con su guía me ha ayudado a culminar con éxito el presente trabajo, gracias por la confianza brindada, al Ingeniero Oswaldo Martínez quien fue un gran mentor y amigo, muchas gracias por su confianza, al Ingeniero Franklin Moreno director de Escuela siempre un gran amigo y consejero, gracias por toda la ayuda a lo largo de mi trayecto estudiantil, augurándoles siempre grandes logros para sus vidas.

Erica Carolina.

## TABLA DE CONTENIDO

<b>RESUMEN</b> .....	xvii
<b>SUMMARY</b> .....	xviii
<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO I</b>	
<b>1. MARCO TEÓRICO REFERENCIAL</b> .....	4
<b>1.1. Ansiedad</b> .....	4
<i>1.1.1. Niveles de ansiedad</i> .....	4
<i>1.1.2. Tipos de ansiedad</i> .....	4
<i>1.1.3. Información estadística sobre salud mental</i> .....	5
<b>1.2. Tecnologías para la obtención de datos de un sistema de monitoreo</b> .....	6
<i>1.2.1. Sensores</i> .....	7
<i>1.2.1.1. Definición</i> .....	7
<i>1.2.1.2. Características</i> .....	7
<i>1.2.1.3. Funcionamiento</i> .....	8
<i>1.2.1.4. Tipos de sensores</i> .....	9
<i>1.2.2. Sensores ópticos</i> .....	10
<i>1.2.2.1. Principio de funcionamiento de los sensores ópticos</i> .....	11
<i>1.2.2.2. Clasificación de los sensores ópticos</i> .....	11
<b>1.3. Almacenamiento de datos</b> .....	13
<b>1.4. Plataformas de Hardware libre para la ejecución del sistema de monitoreo de la actividad ansiolítica</b> .....	13
<i>1.4.1. Plataformas existentes en el mercado</i> .....	13
<i>1.4.1.1. Plataforma Pingüino</i> .....	13
<i>1.4.1.2. Plataforma Waspmote</i> .....	14
<i>1.4.1.3. Plataforma Spark</i> .....	15
<i>1.4.1.4. Plataforma Intel Galileo</i> .....	15
<i>1.4.1.5. Plataforma Arduino</i> .....	16

1.4.2.	<i>Tabla comparativa de las Plataformas existentes</i> .....	16
1.5.	<b>Tecnologías para la transmisión inalámbrica de datos.</b> .....	17
1.6.	<b>Modelos animales para el estudio de la ansiedad.</b> .....	19
1.6.1.	<i>Pruebas condicionadas</i> .....	19
1.6.2.	<i>Pruebas no condicionadas</i> .....	19
1.6.2.1.	<i>Laberinto en Cruz Elevado (EPM)</i> .....	20
1.6.2.2.	<i>Caja Luz/Oscuridad (MCL)</i> .....	20
1.6.2.3.	<i>Test de la Chimenea.</i> .....	21
1.6.2.4.	<i>Prueba de Campo Abierto</i> .....	22
<b>CAPÍTULO II</b>		
2.	<b>MARCO METODOLÓGICO</b> .....	24
2.1.	<b>Metodología de la investigación</b> .....	24
2.1.1.	<i>Tipo de investigación</i> .....	24
2.1.2.	<i>Método de investigación</i> .....	24
2.1.2.1.	<i>Métodos teóricos</i> .....	24
2.1.2.2.	<i>Métodos empíricos</i> .....	25
2.1.3.	<i>Técnicas de investigación</i> .....	25
2.2.	<b>Diseño del sistema de monitoreo de la actividad ansiolítica</b> .....	26
2.2.1.	<i>Esquema general del sistema</i> .....	26
2.2.2.	<i>Diseño del sistema</i> .....	26
2.3.	<b>Requerimiento para el diseño de los sistemas de monitoreo.</b> .....	28
2.3.1.	<i>Bloque de entrada</i> .....	28
2.3.1.1.	<i>Sensores</i> .....	28
2.3.1.2.	<i>Sensor proximidad</i> .....	29
2.3.1.3.	<i>Sensor Infrarrojo Sharp GP2Y0A21YK0F</i> .....	30
2.3.1.4.	<i>Sensor Infrarrojo Sharp GP2Y0A60SZLF</i> .....	30
2.3.1.5.	<i>Método de triangulación sensores Sharp</i> .....	31
2.3.2.	<i>Bloque de procesamiento</i> .....	32
2.3.2.1.	<i>Hardware libre</i> .....	32



2.3.2.2.	<i>Plataforma Arduino</i> .....	33
2.3.2.3.	<i>Arduino Uno</i> .....	34
2.3.2.4.	<i>Distribución de pines Arduino Uno</i> .....	34
2.3.2.5.	<i>Arduino Mega 2560</i> .....	37
2.3.2.6.	<i>Distribución de los Pines Arduino Mega 2560</i> .....	37
2.3.2.7.	<i>Tecnología inalámbrica</i> .....	41
<b>2.3.3.</b>	<b><i>Bloque de almacenamiento</i></b> .....	42
<b>2.3.4.</b>	<b><i>Interfaz de usuario</i></b> .....	42
<b>2.4.</b>	<b>Fuente de alimentación</b> .....	44
<b>2.4.1.</b>	<b><i>Adaptador de Corriente</i></b> .....	44
<b>2.5.</b>	<b>Implementación</b> .....	45
<b>2.5.1.</b>	<b><i>Modelo Claro/Oscuro (LDM)</i></b> .....	45
2.5.1.1.	<i>Bloque de entrada</i> .....	45
2.5.1.2.	<i>Bloque de procesamiento</i> .....	47
2.5.1.3.	<i>Bloque de almacenamiento</i> .....	49
2.5.1.4.	<i>Bloque de interfaz de usuario</i> .....	50
<b>2.5.2.</b>	<b><i>Laberinto en cruz elevado (EPM)</i></b> .....	53
2.5.2.1.	<i>Bloque de entrada</i> .....	53
2.5.2.2.	<i>Bloque de procesamiento</i> .....	55
2.5.2.3.	<i>Bloque de almacenamiento</i> .....	56
2.5.2.4.	<i>Bloque de interfaz de usuario</i> .....	56
<b>2.5.3.</b>	<b><i>Protección física de los sistemas</i></b> .....	58
<b>CAPÍTULO III</b>		
<b>3.</b>	<b>EVALUACIÓN Y COMPARACIÓN DE RESULTADOS</b> .....	59
<b>3.1.</b>	<b>Introducción</b> .....	59
<b>3.2.</b>	<b>Pruebas y evaluación de los sistemas de monitoreo</b> .....	60
<b>3.2.1.</b>	<b><i>Pruebas de transmisión de datos</i></b> .....	60
<b>3.2.2.</b>	<b><i>Recepción de datos en la aplicación de Matlab</i></b> .....	61
3.2.2.1.	<i>Aplicación sistema de monitoreo MCL/EPM</i> .....	61

<b>3.3.</b>	<b>Pruebas de campo .....</b>	<b>63</b>
<b>3.3.1.</b>	<b><i>Evaluación modelo claro/oscurο .....</i></b>	<b>64</b>
<b>3.3.2.</b>	<b><i>Evaluación laberinto cruz elevado .....</i></b>	<b>67</b>
<b>3.4.</b>	<b>Banco de pruebas del sistema de monitoreo .....</b>	<b>70</b>
<b>3.4.1.</b>	<b><i>Análisis del modelo claro oscuro.....</i></b>	<b>70</b>
<b>3.4.2.</b>	<b><i>Análisis del laberinto en cruz elevado .....</i></b>	<b>71</b>
<b>3.4.3.</b>	<b><i>Comparación de las variables del sistema automatizado con el sistema tradicional</i></b>	<b>73</b>
<b>3.5.</b>	<b>Presupuesto.....</b>	<b>74</b>
	<b>CONCLUSIONES.....</b>	<b>75</b>
	<b>RECOMENDACIONES.....</b>	<b>76</b>
	<b>BIBLIOGRAFÍA</b>	
	<b>ANEXOS</b>	

## ÍNDICE DE TABLAS

<b>Tabla 1-1:</b> Porcentaje de ansiedad en Latinoamérica.....	6
<b>Tabla 2-1:</b> Simbolización de los sensores .....	9
<b>Tabla 3-1:</b> Comparación entre Plataformas de Hardware Libre .....	17
<b>Tabla 4-1:</b> Pruebas más usadas para medir ansiedad. ....	22
<b>Tabla 1-2:</b> Bloques de Proceso del Sistema de Monitoreo. ....	27
<b>Tabla 2-2:</b> Cuadro comparativo sensores de proximidad.....	29
<b>Tabla 3-2:</b> Modelos existentes de Arduino. ....	33
<b>Tabla 4-2:</b> Comparación de módulos inalámbricos .....	41
<b>Tabla 5-2:</b> Distribución de Pines módulo NRF24L01 .....	48
<b>Tabla 1-3:</b> Nomenclatura de las ratas a utilizar.....	63
<b>Tabla 2-3:</b> Conjunto de individuos para la evaluación ansiolítica. ....	64
<b>Tabla 3-3:</b> Análisis Modelo Claro/Oscuro (Ligth/Dark Model) .....	70
<b>Tabla 4-3:</b> Análisis Laberinto en Cruz Elevado (EPM) .....	72
<b>Tabla 5-3:</b> Comparación de Variables. Forma automatizada vs Forma Tradicional.....	73
<b>Tabla 6-3:</b> Presupuesto Sistemas de Monitoreo. ....	74

## ÍNDICE DE FIGURAS

<b>Figura 1-1:</b> Diagrama de bloques de un sensor.....	8
<b>Figura 2-1:</b> Sensores por el tipo de salida.....	9
<b>Figura 3-1:</b> Sensor óptico.....	10
<b>Figura 4-1:</b> Sensor de Barrera.....	11
<b>Figura 5-1:</b> Sensor Reflex.....	12
<b>Figura 6-1:</b> Sensor Reflex Polarizado.....	12
<b>Figura 7-1:</b> Sensor Difuso.....	13
<b>Figura 8-1:</b> Board Pingüino OTG.....	14
<b>Figura 9-1:</b> Componentes módulo Waspnote.....	14
<b>Figura 10-1:</b> Placa core Spark.....	15
<b>Figura 11-1:</b> Placa Intel Galileo.....	15
<b>Figura 12-1:</b> Placa Arduino.....	16
<b>Figura 13-1:</b> Distribución de Redes Inalámbricas.....	18
<b>Figura 14-1:</b> Laberinto en Cruz Elevado.....	20
<b>Figura 15-1:</b> Modelo claro/oscurο.....	21
<b>Figura 16-1:</b> Test de la Chimenea.....	21
<b>Figura 17-1:</b> Modelo de Campo Abierto.....	22
<b>Figura 1-2:</b> Esquema General del Sistema.....	26
<b>Figura 2-2:</b> Bloques de Proceso del Sistema.....	27
<b>Figura 3-2:</b> Sensor Sharp 10-80cm.....	30
<b>Figura 4-2:</b> Sensor Sharp 10-150cm.....	30
<b>Figura 5-2:</b> Esquema de la resistencia IR ambiental.....	31
<b>Figura 6-2:</b> Método de Triangulación.....	31
<b>Figura 7-2:</b> Arduino Uno.....	34
<b>Figura 8-2:</b> Distribución Pines Analógicos (Arduino Uno).....	35
<b>Figura 9-2:</b> Pines Digitales (Arduino Uno).....	35
<b>Figura 10-2:</b> Pines de Comunicación (Arduino UNO).....	36
<b>Figura 11-2:</b> Pines PWM (Arduino Uno).....	36
<b>Figura 12-2:</b> Pines Power (Arduino Uno).....	37
<b>Figura 13-2:</b> Arduino Mega.....	37
<b>Figura 14-2:</b> Pines Analógicos (Arduino Mega).....	38
<b>Figura 15-2:</b> Pines Digitales (Arduino Mega).....	39
<b>Figura 16-2:</b> Pines de Comunicación (Arduino Mega).....	39
<b>Figura 17-2:</b> Pines PWM (Arduino Mega).....	40
<b>Figura 18-2:</b> Pines Power (Arduino Mega).....	40

<b>Figura 19-2:</b> Interfaz Gráfica (GUIDE) Matlab.....	43
<b>Figura 20-2:</b> Formas de Alimentación Arduino.....	44
<b>Figura 21-2:</b> Adaptador conectado a Arduino .....	44
<b>Figura 22-2:</b> Implementación de sensores (Bloque de Entrada).....	45
<b>Figura 23-2:</b> Disposición de los sensores. Vista 2D. ....	46
<b>Figura 24-2:</b> Disposición de sensores. Vista superior en 3D. ....	46
<b>Figura 25-2:</b> Arduino IDE.....	47
<b>Figura 26-2:</b> Código Transmisor Uno.....	47
<b>Figura 27-2:</b> Conexión módulo NRF24L01 a Arduino Uno.....	49
<b>Figura 28-2:</b> Conexión Transmisor LDM.....	49
<b>Figura 29-2:</b> Almacenamiento de datos LDM. ....	50
<b>Figura 30-2:</b> Receptor NRF24L01.....	50
<b>Figura 31-2:</b> Interfaz de Usuario del sistema LDM.....	51
<b>Figura 32-2:</b> Matlab Compiler .....	51
<b>Figura 33-2:</b> Diagrama de Flujo LDM.....	52
<b>Figura 34-2:</b> Implementación de sensores Sharp GP2Y0A60SZLF (Bloque de Entrada).....	53
<b>Figura 35-2:</b> Ubicación de los sensores en EPM. Vista 2D.....	54
<b>Figura 36-2:</b> Vista Superior ubicación de los sensores en EPM, plano 3D. ....	54
<b>Figura 37-2:</b> Código Transmisor Mega .....	55
<b>Figura 38-2:</b> Conexión Transmisor Mega.....	55
<b>Figura 39-2:</b> Bloque de Almacenamiento EPM.....	56
<b>Figura 40-2:</b> Interfaz de Usuario EPM. ....	56
<b>Figura 41-2:</b> Diagrama de Flujo EPM. ....	57
<b>Figura 42-2:</b> Caja protectora. ....	58
<b>Figura 1-3:</b> Ubicación sistemas de monitoreo. ....	60
<b>Figura 2-3:</b> Transmisión de datos desde el Arduino Mega. ....	60
<b>Figura 3-3:</b> Recepción de datos mediante comunicación serial.....	61
<b>Figura 4-3:</b> Aplicación MCL/EPM con sus respectivos paneles. ....	62
<b>Figura 5-3:</b> Ventana Emergente botón Guardar.....	63
<b>Figura 6-3:</b> Escenario para pruebas ansiolíticas.....	64
<b>Figura 7-3:</b> Resultados obtenidos en k1.....	65
<b>Figura 8-3:</b> Resultados Obtenidos en k2.....	65
<b>Figura 9-3:</b> Resultados Obtenidos en k3.....	66
<b>Figura 10-3:</b> Resultados Obtenidos en k4.....	66
<b>Figura 11-3:</b> Resultados Obtenidos en k5.....	67
<b>Figura 12-3:</b> Resultados obtenidos en k1 (EPM).....	67
<b>Figura 13-3:</b> Resultados obtenidos en k2 (EPM).....	68

<b>Figura 14-3:</b> Resultados obtenidos en k3 (EPM).....	68
<b>Figura 15-3:</b> Resultados obtenido en k4 (EPM) .....	69
<b>Figura 16-3:</b> Resultados obtenidos en k5 (EPM).....	69

## ÍNDICE DE ABREVIATURAS

<b>CE:</b>	Chip Enable
<b>CSN:</b>	Chip Select
<b>EPM:</b>	Elevated Plus Maze (Laberinto en Cruz Elevado)
<b>FDMA:</b>	Frequency Division Multiple Access (Acceso Múltiple por División de Frecuencia)
<b>FDD:</b>	Frequency Division Duplexing (Duplexado por División de Frecuencia)
<b>GND:</b>	Ground (tierra)
<b>GPRS:</b>	General Packet Radio Service (Servicio General de Paquetes Vía Radio)
<b>GSM:</b>	Global System for Mobile Communications
<b>GUI:</b>	Graphical User Interface
<b>GUIDE:</b>	Graphical User Interface Development Environment
<b>IEEE:</b>	Institute of Electrical and Electronics Engineers (Instituto de Ingeniería Eléctrica y Electrónica)
<b>INSPI:</b>	Instituto Nacional de Investigación en Salud Pública
<b>IRQ:</b>	Interrupt Request (Petición de Interrupción)
<b>IoT:</b>	Internet of Things (Internet de las cosas)
<b>IP:</b>	Internet Protocol (Protocolo de Internet)
<b>IR:</b>	Infrared (Infrarrojo)
<b>JST:</b>	Japanese Solderless Terminal (Terminal Japonesa sin Soldadura)
<b>LDM:</b>	Light-Dark Model. (Modelo Claro/Oscuro)
<b>LTE:</b>	Long Term Evolution (Evolución a Largo Plazo)
<b>MISO:</b>	Master Input Slave Output (Entrada Maestra Salida Esclavo)
<b>MOSI:</b>	Master Output Slave Input (Salida Maestra Entrada Esclavo)
<b>OMS:</b>	Organización Mundial de la Salud

<b>PSD:</b>	Position Sensitive Detector (Detector de Posición)
<b>RF:</b>	Radio Frecuencia
<b>RXD:</b>	Receive data (Receptor)
<b>TDMA:</b>	Time Division Multiple Access (Acceso Múltiple por División de Tiempo)
<b>TXD:</b>	Transmit data (Transmisor)
<b>USB:</b>	Universal Serial Bus (Bus Universal en Serie)
<b>VCC:</b>	Voltaje de Corriente Continua
<b>WIFI:</b>	Wireless Fidelity (Fidelidad Inalámbrica)
<b>WiMAX:</b>	Worldwide Interoperability for Microwave Access (Interoperabilidad Mundial para Acceso por Microondas)
<b>WPAN:</b>	Wireless Personal Area Network (Red Inalámbrica de Área Personal)



## RESUMEN

Para el presente trabajo de titulación se diseñó e implementó dos sistemas de monitoreo para la evaluación de la actividad ansiolítica in vivo en ratas. Estos sistemas posibilitaron visualizar en una aplicación para cada uno respectivamente creadas por la autora llamadas Claro/Oscuro y Laberinto, diversos eventos a medir. Estos sistemas se desarrollaron por bloques de: entrada, procesamiento, almacenamiento y salida (Interfaz de usuario). En el primer bloque con la ayuda de los sensores Sharp se recolectaron datos de distancias, el segundo bloque procesó los datos obtenidos los mismos que se transmiten mediante el módulo NRF24L01 a su receptor, para este bloque se hizo uso de los Arduinos Mega y Uno siendo esta la placa principal; en el tercer bloque se creó vectores para el almacenamiento de las medidas emitidas, en este punto se hizo uso del Software Matlab específicamente de la interfaz gráfica para procesar la información recibida por el módulo receptor NRF24L01 y poder ser visualizada en el cuarto bloque, en la etapa final la aplicación fue programada para mostrar adicionalmente a los datos ya mencionados una gráfica de los desplazamientos. De las pruebas realizadas se concluye que el tiempo de sincronización entre las dos plataformas es de 2 segundos. Presenta un error en los cruces de 0,467 puntos en el LDM y 0,333 puntos en EPM debido a la sensibilidad de los sensores. Se concluye que los sistemas implementados se pueden convertir en una herramienta de ayuda para el desarrollo de las pruebas ansiolíticas. Es recomendable hacer uso de visión artificial para realizar procesamiento de imagen con la ayuda de un software especializado así como el uso de una sola plataforma de hardware con mayor capacidad de transmisión para evitar interrupciones en el envío de datos.

**PALABRAS CLAVES:** <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <TELECOMUNICACIONES>, <INGENIERÍA Y TECNOLOGÍA ELECTRÓNICA>, <ARDUINO (SOFTWARE-HARDWARE)>, <RATAS (RATTUS NORVEGICUS)>, <MÓDULO TRANSCEIVER NRF24L01>, <MONITOREO ACTIVIDAD ANSIOLÍTICA>, <TRANSMISIÓN DE DATOS>

## SUMMARY

For the present work, two monitoring systems were designed and implemented for the evaluation of anxiolytic activity in vivo in rats. These systems made it possible to visualize an application for each one respectively, they were created by the author called Clear/Dark and Labyrinth, various events to be measured. These systems were developed by blocks of: input, processing, storage and output (user interface). In the first block with the help of the Sharp sensors, distance data was collected, the second block processed the data obtained, which are transmitted through the NRF24L01 module to its receiver, for this block the Mega and Uno Arduinos were used. Being this the main board; in the third block, vectors were created for the storage of the emitted measurements, at this point the Matlab Software was used specifically for the graphical interface to process the information received by the receiver module NRF24L01 and be able to be visualized in the fourth block, in the final stage the application was programmed to show in addition to the data already mentioned a graph of the displacements. From the tests carried out, it is concluded that the synchronization time between the two platforms is 2 seconds. It has an error in the crosses of 0.467 points in the LDM and 0.333 points in EPM due to the sensitivity of the sensors. It is concluded that the implemented systems can become an aid tool for the development of anxiolytic tests. It is advisable to use artificial vision to perform image processing with the help of specialized software as well as the use of a single hardware platform with greater transmission capacity to avoid interruptions in sending data.

**KEY WORDS:** <TECHNOLOGY AND SCIENCE OF ENGINEERING>, <TELECOMMUNICATIONS>, <ELECTRONIC ENGINEERING AND TECHNOLOGY>, <ARDUINO (SOFTWARE-HARDWARE)>, <RATS (RATTUS NORVEGICUS)>, <MODULE TRANSCEIVER NRF24L01>, <MONITORING ANXIOLYTIC ACTIVITY >, <DATA TRANSMISSION>

# INTRODUCCIÓN

## ANTECEDENTES

Los modelos animales han sido una herramienta clave en la investigación, pues han permitido esclarecer procesos fisiológicos, bioquímicos, conductuales y demás, que se asemejan a los procesos respectivos en el ser humano. Lo que ha permitido alcanzar una mayor comprensión en gran variedad de campos, así como predicciones más acertadas en cuanto a respuestas del organismo ante diversos estímulos, ambientes y tratamientos. (Hau, 2003).

La problemática a nivel de predicciones de los modelos animales se basa, en su mayor parte, en arriesgadas interpretaciones de los resultados obtenidos. Es fundamental, para no cometer tales errores, conocer las limitaciones de cada modelo, tales como las diferencias entre especies, la intervención de otros factores no asociados con la condición bajo estudio, la falta de un modelo que nos replique la totalidad de las características presentadas en el ser humano, además de las dificultades de replicar funciones humanas complejas. (Shanks, 2009)

En el año 2014 el Instituto Nacional de Investigación en Salud Pública (INSPI) adquirió un biomódulo que fortalecerá la investigación en ciencias biomédicas en el Ecuador. Esta unidad permitirá realizar experimentaciones preclínicas de nuevos fármacos así como estudios sobre parásitos y su comportamiento. (Bustamante, 2014)

En Ecuador existen 76 universidades, de las cuales solo cuatro ofertan carreras afines al uso de un Bioterio, de estas la Universidad Técnica de Machala desde junio del 2016 cuenta con un Biomódulo-Bioterio de alta tecnología, el mismo que cuenta con los equipos necesarios para la realización de estudios preclínicos y de prospección de productos desarrollados desde productos naturales. (Opinión, 2016)

En la Escuela Superior Politécnica existe un Bioterio, el cual no cuenta con el equipamiento necesario para llevar a cabo estudios preclínicos basados en el uso de productos de extractos

naturales, debido a los altos costos de los equipos, la falta de apoyo a estas áreas de investigación, la realización de pruebas de forma manual las mismas que son propensas a errores, surgió la necesidad de desarrollar e implementar sistemas de evaluación y monitoreo empleando dispositivos electrónicos de bajo costo que optimicen la realización de este tipo de pruebas, los resultados que se obtengan sirven para estimar la aceptación a ciertos químicos que facilitan el estilo de vida de las personas.

Con estos antecedentes, el presente trabajo de titulación busca diseñar e implementar dos sistemas de monitoreo para la evaluación de la actividad ansiolítica in vivo en ratas (*rattus norvegicus*), se realizarán dos pruebas, una de las pruebas a desarrollar es <<prueba anti-ansiedad (Modelo Claro-Oscuro)>>, que consiste en un compartimiento de luz y un compartimiento oscuro, dividida por una zona equipada con sensores infrarrojos, la segunda prueba a realizar es un <<Laberinto En Cruz Elevado>>, que consiste en un laberinto en forma de cruz, en el cual dos extremos son cerrados y los otros dos abiertos, el sistema es equipado con sensores de luz, en este punto cabe indicar que la prueba no es válida si la rata solo cruza únicamente sus dos patas, (Miranda, 2009) la investigación busca como elemento adicional constituirse como una herramienta de soporte para las actividades realizadas por los docentes y estudiantes permitiéndoles obtener datos relevantes que avalen su investigación.

Al trabajar con módulos y sensores inteligentes se requiere de un control que abarque estos sistemas electrónicos y que permita determinar el funcionamiento que requieren los sistemas de monitoreo a desarrollar, el diseño se puede llevar a cabo con el uso de plataformas de hardware libre debido a la compatibilidad que existe con estos dispositivos electrónicos. La unión de estas tecnologías hizo que sea viable el proyecto de desarrollar dos sistemas de monitoreo para evaluar la actividad ansiolítica in vivo en ratas, de tal forma que estos sistemas sean capaces de emitir resultados reales y confiables una vez transcurrido el tiempo de duración de cada prueba.

Conociendo de la inexistencia del equipamiento necesario en el Bioterio de la ESPOCH, el presente trabajo tiene como objetivo desarrollar e implementar un sistema de monitoreo para la evaluación de la actividad ansiolítica in vivo en ratas (*rattus norvegicus*). Se planteó para la investigación como objetivos específicos:

- Definir los requerimientos del sistema electrónico a desarrollar.

- Estudiar y seleccionar los elementos electrónicos, tecnologías de comunicación que mejor se adaptan a los requerimientos del sistema electrónico.
- Permitir acceder a la información recolectada desde una pc.
- Realizar las pruebas respectivas de evaluación del sistema a desarrollar.
- Evaluar el funcionamiento sistema de monitoreo de forma tradicional y automatizada.

## CAPÍTULO I

### 1. MARCO TEÓRICO REFERENCIAL

#### 1.1. Ansiedad

La ansiedad es un término proveniente del latín *anxietas*, se define como una respuesta fisiológica normal ante situaciones nuevas, amenazantes o intensas propias del ambiente, la Real Academia de la Lengua Española la define como << *Estado de agitación, inquietud o zozobra del ánimo*>>, en términos más generales la ansiedad normal le permite al individuo adaptarse al entorno que lo rodea, en cambio la ansiedad patológica es el desequilibrio entre el individuo y el ambiente. (Lorenzo et al., 2015)

##### 1.1.1. Niveles de ansiedad

Se ha distinguido cuatro niveles de ansiedad, las cuales se detallan a continuación.

- ***Ansiedad como sentimiento:*** se define como un estímulo normal producido por persuasiones internas como recuerdos, fantasías, problemas o pensamientos.
- ***Ansiedad como síntoma:*** es parte de cualquier enfermedad psiquiátrica siendo un componente psicológico de los trastornos de ansiedad (ej. “presiento que me van a dar una mala noticia”, “nerviosismo”, etc.)
- ***Ansiedad como síndrome:*** constituye un conjunto de síntomas físicos, psicológicos, conductuales influidos por mecanismos neurobiológicos asociados a diversos factores.
- ***Ansiedad como enfermedad:*** en este nivel el cuadro clínico es visiblemente definido, por tanto posee su propia etiopatogenia, síntomas y tratamiento. (Idrobo, 2016)

##### 1.1.2. Tipos de ansiedad

Se destacan seis tipos de ansiedad:

**1. Ansiedad generalizada:** catalogada como leve, la padecen aquellas personas que por lo general se preocupan por situaciones cotidianas de la vida.

**2. Ansiedad relacionada con el estrés:** se identifican factores que aparecen de manera repentina provocando angustia ya sea por eventos traumáticos, un examen, el pago inmediato de deudas, entre otros, por lo general este tipo de ansiedad la padecen estudiantes y trabajadores.

**3. Crisis de pánico:** poseen una duración de entre 3 a 10 minutos provocando la sensación de muerte o crisis existencial. La sensación de angustia puede durar incluso días pero la crisis como tal no dura más de 15 minutos.

**4. Fobias sociales:** conocido también como pánico escénico, se lo puede evidenciar en personas que tienen que hablar en público o abordar a algún desconocido. Este tipo de personas son más introvertidas por tanto les cuesta hacer amistades o relacionarse con los demás.

**5. Alteraciones médicas o medicamentos que generan síntomas de ansiedad:** hay enfermedades y medicamentos que provocan ansiedad, entre este tipo de patologías se encuentran síntomas pre menstruales, enfermedades endocrinas, enfermedades neurológicas degenerativas, alcoholismo, etc., drogas como la cafeína, esteroides, descongestionantes nasales.

**6. Síntomas de ansiedad que forman parte de una enfermedad mental:** por lo general van acompañados de trastornos mentales como esquizofrenia, depresión, demencia, incluso la anorexia y bulimia. (Idrobo, 2016)

### **1.1.3. Información estadística sobre salud mental**

Según la Organización Mundial de la Salud (OMS), Ecuador ocupa el undécimo lugar de países con problemas de depresión y el décimo por casos de trastornos de ansiedad en América Latina. Según la OMS en la última década los casos de personas que sufren algún trastorno de ansiedad a nivel mundial ha incrementado en un 15 por ciento, al menos 264 millones de personas padecen trastornos de ansiedad.

Esta patología la padecen más las mujeres (4,6 por ciento) que los hombres (2,6 por ciento). En Latinoamérica se estima que el 7,7 por ciento de mujeres padecen ansiedad mientras que los

hombres en un 3,6 por ciento. (Mejía, 2017). En la *Tabla 1-1* se detalla el porcentaje de personas con ansiedad en América Latina.

**Tabla 1-1:** Porcentaje de ansiedad en Latinoamérica

N°	País	Porcentaje (%)
1	Brasil	9,3
2	Paraguay	7,6
3	Chile	6,5
4	Uruguay	6,4
5	Argentina	6,3
6	Cuba	6,1
7	Colombia	5,8
8	Perú	5,7
9	República Dominicana	5,7
10	Ecuador	5,6
11	Bolivia	5,4
12	Costa Rica	4,6
13	El Salvador	4,6
14	Nicaragua	4,5
15	Panamá	4,5
16	Venezuela	4,4
17	Honduras	4,3
18	Guatemala	4,3
19	México	3,6

**Realizado por:** Adriano, Erica, 2018

**Fuente:** <https://www.redaccionmedica.ec/secciones/salud-publica/ecuador-entre-los-pa-ses-con-m-s- casos-de-depresi-n-en-latinoam-rica-89705>

## 1.2. Tecnologías para la obtención de datos de un sistema de monitoreo

Para obtener los datos de un sistema de monitoreo, se requiere del uso de sensores específicos, los mismos que aporten con todos los eventos necesarios para la realización del proyecto.



## **1.2.1. Sensores**

### *1.2.1.1. Definición*

“Los sensores son dispositivos electrónicos que nos permiten interactuar con el entorno, de forma que nos proporcionan información de ciertas variables que nos rodean para poder procesarlas y así generar órdenes o activar procesos.” (Serna, Ros, y Rico, 2010, p. 3).

Los sensores repiten la capacidad de apreciación de los seres humanos, por ello es cada vez más habitual encontrarlos en diferentes áreas tecnológicas, denotando los ejemplos más sobresalientes en el área de domótica, sistemas de monitoreo, sistemas vehiculares, dispositivos móviles.

### *1.2.1.2. Características*

Estos dispositivos dependen de la variable a medir, pero en términos generales sus características se las puede agrupar en dos grupos:

*Características estáticas*, las cuales describen el comportamiento del sensor en un estado permanente o con variaciones muy lentas en base a la variable a medir.

*Características dinámicas*, estas en cambio describen el comportamiento del sensor en un tiempo transitorio.

#### *1.2.1.2.1. Características estáticas*

- Rango de medida: conjunto de valores que la señal de entrada puede tomar en base a un valor máximo y mínimo con una tolerancia de error aceptable.
- Resolución: discierne entre valores muy próximos a la señal de entrada.
- Precisión: indica la diferenciación máxima entre la señal real obtenida y la señal teórica como una pauta para el sensor.

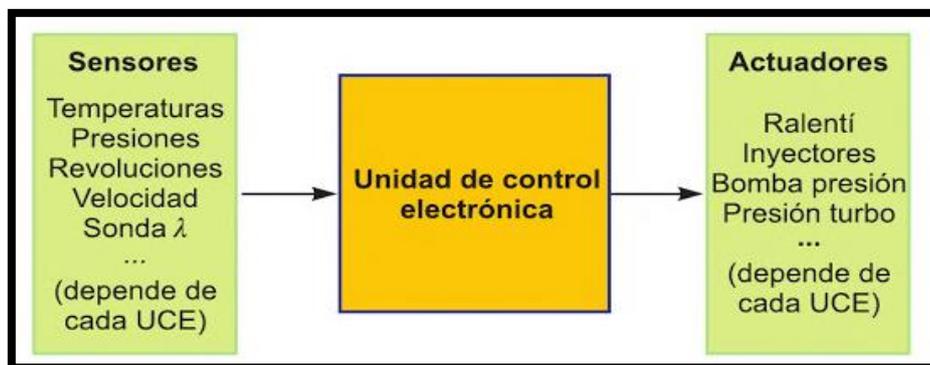
- Repetitividad: permite conocer la máxima variación existente entre valores de salida, siendo estos medidos varias veces por el mismo sensor.
- Sensibilidad: hace referencia a la mayor o menor diferenciación de la señal de salida por unidad de la magnitud de entrada.
- Ruido: cualquier perturbación que afecte a la señal a medir.

#### 1.2.1.2.2. Características Dinámicas

- Velocidad de respuesta: determina la capacidad del sensor a emitir su respuesta sin retraso.
- Estabilidad: indica la desviación de la señal de salida con respecto a valores teóricos, al variar parámetros externos que no son parte de la medida como alimentación, etc. (Vasquez, 2012)

#### 1.2.1.3. Funcionamiento

Estos dispositivos están adecuados para percibir cualquier señal externa, esta señal es enviada a la unidad de control electrónica, tal como se indica en la *Figura 1-1*, la misma que se encarga de procesar la información para que pueda ser emitida a los actuadores, cuya función es cumplir con los parámetros establecidos para que el sistema funcione correctamente.



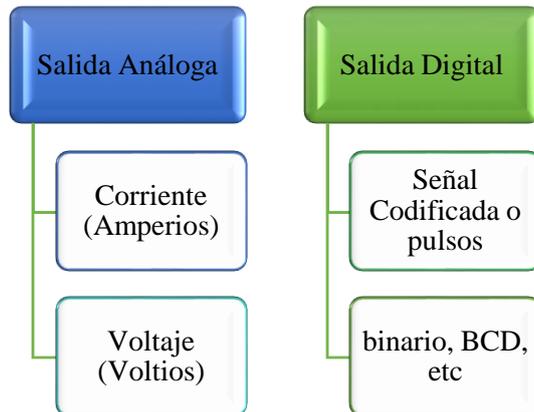
**Figura 1-1:** Diagrama de bloques de un sensor

**Fuente:** (Barrera, O., & Ros, J., (2016). Sensores y actuadores. En Sistemas eléctricos y de seguridad y confortabilidad, (p.204).)

#### 1.2.1.4. Tipos de sensores

En el mercado existe gran variedad de sensores, los mismos que pueden clasificar básicamente en dos grupos, uno de ellos es en función del tipo de señal de salida, y el otro según el tipo de magnitud física a detectar.

##### 1.2.1.4.1. Según el tipo de salida



**Figura 2-1:** Sensores por el tipo de salida

Realizado por: Adriano, E. 2017

Fuente: (Barrera, O., & Ros, J., (2016). Sensores y actuadores. En Sistemas eléctricos y de seguridad y confortabilidad, (p.204).)

##### 1.2.1.4.2. Según el tipo de magnitud física a detectar

Este tipo de sensores abarca una gran variedad de variables a detectar como: posición, proximidad, velocidad, fuerza, temperatura, humedad, entre otros, los mismos que se pueden observar en la *Tabla 2-1*.

**Tabla 2-1:** Simbolización de los sensores

Magnitud Física	Transductor	Característica
<b>Posición (Lineal o Angular)</b>	Finales de carrera	Todo o Nada
	Potenciómetro	Analógico
	Encoders (absolutos/incrementales)	Digital
<b>Pequeños Desplazamientos</b>	Transformador Diferencial (LVDT)	Analógico
	Galgas Extensiométricos	Analógico
<b>Velocidad (Lineal o Angular)</b>	Dínamo Tacométrica	Analógico
	Encoder	Digital
	Detectro Inductivo u Óptico	Digital
<b>Aceleración</b>	Acelerómetro	Analógico
	Sensor de Velocidad+ Calculador	Digital

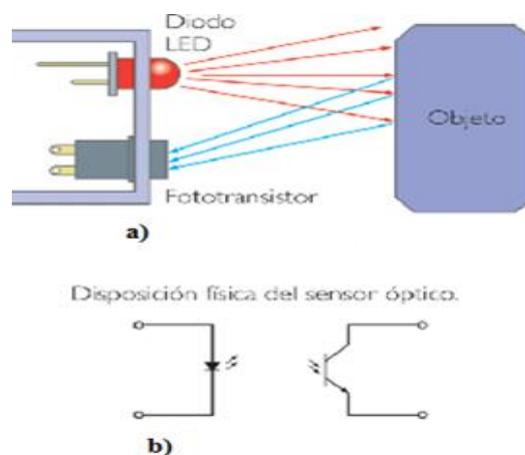
<b>Fuerza y par</b>	Galga Extensiométrica	Analógico
<b>Nivel</b>	Flotador+ Detector desplazamiento	Analógico
	Capacitivos	Analógico
	Ultrasonido	Digital
<b>Presión</b>	Membrana+ Detector desplazamiento	Analógico
	Piezoeléctricos	Analógico
<b>Caudal</b>	De Turbina	Analógico
	Magnéticos	Analógico
<b>Temperatura</b>	Termostatos	Todo o Nada
	Termopar	Analógico
	Resistencias TP 100	Analógico
	Termistores CTN	Analógico
	Termistores CTP	Todo-Nada
<b>De Presencia o Proximidad</b>	Inductivos	Analógico o Todo-Nada
	Capacitivos	Todo-Nada
	Ópticos	Analógico o Todo-Nada
	Ultrasónicos	Analógico

**Realizado por:** Adriano, E., 2017.

**Fuente:** (Fuentes, M., 2013, p.5).

### 1.2.2. Sensores ópticos

Conocidos como sensores fotoelectrónicos, son dispositivos que funcionan al cambio de la intensidad de luz. Requieren de un componente emisor que genera la luz (diodo led) y un componente receptor que recoge la luz emitida por el emisor (Serna, Ros, & Rico, 2010). Generalmente utilizados para determinar posición, como se puede observar en la *Figura 3-1*.



**Figura 3-1:** Sensor óptico.

**Fuente:** ([http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena11/4quincena11\\_contenidos\\_3g.htm](http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena11/4quincena11_contenidos_3g.htm))

Las fuentes de luz más frecuentes son: luz solar y ambiental, que pese a ser menos precisas su uso es únicamente para el control de encendido y apagado, las rojas, verdes y amarillas, son visibles y su uso se da para procesos industriales, detecciones de marcas, etc., finalmente las infrarrojas, son invisibles y generalmente su uso se da para localización en distancias largas y en presencia de polvo.

#### 1.2.2.1. Principio de funcionamiento de los sensores ópticos

“Liberación de cargas al incidir un haz de luz = Efecto Fotoeléctrico” (Vázquez, & García, 2015).

En base a la caracterización del dispositivo iluminado, se puede presentar los siguientes casos:

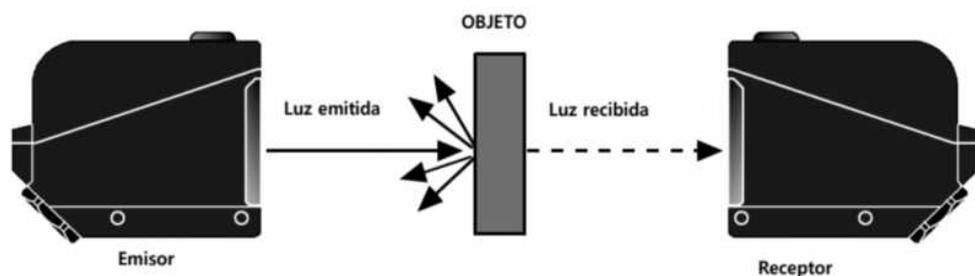
- Fotoconducción: diferenciación de la conductividad del material al incidir la luz.
- Efecto fotovoltaico: provee voltaje al existir radiación.
- Efecto emisor: generación de electrones al incurrir la luz.

#### 1.2.2.2. Clasificación de los sensores ópticos

Se pueden clasificar en cuatro categorías, dependiendo la manera en la que detectan los objetos.

##### 1.2.2.2.1. Barrera (Emisor- Receptor)

Tanto el emisor como el receptor se encuentran apuntando el uno al otro, detectan el objeto en base a la variación de luz que llega desde el emisor al receptor, como se puede observar en la *Figura 4-1*.

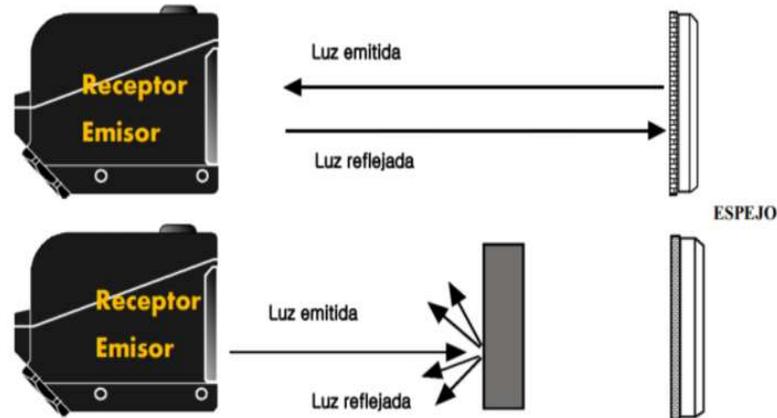


**Figura 4-1:** Sensor de Barrera.

Fuente: (Sensores y Controladores, 2009, p.4)

### 1.2.2.2. Retro reflectivo

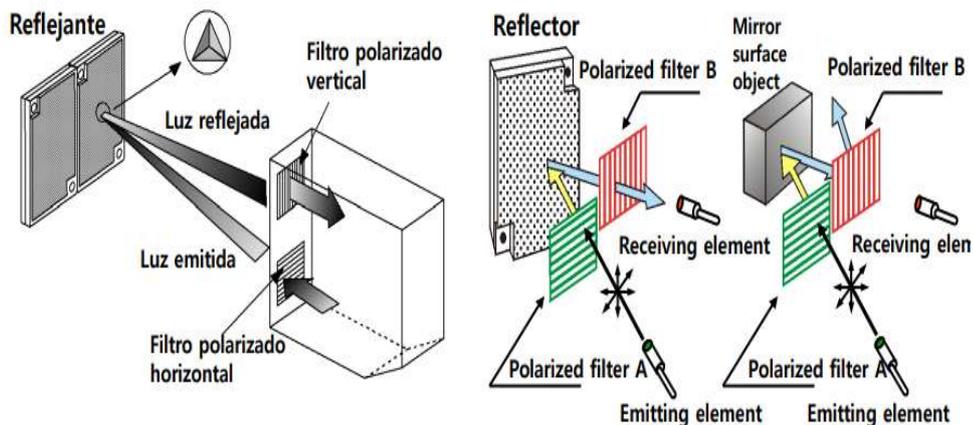
Usa un espejo especial (reflector), cuya particularidad es que devuelve la señal con el mismo ángulo con el que fue emitida, tal como se observa en la *Figura 5-1*.



**Figura 5-1:** Sensor Reflex  
Fuente: (Sensores y Controladores, 2009, p.4)

### 1.2.2.3. Retro reflectivo polarizado

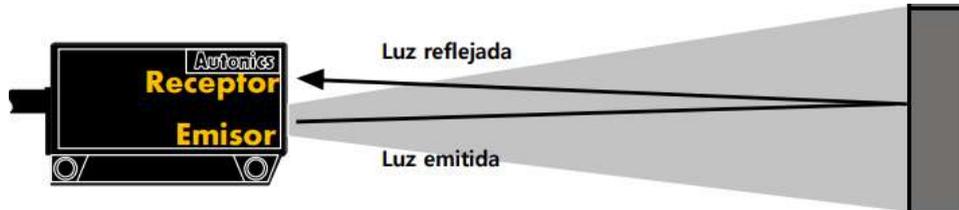
El emisor posee un lente que polariza la señal de luz en un sentido diferente al lente del receptor, que generalmente es a  $90^\circ$ . En la *Figura 6-1* se puede observar el funcionamiento del sensor Reflex Polarizado.



**Figura 6-1:** Sensor Reflex Polarizado.  
Fuente: (Sensores y Controladores, 2009, p.6)

#### 1.2.2.2.4. Difuso reflectivo

La luz reflejada del objeto es detectada directamente por el receptor, tal como se observa en la Figura 7-1.



**Figura 7-1:** Sensor Difuso  
*Fuente:* (Sensores y Controladores, 2009, p.7)

### 1.3. Almacenamiento de datos

Los datos obtenidos para el funcionamiento del sistema son añadidos a una plataforma para ser visualizados, debido a que estos son tomados en tiempo en real y después de un determinado tiempo los resultados están al alcance del usuario final. El uso de este sistema proporcionará un mejor ambiente al usuario, tendrá a su alcance resultados necesarios para validar su investigación.

### 1.4. Plataformas de Hardware libre para la ejecución del sistema de monitoreo de la actividad ansiolítica

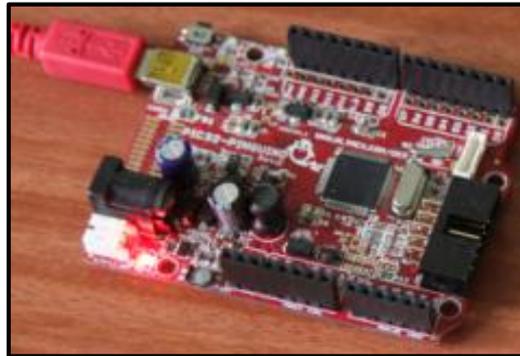
En el mercado existen diferentes tipos de plataformas, cada una de estas posee sus virtudes e inconvenientes, cada una de estas plataformas posee sus propias características para diferentes esquemas, además es necesario tomar en cuenta el ambiente en el cual se desarrollará el proyecto.

#### 1.4.1. Plataformas existentes en el mercado

##### 1.4.1.1. Plataforma Pingüino

Pingüino es una plataforma creada para el desarrollo sobre aplicaciones de microcontroladores, su filosofía de uso es más abierta, es decir todas sus librerías son abiertas, limitando su comunidad y en algunos casos las librerías no son totalmente portadas.

Posee su propia IDE basada en Phyton que se integra con el compilador SDCC y GCC, el software puede ser descargado desde algunos repositorios de Linux, por otra parte el hardware lo compone propiamente dicho la placa que será conectada a la Pc a través del cable USB. (Antuñez, s.f). En la *Figura 8-1* se puede observar la placa Pinguino de 32 bits.

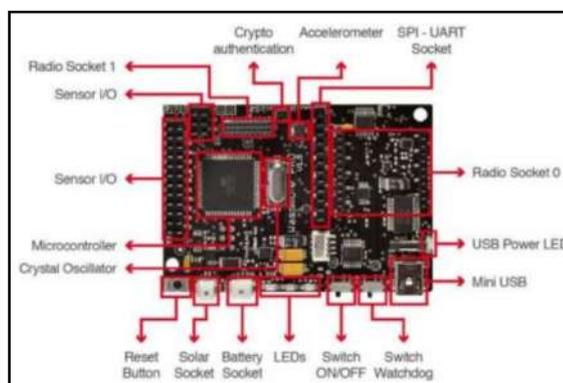


**Figura 8-1:** Board Pinguino OTG

**Fuente:** (<https://cc-electrognu.blogspot.com/2016/11/pre.html>)

#### 1.4.1.2. Plataforma Waspnote

Waspnote es el resultado de años de investigación por parte de la empresa Libelium, está orientada al diseño de redes de sensores, redes malladas y protocolos de redes inalámbricas. Posee un bajo consume de energía, ese ha sido su mayor reto, un consumo de solo 0.7 micro amperios en estado de reposo permite a las redes sensoriales vivir interrumpidamente por varios años (Reyes, 2014). En la placa de Waspnote se puede encontrar elementos como acelerómetro, batería, socket UART, entre otros, como se puede observar en la *Figura 9-1*. Como parte del hardware que se puede utilizar en Waspnote está módulo GPRS, placa de sensores, módulo Bluetooth, módulo Wifi, módulo GPS, módulo 3G y 4G, entre otros.



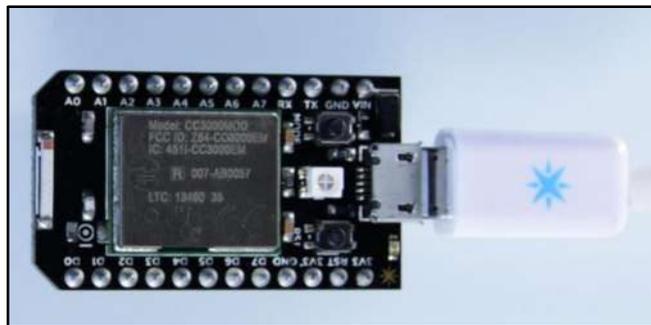
**Figura 9-1:** Componentes módulo Waspnote

**Fuente:** (<http://www.libelium.com/development/waspnote/documentation/waspnote-technical-guide/?action=download>)



#### 1.4.1.3. Plataforma Spark

Una pequeña placa de desarrollo Wifi que permite la creación de hardware a través del Internet, principalmente usado en IoT, lo que lo hace más interesante es el manejo de tres puntos de acceso diferentes, los usuarios principiantes pueden emplear la plataforma visual Tinker, en cambio para usuarios avanzados pueden emplear el IDE Spark basado en código abierto y para los más expertos pueden incurrir al uso de Tinkerer basando sus proyectos en código abierto para la exploración de cualquier aspecto de la placa y Software. En la *Figura 10-1* se puede observar la placa Spark. El software del Core Spark es compatible con iPhone, Android, Windows.



**Figura 10-1:** Placa core Spark

**Fuente:** (<https://docs.particle.io/datasheets/kits/>)

#### 1.4.1.4. Plataforma Intel Galileo

Galileo es una plataforma basada en el Microcontrolador Intel Quark, podría ser considerado como un Pentium de 32 bits compactado en un único chip, lo que lo hace interesante es que permite adentrarse a los procesadores basados en Intel, además de ser compatible con la plataforma de Arduino en su versión uno R3. En la *Figura 11-1* puede apreciarse un ejemplo de la placa Galileo. Entre sus características destacan que opera a los 400Mhz, 8 MB de memoria, reloj en tiempo real, conector Ethernet, entre otros.



**Figura 11-1:** Placa Intel Galileo

**Fuente:** (<https://www.xataka.com/makers/intel-galileo-placa-de-desarrollo-fruto-de-la-colaboracion-entre-intel-y-arduino>)

#### 1.4.1.5. Plataforma Arduino

Arduino es una gama de circuitos electrónicos de código abierto de hardware y software flexible y fácil de usar, como se puede observar en la *Figura 12-1*, ofrece un entorno de desarrollo intuitivo que permite la creación de diversos programas de una manera sencilla. Esta tarjeta es compatible con los principales sistemas operativos como lo son Windows, Mac OS X y Linux. Actualmente existe una gran variedad de tarjetas Arduino cada una con sus ventajas e inconvenientes, los mismos que se deberán analizar para elegir la tarjeta que se adapte con el proyecto que se desea realizar.

Para acceder a la plataforma Arduino se lo puede hacer de dos formas: ensamblado a mano, es decir el usuario se descarga la circuitería de la placa y crea su propia placa, o si no de manera pre-ensamblada, es decir se compra las tarjetas listas para ser usadas.



**Figura 12-1:** Placa Arduino

Fuente: (Fernández, 2014, p.12)

#### 1.4.2. Tabla comparativa de las Plataformas existentes

Una vez contando con toda la información necesaria de las plataformas de hardware libre, se procede a realizar una tabla comparativa para poder identificar la plataforma que más se adapte con las características del diseño que se va a llevar a cabo, en la *Tabla 3-1*, se puede apreciar dicha comparación, es prioritario tomar en cuenta la factibilidad de acceso rápido a las diversas plataformas existentes.

**Tabla 3-1:** Comparación entre Plataformas de Hardware Libre

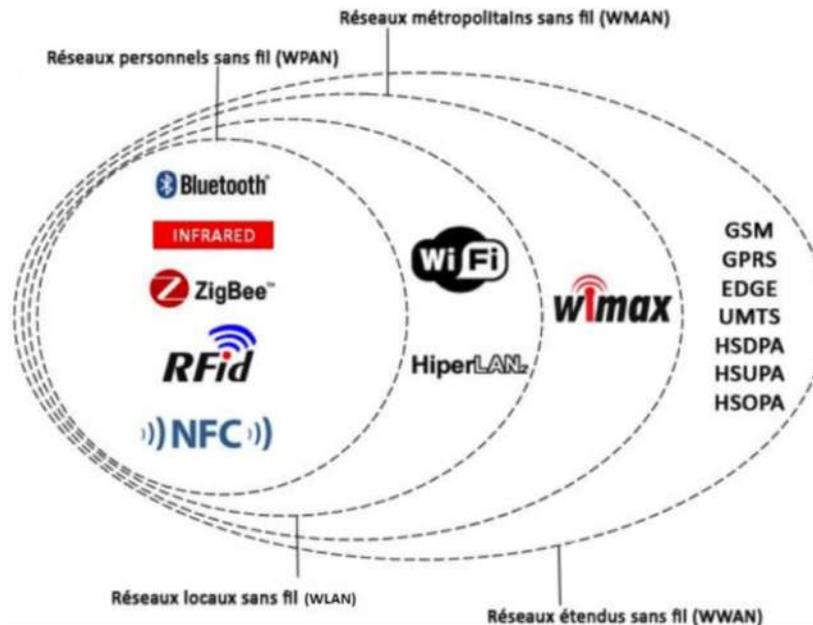
PLATAFORMA	VENTAJAS	DESVENTAJAS	MICRO	MEMORIA	VOLTAJE
<b>Arduino</b>	Disponibilidad de varios elementos en el mercado y bajo costo	Para grandes proyectos se requiere de características específicas.	ATMega 382	32-256 Kb	5 - 9V
<b>INTEL Galileo</b>	Es compatible con Arduino uno R3	Al ser compatible con este los usuarios prefieren Arduino además es de alto costo.	INTEL Quark SO Cx1000	512 Kb	7 - 15V
<b>Spark</b>	Sencillo de configurar y económico.	No existe diversidad en la marca.	ARM-32 Cortex	128 Kb	3.6V
<b>Waspote</b>	Disponibilidad de varios elementos en el mercado	Actúa bajo peticiones específicas, posee un alto costo.	ATMega 1281	128 Kb	3.3 - 4.2V
<b>Pingüino</b>	Basado en un código más abierto.	Las librerías en ocasiones no son bien portadas.	PIC 18F47J53	128 Kb	3.3 – 5V

Realizado por: Adriano, E., 2017

Fuente: (Cherrez, G., 2017, p.20)

### 1.5. Tecnologías para la transmisión inalámbrica de datos.

Las redes inalámbricas surgen como un complemento a las redes cableadas. “Son redes que utilizan ondas de radio para conectar los dispositivos, sin la necesidad de utilizar claves de ningún tipo.” (Salazar, J., 2016). En la *Figura 13-1* se puede observar la distribución de las redes inalámbricas según la distancia que recorre la señal, así tenemos en primer lugar las comunicaciones satelitales, seguidas de la telefonía celular como GSM, la evolución más reciente es LTE (Long Term Evolution), en tercer lugar están las tecnologías Wifi o Zigbee que son de menor alcance pero por su parte Wi-Fi permite una comunicación dentro de un aeropuerto, hotel, para el caso de Zigbee la comunicación entre sensores en cuarto lugar están las tecnologías que unen dispositivos en una red PAM (Área Personal) como es el caso de Bluetooth, por último se encuentran aquellas tecnologías que usan un antena transmisora (Tx) próxima como es el caso de NFC. En la actualidad GSM, GRPS, RF, Bluetooth y Wi-Fi son las tecnologías más usadas. (Fernández et al., 2009, p.28)



**Figura 13-1:** Distribución de Redes Inalámbricas.

Fuente: ([http://monge.univ-mlv.fr/~dr/XPOSE2010/NFC\\_sur\\_mobile/carac\\_rsf.html](http://monge.univ-mlv.fr/~dr/XPOSE2010/NFC_sur_mobile/carac_rsf.html))

*Radiofrecuencia (RF):* Ofrece mayor cobertura debido a que es aplicable para áreas de corto y mediano alcance, ofrecen un mayor ancho de banda además que no requieren una comunicación visual entre sus dispositivos, poseen un bajo consumo de energía, operan en las frecuencias de 433Mhz hasta los 2.4Ghz.

*Bluetooth:* permite la comunicación entre diversos dispositivos en un rango de 10 metros (para el caso de la Clase 1 alcanza 100 metros), opera en la frecuencia de 2.4Ghz, basados en una comunicación de radio de corto alcance por lo cual es de bajo costo. (Jürgens et al., 2008, p.23)

*Wifi:* es una red de área local basada en el estándar IEEE 802.11 ofrece acceso de banda ancha en múltiples entornos, su tasa de transmisión es de hasta 54Kbps para un canal de 20Mhz en la banda de 2.4Ghz, caracterizado por emplear las modulaciones PSK, QPSK y OFDM, relativamente presenta un bajo costo de instalación además de ser una red escalable. (Jürgens et al., 2008, p.20-21)

*WiMax:* diseñada para operar en áreas metropolitanas basada en el estándar IEEE 802.16, caracterizada por alcanzar hasta 50km en condiciones ideales con una tasa de transmisión de hasta 70Mbps, a diferencia de Wi-fi esta tecnología ofrece QoS, su ancho de banda es amplio dado que una misma estación base puede soportar más de 60 enlaces con conectividad T1/E1. (Jürgens et al., 2008, p.22)

*GSM*: permite la comunicación para telefonía móvil, su frecuencia de transmisión esta entre 890-915Mhz, posee un acceso múltiple TDMA/FDMA con Duplexado FDD, el ancho de banda por radiocanal es de 220Khz su velocidad de transmisión es de 270.8Kbps. (Ribadeneira, 2016, p.3)

*GPRS*: es la evolución de GSM, se basa en la conmutación de paquetes, ofrece un servicio de datos con una velocidad de hasta 76.8Kpbs empleando protección y de 115.2Kpbs sin protección, permite el manejo de protocolos IP, su costo es proporcional a la cantidad de información transmitida más no al tiempo de duración. (Ribadeneira, 2016, p.71)

## **1.6. Modelos animales para el estudio de la ansiedad.**

Los modelos animales han sido un instrumento fundamental en el estudio de la ansiedad, pues han permitido desenmarañar procesos fisiológicos, conductuales, bioquímicos entre otros, que se asemejan a las conductas del ser humano.

En el caso de la ansiedad el método que ha resultado ser más efectivo es la administración vía oral en roedores para luego ser sometidos a pruebas psicolepticas para evidenciar niveles de estrés, temor, y ansiedad.

Las pruebas en roedores se pueden clasificar en: condicionadas y no condicionadas.

### **1.6.1. Pruebas condicionadas**

Están basadas en la capacidad de responder a un estímulo con la respuesta específica a otro estímulo. Estos modelos permiten un control altamente preciso de los niveles de conductas basales por parte del investigador, las limitaciones de este modelos es que las especies que van a ser parte de la experimentación deben tener un entrenamiento previo, además muchas de las versiones de este tipo de pruebas son condicionadas según el grupo que las usa por ende en cada una de ellas hay modificaciones de los parámetros de la prueba.

### **1.6.2. Pruebas no condicionadas**

Se basan en la medición de respuestas conductuales o fisiológicas ante estímulos nuevos o estresantes, el sujeto de prueba es sometido a un espacio abierto, alto o luces brillantes. Una de

las ventajas que más se destaca es que poseen cierta validez ecológica al ser pruebas rápidas y relativamente sencillas. (Polanco, Vargas-Irwin & Góngora, 2011).

#### *1.6.2.1. Laberinto en Cruz Elevado (EPM)*

Es uno de los modelos más utilizados en la evaluación ansiolítica debido a que se emplean estímulos naturales, miedo a espacios abiertos así como el miedo a caminar sobre espacios elevados y relativamente estrechos.

Este aparato está constituido por dos brazos abiertos, dos brazos cerrados y un área central elevados a una altura determinada sobre el suelo, el roedor es ubicado en el centro con dirección a uno de los brazos abiertos, se le permite la exploración por un determinado tiempo por lo general dura cinco minutos. En este tipo de prueba se cuantifica el número de entradas en los respectivos brazos, del mismo modo el tiempo que dedica a los brazos. En la *Figura 14-1*, se puede observar este tipo de equipos.



**Figura 14-1:** Laberinto en Cruz Elevado.

Fuente: [http://labequim.com.mx/html/STOELTING/html/Laberinto\\_en\\_cruz\\_elevado.htm](http://labequim.com.mx/html/STOELTING/html/Laberinto_en_cruz_elevado.htm)

#### *1.6.2.2. Caja Luz/Oscuridad (MCL)*

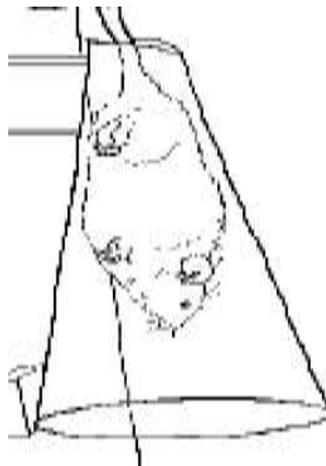
Este método se desarrolla en una caja de prueba oscurecida un tercio de ella con pintura negra, en el interior de la caja existe una abertura que separa la zona clara iluminada con una lámpara fluorescente de la zona oscurecida. El sujeto es colocado en la zona clara con dirección a la parte oscura, la duración de esta prueba es de cinco minutos en los cuales se cuantifica su comportamiento. En la *Figura 15-1* se puede apreciar este tipo de equipos.



**Figura 15-1:** Modelo claro/oscuo.  
**Fuente:** Idrobo, T., 2016, p. 38

### 1.6.2.3. Test de la Chimenea.

En este tipo de prueba se evalúa la capacidad del roedor de subir marcha atrás por un tubo de vidrio, el mismo que posee las dimensiones de diámetro apropiado según sea el peso del roedor y que está marcado a una distancia de 20 centímetros de uno de los extremos. El roedor en condiciones normales sobrepasa la marca en un tiempo inferior a los 30 segundos si no es el caso no consigue sobrepasar dicha marca. En la *Figura 16-1* se puede observar el test de la chimenea.



**Figura 16-1:** Test de la Chimenea.  
**Fuente:** Idrobo, T., 2016, p. 38

#### 1.6.2.4. Prueba de Campo Abierto

Esta prueba ha sido una de más utilizadas para medir ansiedad, básicamente consta de un recinto cuadrado, rectangular o circular de donde el roedor no puede escapar, el tiempo de duración de esta prueba varía según el laboratorio donde se experimente puede durar desde dos a treinta minutos, donde se evalúa la exploración del roedor a un ambiente “novedoso”. (Mora & Salas, 2014, p.5). En la *Figura 17-1* se aprecia la caja de la prueba de campo abierto.



**Figura 17-1:** Modelo de Campo Abierto.

Fuente: <http://dyrefaciliteter.au.dk/institut-for-biomedicin/phenotyping/behavioural-phenotyping/>

En la Tabla 4-1 se puede visualizar el compendio de pruebas condicionadas y no condicionadas más utilizadas para evaluar la ansiedad.

**Tabla 4-1:** Pruebas más usadas para medir ansiedad.

Modelos de respuesta no condicionada	Modelos de respuesta condicionada
Campo abierto	Respuesta emocional condicionada
Tabla de agujeros	Supresión condicionada
Caja claro/oscuro	Evitación pasiva/activa
Laberinto en cruz elevado	Prueba de cuatro platos



Interacción social	Test de conflicto de Geller-Seifter y el de Vogel
Supresión de la ingesta sumergida por la novedad	Condicionamiento de vocalizaciones ultrasónicas
Contraste negativo	Estimulación eléctrica cerebral
Conducta defensiva de enterramiento	Modelo de Thiébot de retira de la señal de seguridad
Holeboard	Efecto del reforzamiento parcial sobre la extinción
	Potenciación de la respuesta de sobresalto

**Realizado por:** Adriano, E., 2018

**Fuente:** (Polanco, Vargas-Irwin & Góngora, 2011, p.146).

## CAPÍTULO II

### 2. MARCO METODOLÓGICO

#### 2.1. Metodología de la investigación

Para el desarrollo del presente trabajo de titulación fue preciso establecer los requerimientos idóneos para llevarlo a cabo. Se realizó un estudio mediante la aplicación de métodos y técnicas de investigación que a continuación son detallados.

##### 2.1.1. Tipo de investigación

La presente investigación se establece como una propuesta tecnológica, se necesitó de un estudio mediante la aplicación de métodos y técnicas de investigación tipo básica/aplicativa. Se eligió este tipo de investigación debido a que se centra su idea en aplicar y utilizar los conocimientos adquiridos durante la investigación del problema establecido, siguiendo de ese modo una investigación básica. Con el uso de la investigación aplicada se buscó mejorar el rendimiento de las pruebas de la actividad ansiolítica en ratas y así poder determinar el *hardware* y *software* necesario para la implementación de los sistemas de monitoreo.

##### 2.1.2. Método de investigación

Se decidió emplear la combinación de métodos teóricos empíricos, los cuales son:

###### 2.1.2.1. Métodos teóricos

**Revisión de documentación:** Con la finalidad de obtener información que se relacione con los modelos animales para evaluar la actividad ansiolítica y dispositivos que permitan mejorar el rendimiento de este tipo de pruebas, información sobre los avances tecnológicos existentes en el

campo de la electrónica y en particular en el área de sensores, módulos de transmisión y recepción de datos.

**Sistematización:** Para poder aplicar la información recolectada al diseño de los sistemas de monitoreo para evaluar la actividad ansiolítica en ratas.

#### 2.1.2.2. *Métodos empíricos*

**Experimentación:** Necesario para poder evaluar cada uno de los bloques que integran el sistema.

**Observación:** Servirá para validar los sistemas diseñados e implementados.

Al analizar cada método se sigue una serie de pasos los mismos son:

- a. Estudio y análisis de los modelos animales para la evaluación ansiolítica y su impacto en la farmacología.
- b. Descripción de los distintos tipos de transmisión de la información.
- c. Selección del tipo de sistema de monitoreo que más se adapte para evaluar la actividad ansiolítica.
- d. Diseño e implementación del *hardware* y *software*.
- e. Realización de pruebas y comprobación de los sistemas.
- f. Validación de resultados, conclusiones y recomendaciones.

#### 2.1.3. *Técnicas de investigación*

Con la finalidad de compilar la información que será necesaria para la implementación de los sistemas de monitoreo para evaluar la actividad ansiolítica en ratas, se usará las siguientes técnicas de investigación:

**Documental:** Recopilación de la información como artículos científicos, libros, investigaciones, los cuales son necesarios para poder definir parámetros, elegir dispositivos que se deben adaptar a los sistemas para su implementación, los mismos que deben ser adecuados para no interferir con el estado de las ratas.

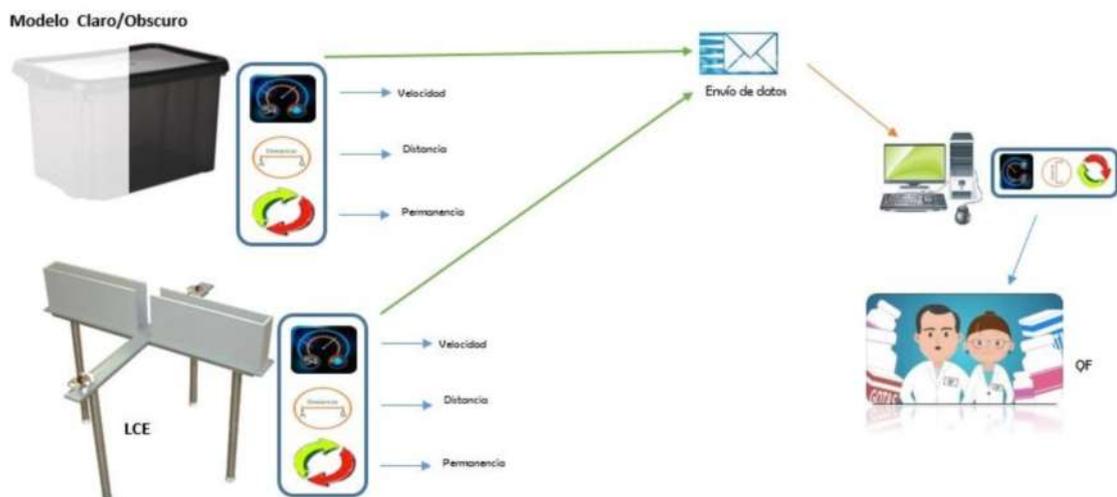
**De campo:** Una vez seleccionados los elementos que forman parte de cada sistema, es necesario verificar que el monitoreo de cada uno, validando que los tiempos de permanencia, el número de cruces sea el correcto, distancia recorrida, velocidad media sean los correctos.

## 2.2. Diseño del sistema de monitoreo de la actividad ansiolítica

### 2.2.1. Esquema general del sistema

En este apartado se desarrollará el diseño de los sistemas de monitoreo para medir la actividad ansiolítica en ratas. En la *Figura 1-2* se observa el esquema general de los sistemas, los cuales tienen varios entornos descritos a continuación.

- Entorno de los equipos (de donde se obtendrá los datos)
- Medio de transmisión (Tecnología Inalámbrica a usar).
- Entorno de Visualización (tecnología portátil).



**Figura 1-2:** Esquema General del Sistema

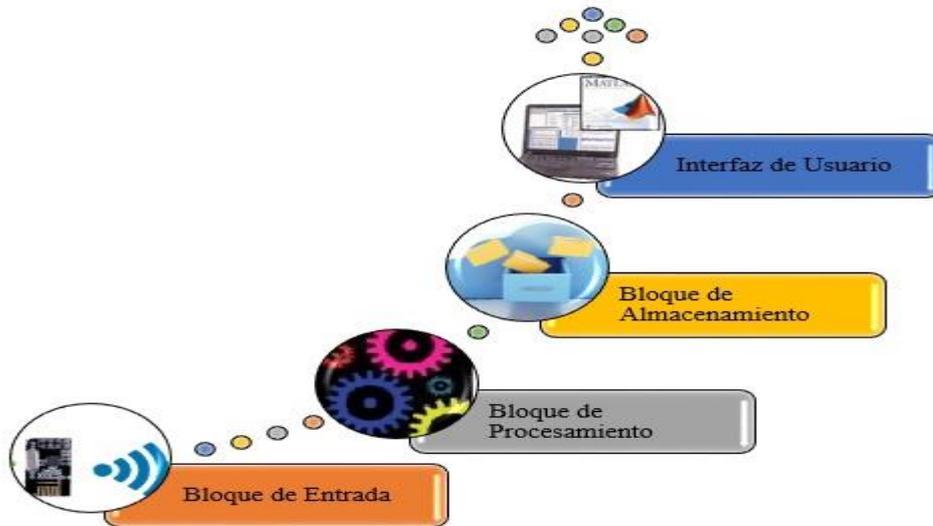
Realizado por: Adriano Erica, 2018.

### 2.2.2. Diseño del sistema

En el desarrollo de los sistemas de monitoreo para medir la actividad ansiolítica in vivo en ratas se adapta tecnologías y sensores que permiten medir dicha actividad para precisar las variables útiles que precisen la actividad ansiolítica dependiendo de factores externos.

Los datos obtenidos se transmitirán a través de un módulo Transceptor NRF24L01 el mismo que se interconectará a su respectivo receptor, el cual deberá estar conectado de manera serial a la Pc. En la *Figura 2-2* se visualiza el diseño para que el sistema este operativo.

Cada bloque se estudiará por separado para una mayor comprensión, pero a su vez cada uno de estos debe ser compatible para un correcto funcionamiento.



**Figura 2-2:** Bloques de Proceso del Sistema  
**Realizado por:** Adriano Erica, 2018.

En la *Tabla 1-2* se realizará la descripción de cada uno de los bloques de proceso, descritos en la *Figura 2-2* para una mayor comprensión.

**Tabla 1-2:** Bloques de Proceso del Sistema de Monitoreo.

BLOQUE	DETALLE
<b>ENTRADA</b>	Es el bloque encargado de recopilar todos los datos proporcionados por los equipos de monitoreo.
<b>PROCESAMIENTO</b>	Bloque cuya función es manipular los datos entregados por el bloque de entrada por medio de una plataforma libre con su respectiva programación.
<b>ALMACENAMIENTO</b>	Los datos recogidos anteriormente se colocarán en la aplicación desarrollada para su visualización.
<b>INTERFAZ DE USUARIO</b>	Una vez recogidos y manipulados los datos de los bloques anteriores, estos podrán ser visualizados en tiempo real para tener conocimiento de lo que ocurre con los sistemas de monitoreo por medio interfaz amigable

**Realizado por:** Adriano Erica, 2018.  
**Fuente:** Cherez G, 2017.

### **2.3. Requerimiento para el diseño de los sistemas de monitoreo.**

Una vez determinados todos los bloques para el diseño de los sistemas de monitoreo de la actividad ansiolítica se buscará todos los elementos necesarios para satisfacer las necesidades de dichos procesos. A continuación, se especificará todos los elementos en cada uno de los bloques descritos anteriormente.

#### **2.3.1. Bloque de entrada**

Esta es la parte inicial donde se detallará todos los dispositivos benéficos para recolectar la información necesaria y captar todos los eventos para complementar el bloque de entrada.

Para la recolección de la información que se realizará con sensores, se han manifestado el evento producido en el primer bloque de los sistemas de monitoreo. Dicho evento se describe a continuación:

- Medición de distancias (Recolección de datos para el funcionamiento de los sistemas de monitoreo),

Este evento es muy importante debido a la situación de la rata, este evento antes mencionado apoyará la recolección de los datos para la medición de la distancia total recorrida, la velocidad media, tiempos de permanencia y números de cruces a los diferentes estados por parte de la rata. El dispositivo utilizado en dicho evento se detalla a continuación.

##### **2.3.1.1. Sensores**

Para la valoración del evento antes citado se requiere de dispositivos específicos que comprendan las variables a medir, los sensores seleccionados para la medición de este evento son:

- Proximidad

### 2.3.1.2. Sensor proximidad

Los sensores de proximidad detectan la cercanía del objeto y/o miden que tan lejos está. Estos sensores “trabajan en los fundamentos de triangulación de luz reflejada, tiempo lapso del sonido reflejado, o en las corrientes de Farcault inducidas.” (Jeffus, L, 2011, p 908)

Dada la naturaleza de este tipo de sensores, la luz permite el soporte de envío de la información que ella misma emite, debido a las modificaciones de las radiaciones que afectan las variables físicas estas pueden ser detectadas usando un sensor adecuado. Este tipo de sensores se sugirió para la medición de desplazamientos de las ratas, los valores obtenidos serán enviados a través de la tecnología transceiver NRF24L01 para ser procesados en Matlab, cuyos datos procesados serán visualizados en una aplicación.

En la *Tabla 2-2* se describirá los tipos de sensores de proximidad existentes en el mercado.

**Tabla 2-2:** Cuadro comparativo sensores de proximidad

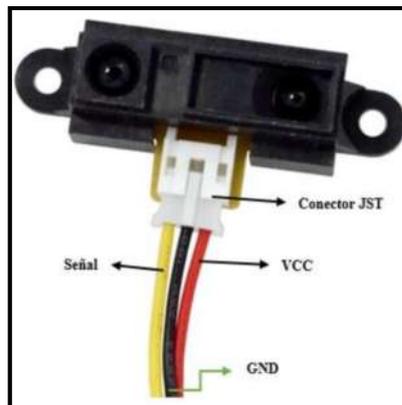
MODELO	Sharp GP2Y0A21YK0F	Sharp GP2Y0A60SZLF	Sensor Infrarrojo IR FC-51	Sensor Ultrasónico
<b>Rango de medida.</b>	10-80cm	10-150cm	2-30cm (ajustables con potenciómetro)	2-400cm
<b>Dimensiones</b>	29.5x13x13.5mm	33x10.4x10.2mm	4.5x1.4x0.7cm	45x20x15 mm
<b>Alimentación</b>	4.5-5.5V	2.7-5.5V	3.3-6V	5V
<b>Corriente de salida max</b>	30mA	33mA	23-43mA	15mA
<b>Ángulo de Cobertura</b>	40°	>40°	35°	15°
<b>Tipo de Salida</b>	Analógica	Analógica	Digital	Analógica

Realizado por: Adriano Erica, 2018.

Con la información de la *Tabla 2-2* se definió los sensores más apropiados para la realización de los sistemas:

### 2.3.1.3. Sensor Infrarrojo Sharp GP2Y0A21YK0F

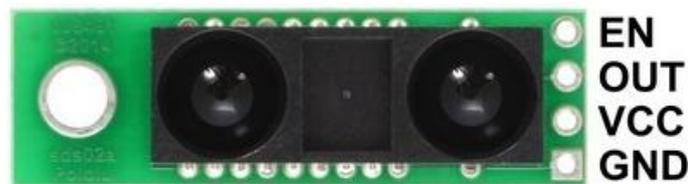
Este tipo de sensor está compuesto por un Led Infrarrojo junto con un dispositivo detector de posición (PSD) y un procesador integrado cuya función es calcular la distancia (Llamas, 2016), al ser del tipo IR no produce efectos negativos en las ratas, la resolución que faculta es de 1 centímetro, con un tiempo de muestreo de  $38\pm 10$ ms, posee un conector de 3 pines JST, son de fácil manipulación por la que facilita su implementación, basado en el método de triangulación para detectar la distancia del objeto. En la *Figura 3-2* se puede observar los pines de este sensor. Este sensor será usado en el modelo Claro/Oscuro.



**Figura 3-2:** Sensor Sharp 10-80cm  
Realizado por: Adriano Erica, 2018.

### 2.3.1.4. Sensor Infrarrojo Sharp GP2Y0A60SZLF

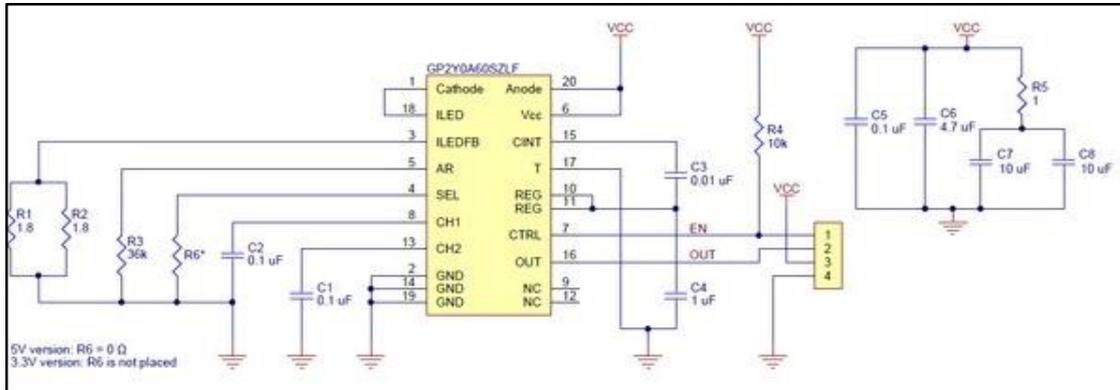
Este al igual que el anterior son del tipo IR, en particular este sensor es pequeño y asequible, posee una tasa de actualización de 60 Hz, su amplio rango de detección y su resistencia a la interferencia IR ambiental lo diferencian de los otros modelos IR de distancia. El tiempo de muestro es de  $16\pm 4$ ms, al igual que su antecesor se basa en el método de triangulación para obtener medidas de distancias, este tipo de sensor será usado en el laberinto en cruz elevado. En la *Figura 4-2* se puede visualizar los pines de conexión.



**Figura 4-2:** Sensor Sharp 10-150cm  
Realizado por: Adriano Erica, 2018.



Como se indicó este tipo de sensor ofrece resistencia al IR ambiental, en la *Figura 5-2* se puede observar el diagrama de conexión.

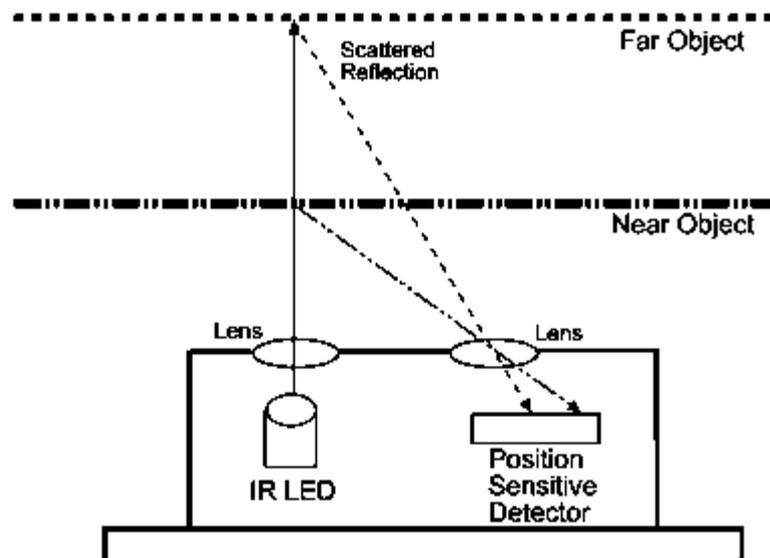


**Figura 5-2:** Esquema de la resistencia IR ambiental

**Fuente:** (<https://teslabem.com/index.php/catalog/product/view/id/316/s/sensor-sharp-de-10-150-cm-5v-gp2y0a60sz/>)

### 2.3.1.5. Método de triangulación sensores Sharp

Este método consiste en medir uno de los ángulos que forma el triángulo entre el: emisor-objeto-receptor, el receptor es un PSD cuya función es detectar el punto de incidencia que depende del ángulo y de la distancia del objeto. En la *Figura 6-2* se puede apreciar la operatividad del método de triangulación, en base a esto podemos decir que a mayor cercanía del objeto más amplio es su ángulo, en cambio mientras más alejado se encuentra el objeto más estrecho es el ángulo de visión del sensor Sharp. (*NaylampMechatronics, 2016.*)



**Figura 6-2:** Método de Triangulación

**Fuente:** Llamas, L. 2016.

La geometría del sensor sus ópticas son limitantes para el rango del sensor. Estos sensores poseen un ángulo de visión muy reducido, esto se debe a que la luz que emite es puntual. Se puede usar varios sensores Sharp para ampliar el rango de visión, los mismos que pueden estar en direcciones opuestas o en la misma dirección tomando en cuenta que las líneas de visión no queden muy juntas.

### **2.3.2. *Bloque de procesamiento***

Este es el bloque donde se encuentran todos los datos recolectados en el bloque anterior para procesarla, filtrarla y enviarla a la aplicación para su visualización, además en esta etapa se usa hardware libre para procesar la información recolectada.

Una vez que se ha obtenido la información el hardware libre hace uso de algoritmos particulares para que los sensores mencionados anteriormente puedan ser controlados.

Para tomar la decisión correcta respecto a un hardware libre en particular se analizó los requerimientos de los sistemas de monitoreo, tomando en cuenta por otro lado la compatibilidad con los sensores ya mencionados además de su fácil acceso en el mercado.

#### **2.3.2.1. *Hardware libre***

Al hablar de hardware libre se puede decir que son todos aquellos dispositivos que poseen código abierto, es decir son de acceso público incluso para que sus versiones sean mejoradas, estos se consiguen de manera gratuita.

Las plataformas que están dentro de esta área son:

- Arduino
- Intel Galileo
- Spark
- Waspote
- Pingüino

Entre todas las plataformas mencionadas, se eligió Arduino dado que es la más idónea para adaptarse a las necesidades de los sistemas de monitoreo, posee alta disponibilidad en el mercado y cumple con los requerimientos necesarios para su programación.

Las demás plataformas mencionadas poseen una poca disponibilidad en el mercado, lo cual es un factor importante para no tomarlas en cuenta, además de sus costos altos para poder acceder a ellas.

### 2.3.2.2. Plataforma Arduino

Arduino es una plataforma open source, es decir de código abierto, basada tanto en software como hardware flexibles y de fácil uso, su principal objetivo es brindar soporte en la elaboración de prototipos y a su vez en diseños electrónicos con una programación acorde a cada necesidad.

Su hardware se basa en un Microcontrolador que cuenta tanto con entradas analógicas como digitales para ejecutar las órdenes que el usuario desee, su entorno de desarrollo está basado en el lenguaje de programación C++, cuyas instrucciones son de fáciles de usar y aprender.

Desde su aparición en el año 2005, Arduino no ha perdido importancia, actualmente es uno de los dispositivos más aclamados para la elaboración de proyectos electrónicos tanto por estudiantes, profesionales e incluso empresas tecnológicas que buscan satisfacer las necesidades de sus clientes quienes pueden ensamblar sus placas acorde a su gusto y uso de esta plataforma.

Los principales modelos existentes en el mercado se pueden observar en la *Tabla 3-2*, donde se muestran sus principales características así como sus desventajas.

**Tabla 3-2:** Modelos existentes de Arduino.

MODELO	YUN	NANO	101	UNO	MEGA 2560
<b>Imagen</b>					
<b>Dimensiones</b>	73 x 53 mm	45 x 18 mm	68.6 x 53.4 mm	45 x 18 mm	101.52 x 53.3 mm
<b>Memoria</b>	32 Kb	16.32 Kb	196 Kb	32 Kb	256Kb
<b>Microcontrolador</b>	ATmega32U4	ATmega168 o ATmega328	Intel Curie	ATmega328	ATmega2560
<b>Puerto Serial</b>	1	1	1	1	4
<b>Pines Analógicos</b>	12	8	6	6	16
<b>Pines Digitales</b>	20	14	14	14	54
<b>Pines PWM</b>	7	0	4	6	15
<b>Alimentación</b>	5VDC	7 - 12VDC	7 - 12VDC	7 - 12VDC	7 - 12VDC
<b>Observación</b>	Microprocesador Linux, Ethernet, Wifi, USB	Conector mini USB, botón Reset.	Conector USB, power Jack ICSP, Botón de Reset.	Conector USB, Power Jack ICSP, Botón de Reset.	Conector USB, Power Jack ICSP, Botón de Reset.

**Realizado por:** Adriano Erica, 2018.

**Fuente:** (Cherrez G, 2017)

A partir de la *Tabla 3-2* que describe los modelos de Arduino existentes en el mercado, se tomó la decisión de usar 2 modelos para el diseño de los sistemas de monitoreo.

Se decidió usar 4 arduinos para llevar a cabo la elaboración de los sistemas de monitoreo, el primero y el tercero para la medición de distancias dadas por los sensores de proximidad, estos además serán los transmisores (Tx), el segundo y el cuarto serán los receptores (Rx), tanto en los transmisores como en los receptores estarán los módulos NRF24L01.

Con lo mencionado anteriormente se ha definido los modelos de Arduino a utilizar que son:

### 2.3.2.3. *Arduino Uno*

Para la recolección de los datos de las distancias emitidas por los sensores del modelo claro/oscurito y para los receptores se escogió la placa Arduino Uno como se muestra en la *Figura 7-2*, ya que cumple con las características necesarias para adaptarse al sistema, además de contar con la memoria suficiente para el proceso.



**Figura 7-2:** Arduino Uno  
Realizado por: Adriano Erica, 2018.

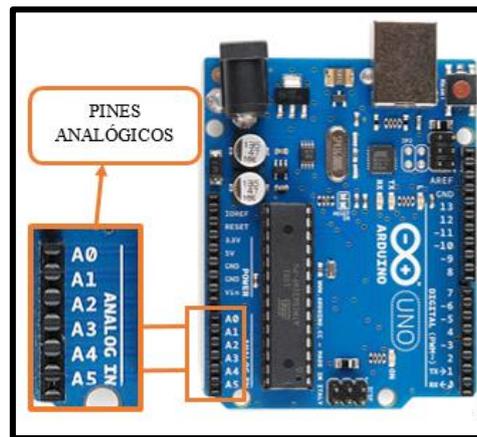
### 2.3.2.4. *Distribución de pines Arduino Uno*

Arduino uno está distribuido en cinco grupos de pines, los cuales son:

- Pines Analógicos.
- Pines Digitales

- Pines de Comunicación
  - Pines PWM
  - Pines Power
- 
- Pines Analógicos

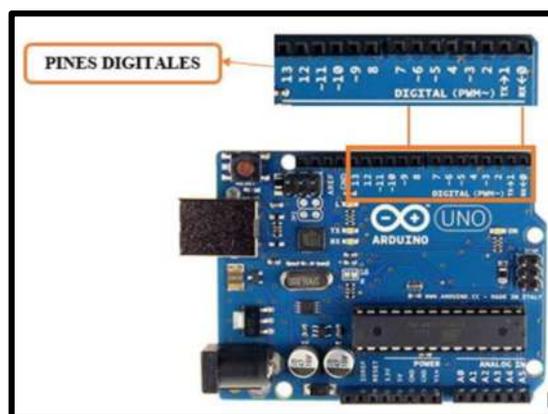
Los pines analógicos van desde A0-A5, su ubicación se puede observar en la *Figura 8-2*.



**Figura 8-2:** Distribución Pines Analógicos (Arduino Uno)  
 Realizado por: Adriano Erica, 2018.

- Pines Digitales

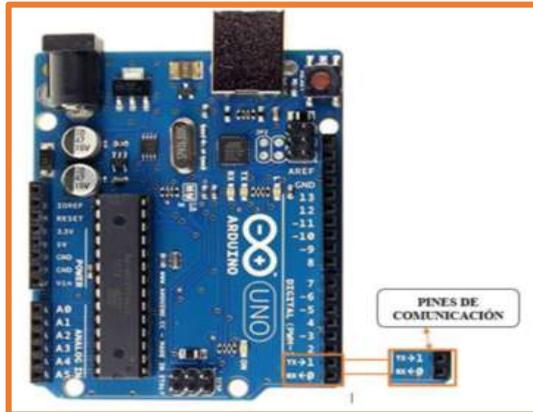
Denotados desde D0-D13 funcionan a 5V, su ubicación se la puede observar en la *Figura 9-2*.



**Figura 9-2:** Pines Digitales (Arduino Uno)  
 Realizado por: Adriano Erica, 2018.

- Pines de Comunicación

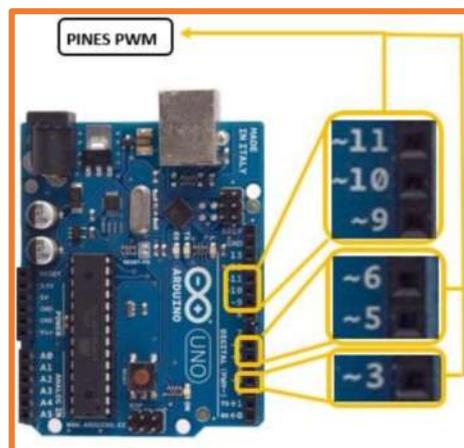
Estos pines sirven para la transmisión y recepción de información de cualquier elemento que esté conectado a estos se los puede observar en la *Figura 10-2*.



**Figura 10-2:** Pines de Comunicación (Arduino UNO)  
Realizado por: Adriano Erica, 2018.

- Pines PWM

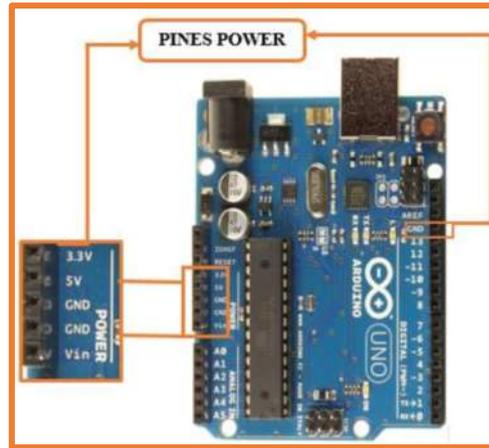
De los 14 pines digitales 6 de estos tienen salida PWM DE 8 bits, se los puede identificar porque llevan una virgulilla (~), así como se muestra en la *Figura 11-2*.



**Figura 11-2:** Pines PWM (Arduino Uno)  
Realizado por: Adriano Erica, 2018.

- Pines Power

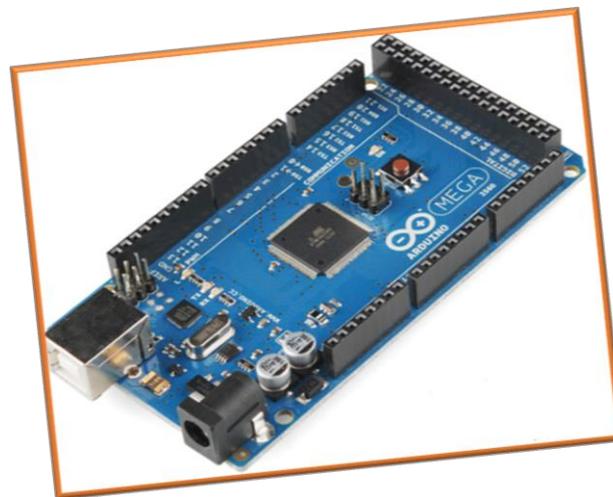
Estos pines son los encargados de brindar la alimentación a la placa Arduino, pueden ser alimentados con 3 o 5 voltios, tal como se muestra en la *Figura 12-2*.



**Figura 12-2:** Pines Power (Arduino Uno)  
Realizado por: Adriano Erica, 2018.

### 2.3.2.5. Arduino Mega 2560

Para la recolección de los valores de distancias medidas por los sensores del Laberinto en Cruz Elevado (EPM), se decidió utilizar este modelo de Arduino como se muestra en la *Figura 13-2*, ya que cuenta con la memoria necesaria para el cumplimiento de este proceso y se adapta fácilmente a las características del tipo de sensor a usar en el EPM.



**Figura 13-2:** Arduino Mega  
Realizado por: Adriano Erica, 2018.

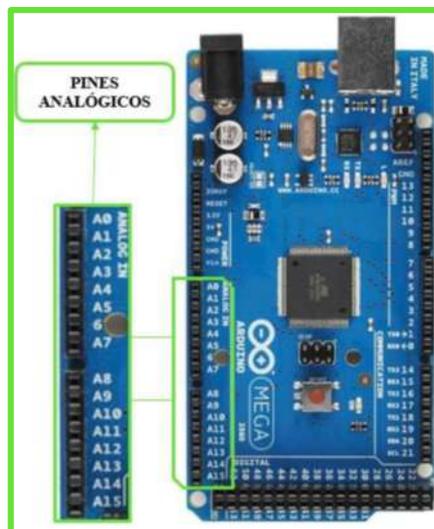
### 2.3.2.6. Distribución de los Pines Arduino Mega 2560.

Al igual que Arduino Uno este modelo también cuenta con cinco grupos para definir sus pines, los cuales son:

- Pines Analógicos
- Pines Digitales
- Pines de Comunicación
- Pines PWM
- Pines Power

- Pines Analógicos

Este modelo cuenta con 16 entradas analógicas identificadas desde A0 – A15, tal como se muestra en la *Figura 14-2*.

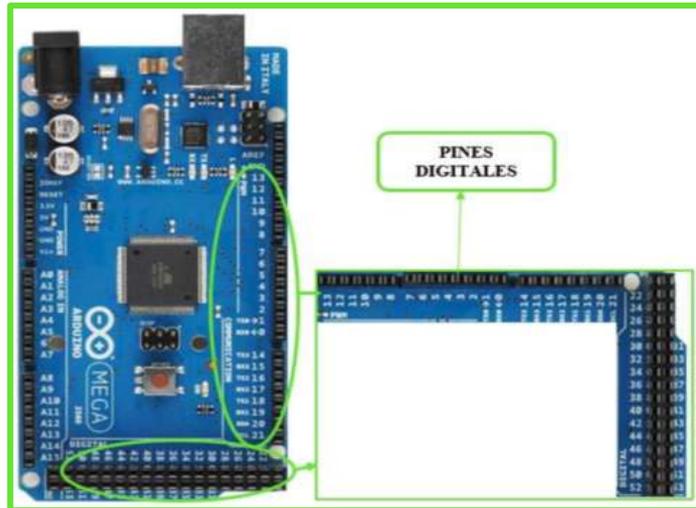


**Figura 14-2:** Pines Analógicos (Arduino Mega)  
Realizado por: Adriano Erica, 2018.

- Pines Digitales

Arduino Mega cuenta con 54 pines digitales, a diferencia de Arduino Uno, este modelo presenta una subdivisión de estos pines, es decir muestra identificados la funcionalidad de cada grupo de pines, tal como se muestra en la *Figura 15-2*.

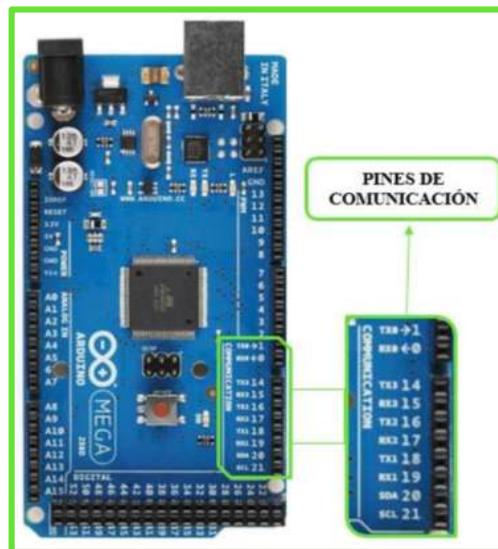




**Figura 15-2:** Pines Digitales (Arduino Mega)  
 Realizado por: Adriano Erica, 2018.

- Pines de Comunicación

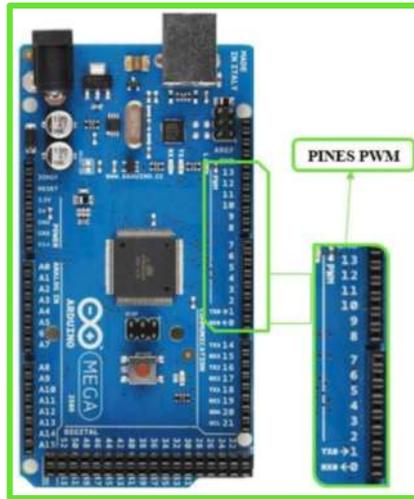
Estos pines son usados para la transmisión y recepción de información según sea las necesidades, cuenta con 4 pares claramente identificados, es decir va desde Tx0-Rx0 al Tx3-Rx3, tal como se puede observar en la *Figura 16-2*.



**Figura 16-2:** Pines de Comunicación (Arduino Mega)  
 Realizado por: Adriano Erica, 2018.

- Pines PWM

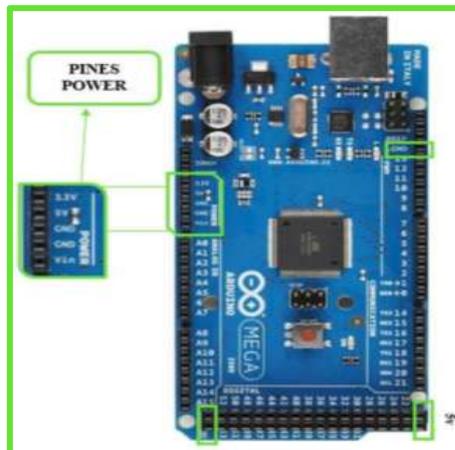
De los 54 pines digitales, 15 son destinados para PWM tal como se muestra en la *Figura 17-2*.



**Figura 17-2:** Pines PWM (Arduino Mega)  
 Realizado por: Adriano Erica, 2018.

- Pines Power

Estos pines son importantes para el correcto funcionamiento de la placa, su alimentación va desde 3 a 5 voltios, en la *Figura 18-2* se muestran estos pines.



**Figura 18-2:** Pines Power (Arduino Mega)  
 Realizado por: Adriano Erica, 2018.

Como se describió en la tabla anterior tanto el Arduino Uno como el Arduino Mega son los más idóneos para la realización del trabajo de titulación, para la alimentación de los arduinos se utilizará conexiones externas en un rango de voltaje de 5-12 Voltios que es lo que soportan estas placas.

### 2.3.2.7. Tecnología inalámbrica

Para seleccionar la tecnología de comunicación que mejor se adapte al sistema y a su vez al hardware libre se realizó una tabla comparativa destacando sus características principales la misma que se observa en la *Tabla 4-2*.

**Tabla 4-2:** Comparación de módulos inalámbricos

MÓDULO	Bluetooth HC-05	XBee XB24- AWI-001	Transceiver RF NRF24L01	Wifi ESP8266- ESP-01	Tarjetas RFID NFC MF522-ED
Imagen					
Dimensiones	1.7x 4 cm	2.4 x 2.7 cm	15x44mm	11.5 x 11.5 mm	40x60 mm
Frecuencia	2.4Ghz, Banda ISM	2.4Ghz, Banda ISM	2.4Ghz – 2.4835Ghz Banda ISM	2.4Ghz 802.11 b/g/ n	13.56Mhz
Potencia de emisión	≤ 4 dBm, Clase 2	1 mW (+0 dBm)	+7dB	802.11b mode + 19.5dBm	
Alcance	5- 10 m	90 m en línea de vista, 30 m en interiores	15 a 30 m en interiores, 70 a 100 m en línea de vista	30 metros teórico	0-60mm
Sensibilidad	≤ -84 dBm a 0.1% BER	-92 dBm	≤: -90dB	-92 dBm	≤ -84 dBm
Velocidad	Asincrónica: 2.1 Mbps (max.)/160 kbps, sincrónica: 1 Mbps/1 Mbps	250 kbps	250kbps, 1Mbps and 2Mbps on-air data-rate	9600 o 115200 baudios	106Kbaud
Seguridad	Autenticación y encriptación (Password por defecto: 1234)	Encriptación AES de 128 bits	Encriptación AES	WPA / WPA2	No determinada
Consumo de corriente	50 mA	RX 50 mA, TX 45 mA	13mA	10uA	13-26mA a 3.3V
Voltaje de alimentación	3.6 V a 6 V	2.8 V a 3.4 V	1.9 V a 3.6 V	3.3V	3.3V

<b>Temperatura de operación</b>	-20 °C a +75 °C	-40 °C a +85 °C	-40 °C a +85 °C	-40 ~ 125 °C	-20 a 80°C
<b>Observación</b>	El pin RX del requiere resistencia de pull-up a 3.3 V (4.7 k a 10 k). Programación media-baja.	Las señales deben ser de máximo 3.3 V, voltajes superiores en sus entradas pueden dañar el módulo. Programación media-baja	Permite la conexión de hasta seis dispositivos simultáneamente. Fácil programación.	Integrated low power 32-bit MCU SDIO 2.0, SPI, UART, I2C. Programación Alta	SPI, Serial UART, I2C. Fácil de programación.

Realizado por: Adriano Erica, 2018.

De la información recolectada en la *Tabla 4-2*, se determina que la tecnología a usar es el transceiver NRF24L01, por ser asequible en el mercado, además de poseer una fácil programación, tamaño reducido, bajo consumo de energía, en términos económicos es relativamente bajo en comparación a XBEE, del mismo modo se descartan tecnologías con bluetooth y NFC por las limitaciones en cuanto a el rendimiento de su transmisión.

### 2.3.3. *Bloque de almacenamiento*

Este bloque es muy importante para el funcionamiento de los sistemas de monitoreo donde se dispone del tiempo de duración de las pruebas de actividad ansiolítica ya que de esto va a depender el número de muestras que los sensores deberán recolectar. Los datos obtenidos se guardarán dentro del proceso de la aplicación para después de ser procesados brindar la información referente sobre la actividad ansiolítica.

La idea de guardar directamente los datos es que se puede procesar la información con lo que está ocurriendo en ese momento para poder visualizarla después de haber pasado por todas las etapas de procesamiento para garantizar la obtención de resultados confiables, siendo estos visualizados después que la aplicación del usuario haya finalizado el procesamiento de datos.

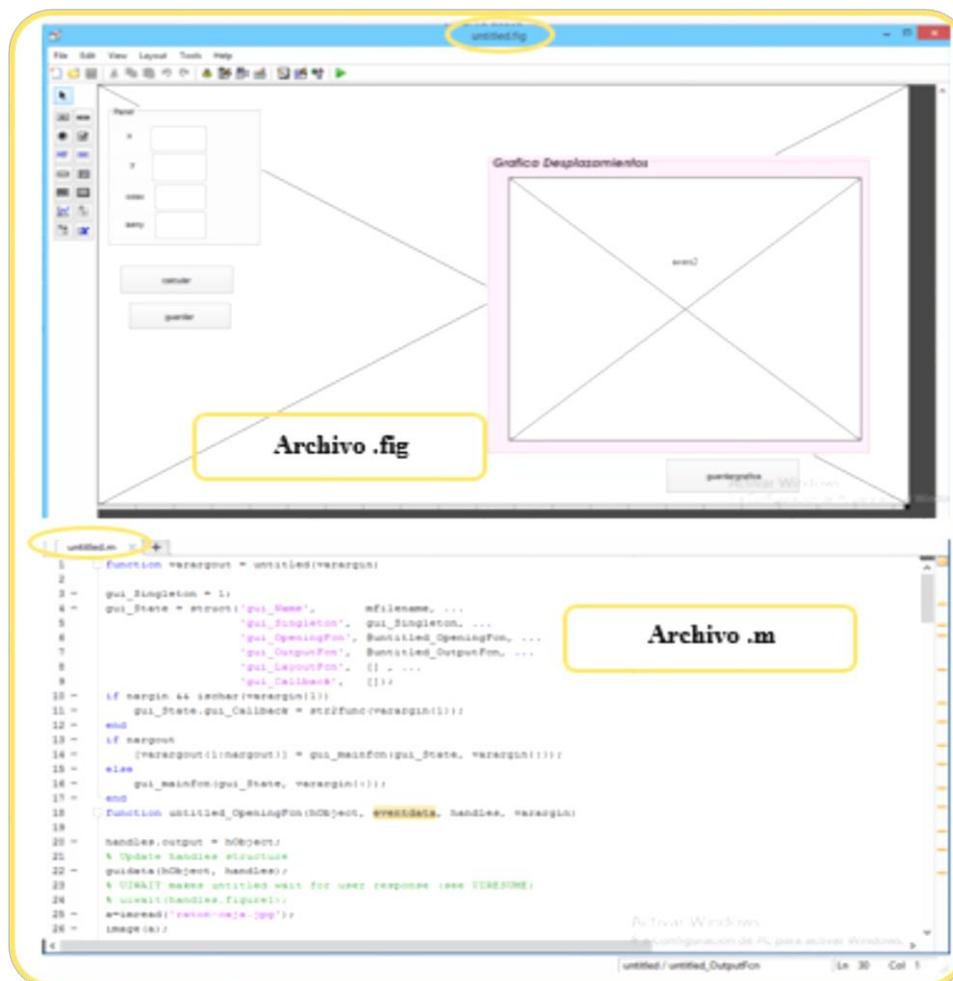
### 2.3.4. *Interfaz de usuario*

El bloque final muestra los datos que fueron obtenidos por los sensores, los mismos que fueron procesados por una plataforma especializada para su visualización en una aplicación

computacional con una interfaz amigable para el usuario. El entorno para la realización de la aplicación es Matlab haciendo uso de la interfaz gráfica (GUIDE), este software es creado por MathWorks compatible con Unix, Windows, Mac OS X y GNU/Linux.

La característica más importante de Matlab es su capacidad de crecimiento, basándose su escritura del programa en lenguaje matemático permitiendo adaptar y ampliar el lenguaje mediante ficheros script y funciones .m, Matlab cuenta con posibilidad de adquirir una versión beta de fácil acceso, para que se pueda comunicar con Arduino se debe contar con paquetes de soporte de Arduino para Matlab, los mismo que se pueden descargar directamente en la página de MathWorks.

La interfaz gráfica de Matlab (GUIDE) permite al usuario trabajar en dos partes en conjunto, la primera es el archivo .fig, que es en donde se creará la aplicación permitiendo agregar botones, axes, o demás según sea el criterio del diseñador para permitir una interfaz amigable al usuario, la segunda es el .m que es donde se procesará la información emitida por la plataforma libre, es de fácil entendimiento ya que Matlab cuenta con un lenguaje de programación propio. En la *Figura 19-2*, se puede observar ambos archivos.

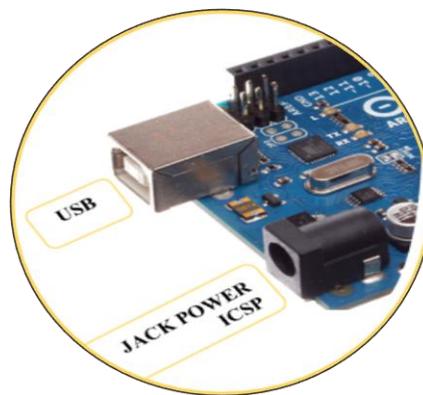


**Figura 19-2:** Interfaz Gráfica (GUIDE) Matlab  
Realizado por: Adriano Erica, 2018.

## 2.4. Fuente de alimentación

El diseño de dos sistemas de monitoreo para evaluar la actividad ansiolítica in vivo en ratas es totalmente independiente para la recolección y procesamiento de datos y para la alimentación en el caso de los receptores se puede obtener de la computadora a través de los puertos USB ya que proporcionan una salida de 5 voltios con un amperaje máximo de 500 mA.

Para el caso de los transmisores se empleará una fuente externa que estará conectada al Power Jack ICSP de Arduino, este adaptador de fuente emite 5 voltios a 2 Amperios. En la *Figura 20-2*, se puede observar las dos formas de alimentación.



**Figura 20-2:** Formas de Alimentación Arduino  
Realizado por: Adriano Erica, 2018.

### 2.4.1. Adaptador de Corriente

Se utilizará esta fuente para alimentar los arduinos de los transmisores ya que proporcionará el voltaje necesario y el amperaje idóneo para que cada uno de los dispositivos conectados al Arduino opere como es debido. En la *Figura 21-2* se puede observar el adaptador a utilizar.



**Figura 21-2:** Adaptador conectado a Arduino  
Realizado por: Adriano Erica, 2018.

Se decidió alimentar a los sistemas de esta manera ya que al adicionar otro tipo de componentes estos proporcionarán más peso además la alimentación será de forma permanente debido a que los sistemas no estarán en movimiento.

## 2.5. Implementación

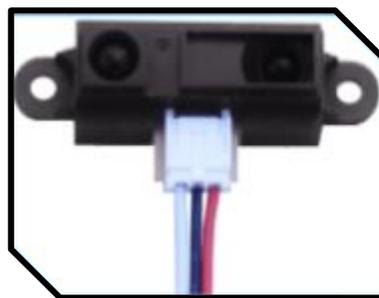
La implementación se realizó mediante los bloques descritos anteriormente para corroborar el correcto funcionamiento de los dispositivos en cada uno de los equipos implementados.

### 2.5.1. Modelo Claro/Oscuro (LDM)

Para indicar el desarrollo de este equipo se describirá por bloques cada uno de los procesos realizados.

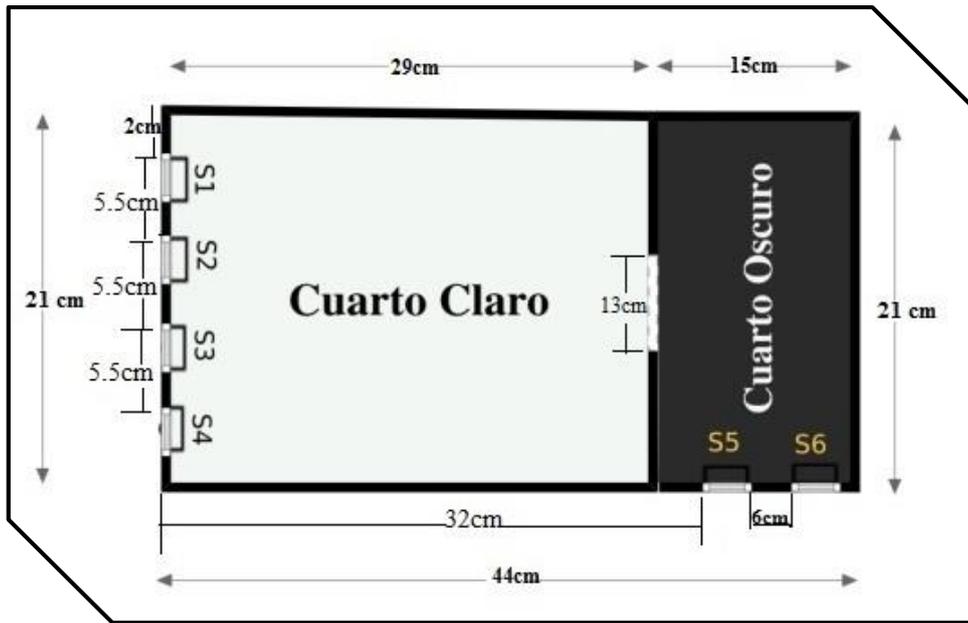
#### 2.5.1.1. Bloque de entrada

En la *Figura 22-2* se puede observar el sensor infrarrojo Sharp GP2Y0A21YK0F con su respectiva conexión para ser ubicados en su respectiva placa, siendo esta el transmisor, además se encuentran conectados a través de las entradas analógicas del Arduino permitiendo disponer de las entradas digitales para cumplir con el proceso de transmisión del transceiver.



**Figura 22-2:** Implementación de sensores (Bloque de Entrada)  
Realizado por: Adriano Erica, 2018.

Con la ayuda del programa Floorplanner en la *Figura 23-2* se puede apreciar en dos dimensiones (2D) la disposición de los seis sensores Sharp GP2Y0A21YK0F dentro de la caja que será usada para esta prueba, repartidos tanto en la zona clara así como en la zona oscura, la nomenclatura usada para definir los sensores es S seguido de un número para definir la posición (S1, S2..., etc.)



**Figura 23-2:** Disposición de los sensores. Vista 2D.  
 Realizado por: Adriano Erica, 2018.

En la *Figura 24-2* se visualiza la vista en tres dimensiones (3D) de la disposición de los sensores.

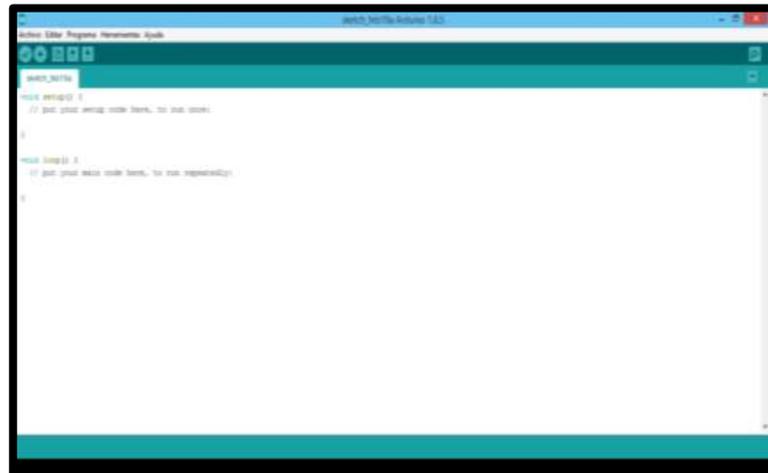


**Figura 24-2:** Disposición de sensores. Vista superior en 3D.  
 Realizado por: Adriano Erica, 2018.



### 2.5.1.2. Bloque de procesamiento

En este bloque es donde los datos emitidos por los sensores son procesados por medio de la plataforma Arduino, el procesamiento de esta información se rige bajo el entorno propio de Arduino, es decir bajo Arduino IDE que está basado en un lenguaje de programación de fácil entendimiento como se muestra en la *Figura 25-2*.



**Figura 25-2:** Arduino IDE.  
Realizado por: Adriano Erica, 2018.

En la *Figura 26-2* se puede apreciar parte del código usado para el transmisor de este sistema.

```
EmisorUno $
/
#include <DistanceGP2Y0A21YK.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <RF24_config.h>
#include <SPI.h>
DistanceGP2Y0A21YK Dist;
int ledon=A3;
int ledtx=A4;
int distance;
int msg[1]; // Array a transmitir
RF24 radio(9,10); // Creamos un objeto radio del tipo RF24
const uint64_t pipe = 0xE8E8F0F0E1LL; // Usamos este canal
void setup(void){
  Serial.begin(9600);
  radio.begin();
  Dist.begin(0);
  radio.openWritingPipe(pipe); // Abrir para escribir
  pinMode(ledon, OUTPUT);
  pinMode(ledtx, OUTPUT);
  digitalWrite(ledon, HIGH);
}
void loop(void){
  distance = (Dist.getDistanceCentimeter()*2);
  Serial.print("\nDistance in centimeters: ");
  Serial.print(distance);
  msg[0] = distance;
  radio.write(msg, 1);
  digitalWrite(ledtx, HIGH);
  delay(10);
  digitalWrite(ledtx, LOW);}
|
```

**Figura 26-2:** Código Transmisor Uno.  
Realizado por: Adriano Erica, 2018.

Cabe indicar que este bloque estará el módulo NRF24L01 para cumplir con el proceso de transmisión de los datos emitidos por los sensores hacia su receptor que estará conectado con la interfaz de Matlab.

Para el funcionamiento de este módulo este debe estar conectado directamente al Arduino y a su vez a los sensores para que cuando estos recojan los datos, la placa los procese y envíe a su respectivo módulo receptor.

Para que este módulo funcione correctamente debe estar alimentado a través del pin de 3.3v, ya que funciona en un rango de 1.9 a 3.6 voltios, si sobrepasa de este rango el módulo se quemará pudiendo incluso afectar a la placa de Arduino.

Los módulos receptores y uno de sus transmisores estarán conectados en la placa de Arduino Uno, el otro transmisor estará en la placa de Arduino Mega 2560, para su conexión se emplearán los pines digitales que indique el datasheet según sea el caso del Arduino a usar. En la *Tabla 5-2*, se describe la distribución de los pines según el Arduino a usar.

**Tabla 5-2:** Distribución de Pines módulo NRF24L01

<b>NRF24L01</b>	<b>ARDUINO UNO/NANO</b>	<b>ARDUINO MEGA</b>
<b>1: GND</b>	Pin GND	Pin GND
<b>2: VCC</b>	Pin 3.3V	Pin 3.3V
<b>3: CE</b>	Pin 9	Pin 9
<b>4: CSN</b>	Pin 10	Pin 10
<b>5: SCK</b>	Pin 13	Pin 52
<b>6: MOSI</b>	Pin 11	Pin 51
<b>7: MISO</b>	Pin 12	Pin 50
<b>8: IRQ</b>	Pin 2	-

**Realizado por:** Adriano Erica, 2018

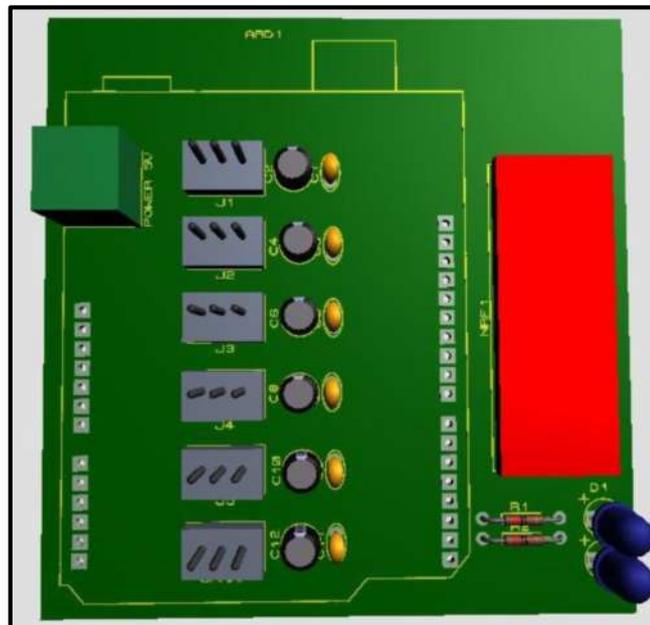
**Fuente:** <https://www.prometec.net/nrf2401/>

En la *Figura 27-2* se puede observar el módulo NRF24L01 conectado al Arduino Uno.



**Figura 27-2:** Conexión módulo NRF24L01 a Arduino Uno  
**Realizado por:** Adriano Erica, 2018.

En la *Figura 28-2* se puede distinguir la conexión en 3D del transmisor.



**Figura 28-2:** Conexión Transmisor LDM.  
**Realizado por:** Adriano Erica, 2018.

### 2.5.1.3. Bloque de almacenamiento

Los datos obtenidos durante el tiempo de duración de la pruebas son almacenados dentro de la aplicación para ser procesados e inmediatamente mostrar los resultados, en la *Figura 29-2* se observa el proceso de almacenamiento de cada dato recolectado, los mismos que se almacenarán en vectores para un fácil manejo de los datos.

```

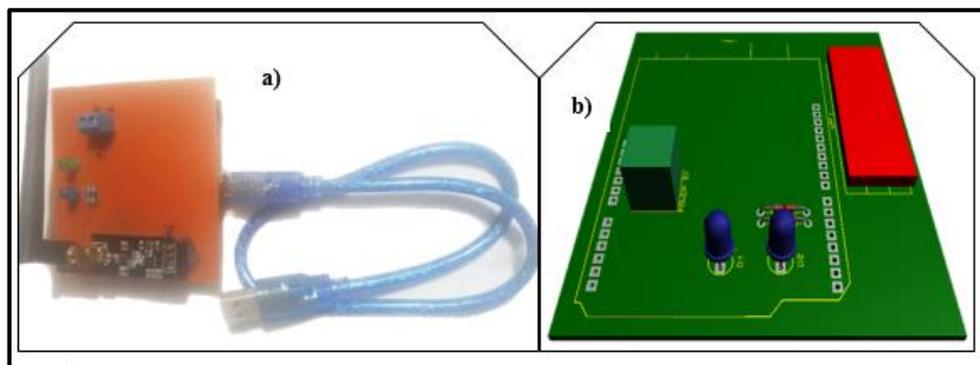
t=str2double(get(handles.edit1,'string'))*2690;
v1=zeros(1,t);
v2=zeros(1,t);
v3=zeros(1,t);
v4=zeros(1,t);
v5=zeros(1,t);
v6=zeros(1,t);
for k=1:t
    valor=fscanf(s, '%d,%d');
    valor1=fscanf(s, '%d,%d');
    valor2=fscanf(s, '%d,%d');
    v1(k)=valor(1);
    v2(k)=valor(2);
    v3(k)=valor1(1);
    v4(k)=valor1(2);
    v5(k)=valor2(1);
    v6(k)=valor2(2);
end
k=1:t

```

**Figura 29-2:** Almacenamiento de datos LDM.  
**Realizado por:** Adriano Erica, 2018.

#### 2.5.1.4. Bloque de interfaz de usuario

En el proceso de recepción se utilizó el módulo NRF24L01 el mismo que recibirá los resultados emitidos por su respectivo transmisor, este módulo se comunicará mediante comunicación serial para que los resultados obtenidos puedan ser visualizados, en la *Figura 30-2 a)* se observa la PCB del receptor con su respectivo cable serial, b) el diseño de la PCB en 3D



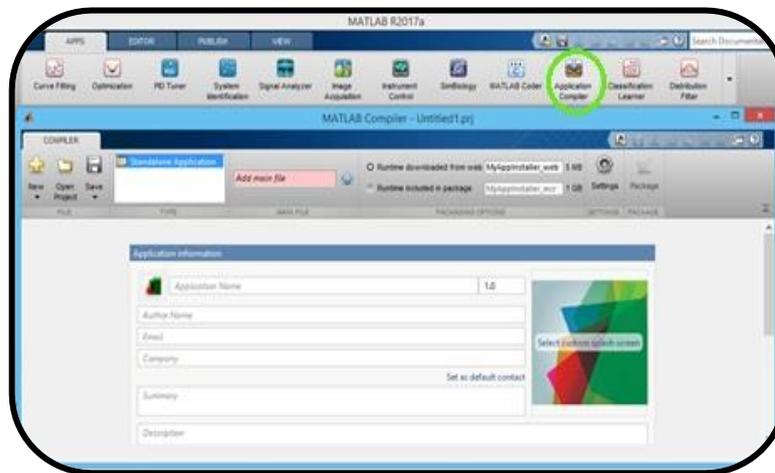
**Figura 30-2:** Receptor NRF24L01.  
**a)** PCB del Receptor con cable serial. **b)** PCB en 3D  
**Realizado por:** Adriano Erica, 2018.

La aplicación es muy amigable con el usuario, Matlab 2017a específicamente el Guide de Matlab es la herramienta que se utilizó para la programación de la misma, llamada según su sistema, es decir claro-oscuro para el LDM. En la *Figura 31-2* se puede observar esta aplicación.



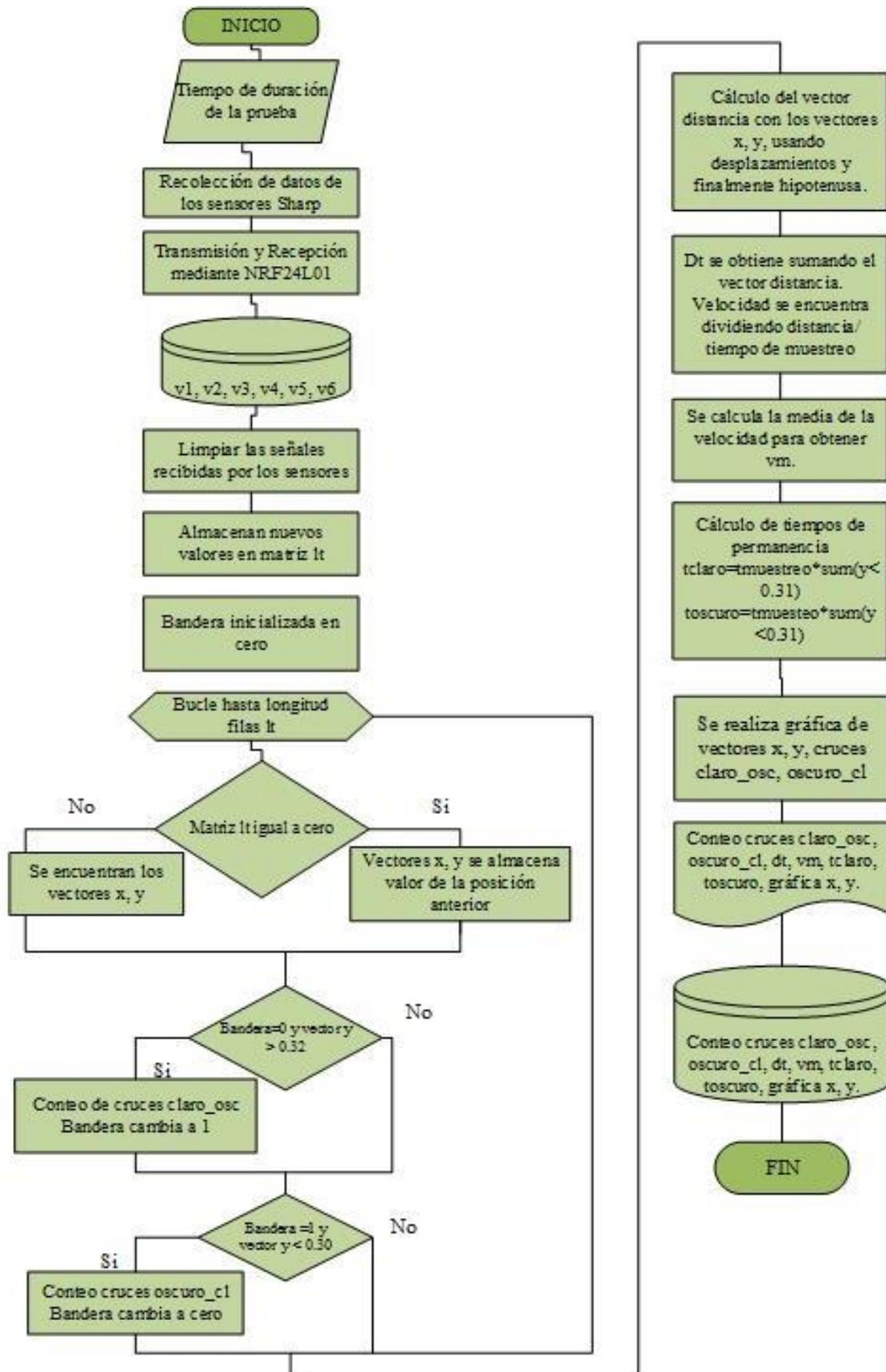
**Figura 31-2:** Interfaz de Usuario del sistema LDM.  
**Realizado por:** Adriano Erica, 2018.

Matlab permite crear archivos independientes mediante Matlab Compiler, un complemento que permite compartir programas de Matlab como aplicaciones autónomas, todas las aplicaciones usan Matlab Runtime que permite una implementación gratuita para usuarios que no requieran de Matlab. En la *Figura 32-2* se puede observar el área de ejecución de Matlab Compiler.



**Figura 32-2:** Matlab Compiler  
**Fuente:** Creado por la Autora

En la *Figura 33-2* se aprecia el diagrama de Flujo del funcionamiento de este sistema, donde se visualiza desde el momento en se ingresa el tiempo de duración de la prueba, el proceso de recolección procesamiento de los datos, hasta el momento final donde se produce el almacenamiento de los resultados obtenidos por parte del usuario. Para el almacenamiento de los valores emitidos por cada sensor se usa la nomenclatura de v1, v2, v3,..., v6, según el número del sensor al que pertenecen.



**Figura 33-2:** Diagrama de Flujo LDM.  
Realizado por: Adriano Erica, 2018.

## 2.5.2. Laberinto en cruz elevado (EPM)

Al igual que el sistema anterior la implementación de este sistema se describirá por bloques para un mayor entendimiento.

### 2.5.2.1. Bloque de entrada

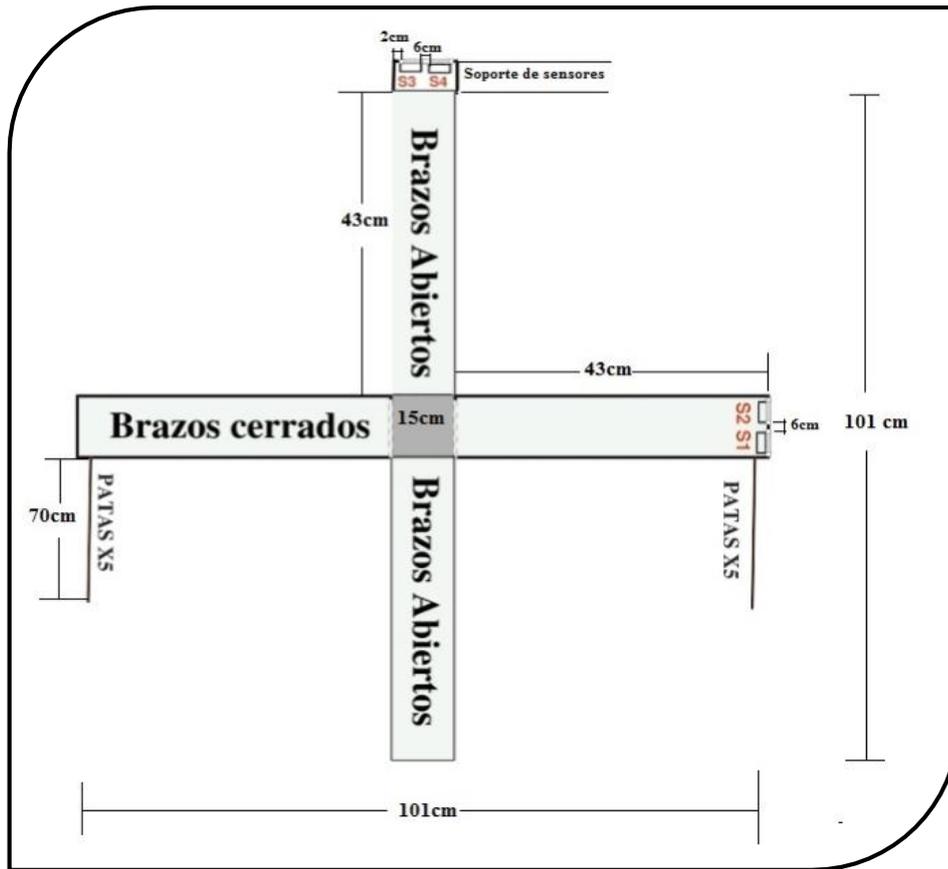
En la *Figura 34-2* se puede observar los sensores infrarrojos Sharp GP2Y0A60SZLF con su respectiva conexión para ser ubicados a la placa a la que pertenecen, siendo este el segundo transmisor, al igual que el modelo anterior se encuentran conectados a través de las entradas analógicas del Arduino Mega 2560, permitiendo disponer de las entradas digitales para cumplir con el proceso de transmisión del transceiver.



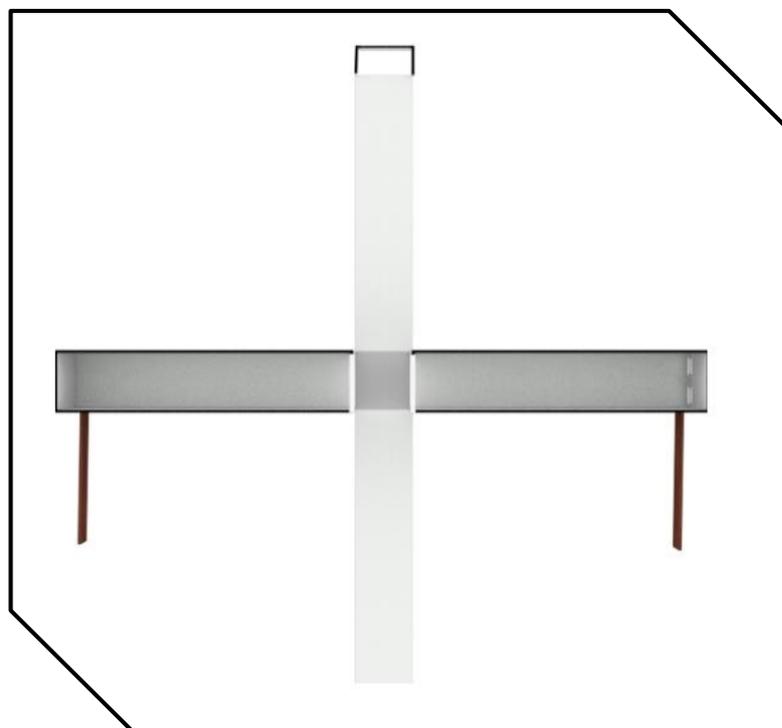
**Figura 34-2:** Implementación de sensores Sharp GP2Y0A60SZLF (Bloque de Entrada)  
Realizado por: Adriano Erica, 2018.

En la *Figura 35-2* se visualiza del mismo modo en un plano 2D, la disposición de los cuatro sensores Sharp GP2Y0A60SZLF, repartidos tanto para los brazos cerrados como para los brazos abiertos, para este equipo se usó un soporte para los sensores para de ese modo tener una buena ubicación de los mismos. Así como en el modelo anterior se usó la letra S para definir el sensor seguido del número correspondiente.

En cambio en la *Figura 36-2* se aprecia la vista superior en 3D del diseño de este modelo, donde se aprecian las únicamente patas usadas en los brazos cerrados para la elevación del suelo de este equipo, pero cabe indicar que se disponen de cinco patas repartidas dos para los brazos cerrados, dos en los brazos abiertos y una en el centro para brindar estabilidad en el uso de este equipo.



**Figura 35-2:** Ubicación de los sensores en EPM. Vista 2D.  
 Realizado por: Adriano Erica, 2018.



**Figura 36-2:** Vista Superior ubicación de los sensores en EPM, plano 3D.  
 Realizado por: Adriano Erica, 2018.



### 2.5.2.2. Bloque de procesamiento

En este bloque es donde los datos emitidos por los sensores son procesados por medio de la plataforma Arduino, al igual que el sistema anterior en este bloque se encuentra el módulo transceiver NRF24L01, el mismo que estará en el transmisor Mega y en el segundo receptor Arduino Uno, en la *Tabla 4-2* se encuentra la información respecto a los pines de conexión para un correcto funcionamiento de este módulo.

En la *Figura 37-2* se encuentra parte del código usado para el transmisor de este sistema.

```
EmisorMega5
#include <DistanceGP2Y0A21YK.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <RF24_config.h>
#include <SPI.h>
int ledon=24;
int ledtx=22;
int msg[1]; // Array a transmitir
RF24 radio(9,59); // Creamos un objeto radio del tipo RF24
const uint64_t pipe = 0xE8E8F0F0E1LL; // Usamos este canal
double Distancia1,Distancia2, val1, val2;
void setup(void){
  Serial.begin(9600);
  radio.begin();
  radio.openWritingPipe(pipe); // Abrir para escribir
  pinMode(ledon, OUTPUT);
  pinMode(ledtx, OUTPUT);
  digitalWrite(ledon, HIGH);}
void loop(void){
  int VAO = analogRead(A0);
  int VAL = analogRead(A1);
  Distancia1 = 187754 * pow(VAO, -1.51);
  Distancia2 = 187754 * pow(VAL, -1.51);
  int val1=Distancia1;
  int val2=Distancia2;
  val1=val1+100;
  radio.write( &val1, sizeof(double) );
  val2=val2+300;
  radio.write( &val2, sizeof(double) );
  digitalWrite(ledtx, HIGH);
  delay(10);}
```

**Figura 37-2:** Código Transmisor Mega  
Realizado por: Adriano Erica, 2018.

En la *Figura 38-2* se aprecia la conexión en 3D de este transmisor.



**Figura 38-2:** Conexión Transmisor Mega.  
Realizado por: Adriano Erica, 2018.

### 2.5.2.3. Bloque de almacenamiento

Los datos obtenidos durante el tiempo de duración de la prueba son almacenados dentro de la aplicación para ser procesados e inmediatamente mostrar los resultados, en la *Figura 39-2* se observa el proceso de almacenamiento de cada dato recolectado, los mismos que se almacenarán en vectores para una fácil manipulación de los mismos

```
v1=zeros(1,t);
v2=zeros(1,t);
v3=zeros(1,t);
v4=zeros(1,t);
for k=1:t
    valor=fscanf(s, '%d,%d');
    valor1=fscanf(s, '%d,%d');
    v1(k)=valor(1);
    v2(k)=valor(2);
    v3(k)=valor1(1);
    v4(k)=valor1(2);
end
v11=(101/80)*v1;
```

**Figura 39-2:** Bloque de almacenamiento EPM.  
Realizado por: Adriano Erica, 2018.

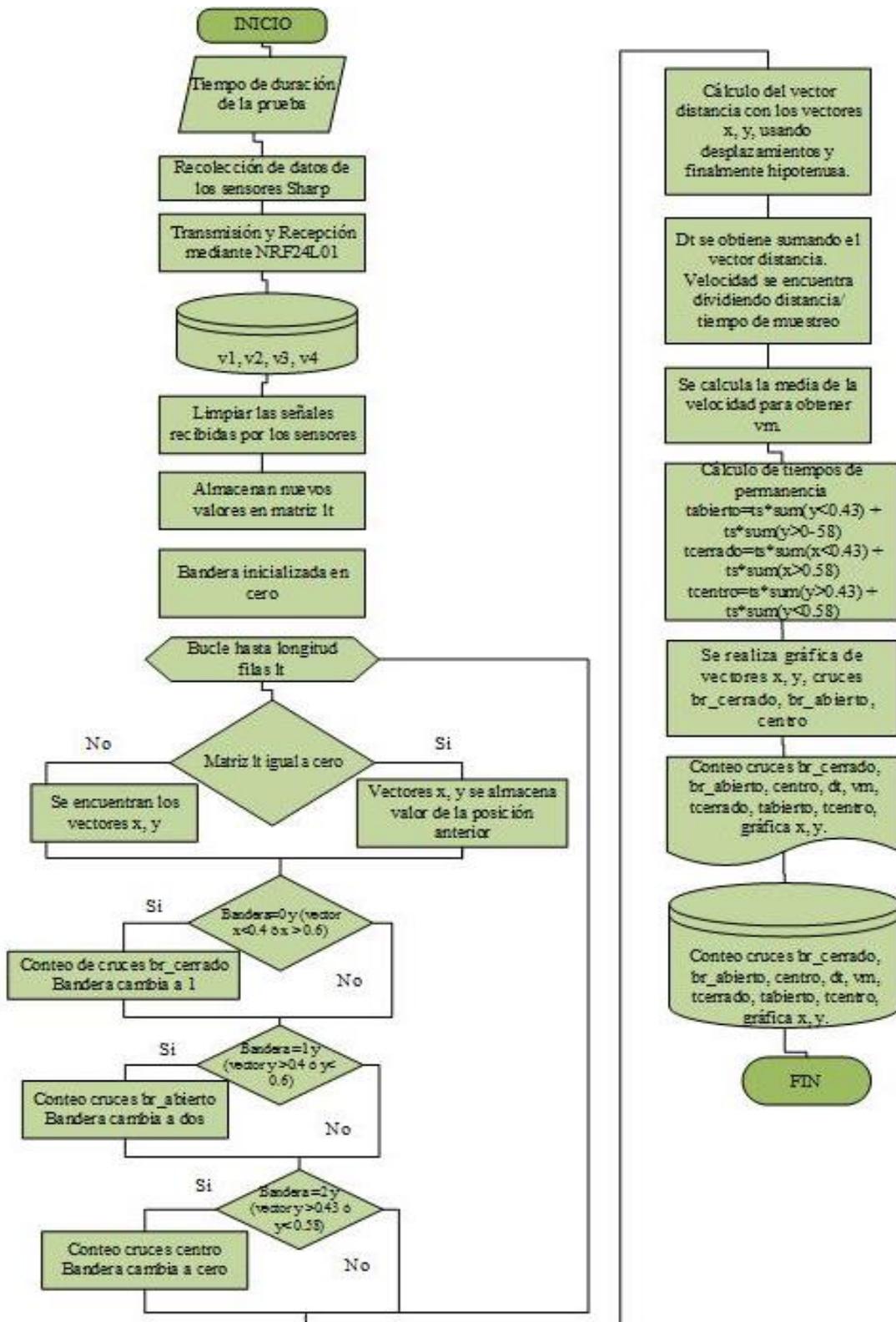
### 2.5.2.4. Bloque de interfaz de usuario

La aplicación es amigable con el usuario llamada Laberinto para ser identificada por el sistema al que pertenece, del mismo modo se emplea comunicación serial para poder comunicar Matlab con el receptor Arduino. En la *Figura 40-2* se visualiza la ventana que el usuario manejará en este sistema.



**Figura 40-2:** Interfaz de Usuario EPM.  
Realizado por: Adriano Erica, 2018.

En la *Figura 41-2* se aprecia el diagrama de flujo para el funcionamiento del sistema, desde la inserción del tiempo de duración, la manipulación de los datos recolectados hasta la visualización final por parte del usuario. Para la ejecución del mismo se hizo uso del software Visio.



**Figura 41-2:** Diagrama de Flujo EPM.  
Realizado por: Adriano Erica, 2018.

### 2.5.3. *Protección física de los sistemas*

Para la protección de los elementos contra golpes, abolladuras, maltratos se ha diseñado una caja protectora acorde a los dispositivos usados tal como se puede observar en la *Figura 42-2*, además esta caja cuenta con las aberturas necesarias para que puedan salir las antenas de los módulos transceiver para que puedan transmitir y recibir los datos así como los cables seriales de los receptores.



**Figura 42-2:** Caja protectora.  
**Realizado por:** Adriano Erica, 2018.

## CAPÍTULO III

### 3. EVALUACIÓN Y COMPARACIÓN DE RESULTADOS

#### 3.1. Introducción

En este capítulo se describen las pruebas realizadas de los sistemas, los procedimientos de implementación tanto en el modelo claro/oscura así como en el laberinto en cruz elevado, las mediciones de distancia para posteriormente obtener la velocidad media, tiempos de permanencia, número de cruces.

Una vez realizada la implementación física se procede a realizar pruebas con las respectivas aplicaciones ejecutables realizadas en Matlab, las cuales mediante el receptor transceiver reciben los datos del circuito y los visualiza en la pantalla dando los resultados requeridos para la evaluación ansiolítica tales como son distancia total, velocidad media, tiempos de permanencia, número de cruces además nos muestra una gráfica de la trayectoria seguida por el roedor.

- **Comportamiento de los roedores en un ambiente normal y con el uso del ansiolítico.**

Las ratas son mamíferos cosmopolitas que se adaptan a una mayor diversidad de ambientes, de hábito nocturno y su comportamiento está influenciado por feromonas, su sentido de audición es muy agudo, por lo que se debe tener cuidado con los equipos que se utilizan. Su olfato es muy sensible, no solo para detectar comida y depredadores, sino también para percibir un orden social. Son sociales, forman grupos y acostumbran acicalarse unas a otras. Su conducta comportamental en estos sistemas se describe a continuación:

*Modelo claro/oscura:* En esta prueba se evalúa la mayor o menor preferencia que los roedores muestran con respecto al compartimiento claro, a la cuál presentan una “fobia” innata la mayor parte del tiempo, dicha respuesta fóbica o ansiosa, resulta disminuida tras la administración de fármacos ansiolíticos, de tal manera que los individuos tratados con el fármaco realizan mayor conducta exploratoria del compartimiento claro. (Fernández, A., 2008)

*Laberinto en cruz elevado:* Habitualmente los roedores muestran cierta preferencia por los brazos cerrados, aventurándose muy poco a la exploración de los brazos abiertos, donde se encuentran más expuestos, con el uso de fármacos ansiolíticos se mejora la frecuencia entradas y permanencia en los brazos abiertos.

### 3.2. Pruebas y evaluación de los sistemas de monitoreo

Al concluir con la implementación se procedió a la ubicación de los sistemas de monitoreo dentro de sus respectivas pruebas, en la *Figura 1-3 a* se puede observar el Modelo Claro/Oscuro con sus respectivos sensores, en la *Figura 1-3 b* se observa en cambio el EPM en sus respectivos brazos ubicados los sensores.



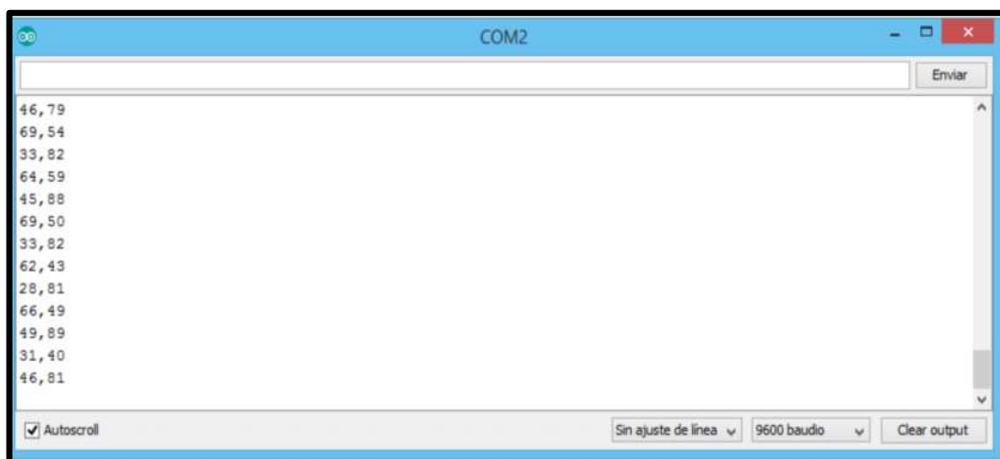
**Figura 1-3:** Ubicación sistemas de monitoreo.

a) Sensores LDM b) Sensores EPM

Realizado por: Adriano Erica, 2018.

#### 3.2.1. Pruebas de transmisión de datos

En las pruebas de transmisión de datos desde el Arduino hacia la aplicación de Matlab el receptor se conecta mediante serial hacia la computadora donde estará instalada la aplicación. En la *Figura 2-3* se puede observar la transmisión de datos desde el Arduino hacia su receptor empleando NRF24L01.

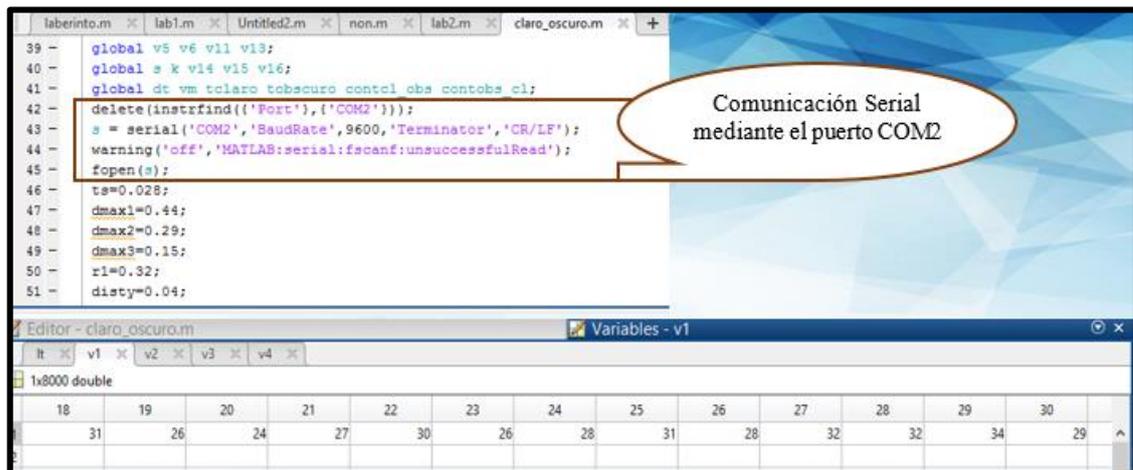


**Figura 2-3:** Transmisión de datos desde el Arduino Mega.

Realizado por: Adriano Erica, 2018.

### 3.2.2. Recepción de datos en la aplicación de Matlab

La recepción de datos en la aplicación de Matlab tanto para el Modelo Claro/Oscuro como para el Laberinto en Cruz Elevado se realiza a través del módulo NRF24L01 cuyo receptor se comunica con la PC mediante comunicación serial a través del puerto COM2, con la misma velocidad que envían los arduinos, se empleó la velocidad estándar de 9600 baudios, en la *Figura 3-3* se puede observar la recepción de los datos emitidos por Arduino, los mismos que son almacenados en vectores para facilitar el procesamiento de los datos recibidos.



The screenshot shows a MATLAB editor window with several tabs. The active tab is 'claro\_oscuro.m'. The code in the editor includes the following lines:

```
39 - global v5 v6 v11 v13;
40 - global s k v14 v15 v16;
41 - global dt vm tclaro toboscuro contcl obs contobs cl;
42 - delete(instrfind({'Port'}, {'COM2'}));
43 - s = serial('COM2', 'BaudRate', 9600, 'Terminator', 'CR/LF');
44 - warning('off', 'MATLAB:serial:fscanf:unsuccessfulRead');
45 - fopen(s);
46 - ts=0.028;
47 - dmax1=0.44;
48 - dmax2=0.29;
49 - dmax3=0.15;
50 - r1=0.32;
51 - disty=0.04;
```

A callout bubble points to the serial communication code with the text: "Comunicación Serial mediante el puerto COM2".

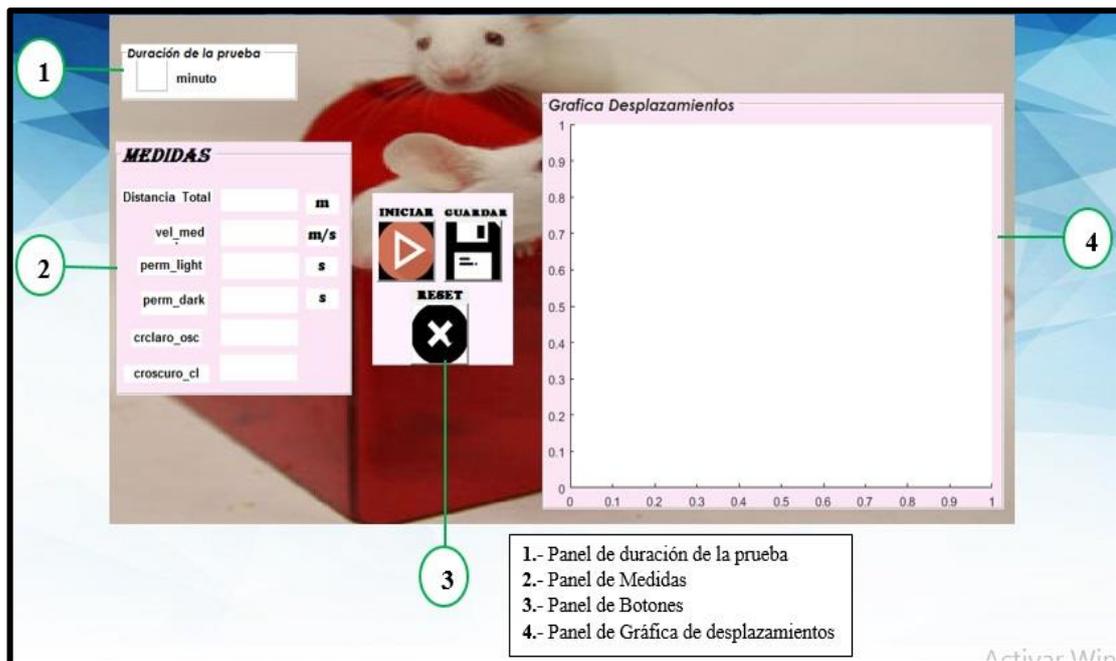
Below the code editor is a window titled "Variables - v1" showing a table of data:

18	19	20	21	22	23	24	25	26	27	28	29	30
31	26	24	27	30	26	28	31	28	32	32	34	29

**Figura 3-3:** Recepción de datos mediante comunicación serial  
Realizado por: Adriano Erica, 2018.

#### 3.2.2.1. Aplicación sistema de monitoreo MCL/EPM

A continuación se describirá el funcionamiento de la aplicación para el manejo del sistema de monitoreo, así como de los botones con los que cuenta para que el usuario pueda hacer buen uso de la misma. En la *Figura 4-3* se puede observar la aplicación con sus respectivas partes, las mismas que serán detalladas a continuación.



**Figura 4-3:** Aplicación MCL/EPM con sus respectivos paneles.

Fuente: Creado por la Autora.

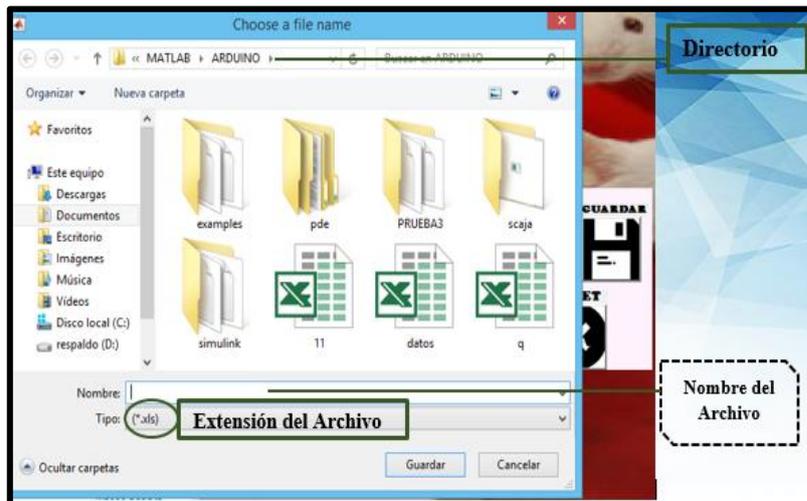
**Panel 1: Duración de la prueba.-** En esta parte el usuario debe ingresar el tiempo en minutos que durará la prueba a realizar, es decir si la prueba dura cinco minutos, en el *EditText* (parte en blanco) debe ingresar el número “5”, solo se aceptan valores numéricos.

**Panel 2: Medidas.-** Una vez finalizada la prueba se mostrarán los resultados finales como son la distancia total en metros, velocidad media en metros/segundos, tiempos de permanencia en segundos, número de cruces adimensional.

**Panel 3: Botones.-** En este panel se puede encontrar tres botones: Iniciar, Guardar y Reset.

- a) **Iniciar.-** Este botón comienza el procesamiento de los datos, es decir inicia con la comunicación serial entre Arduino y Matlab, así como el procesamiento de los valores emitidos, los mismos que serán visualizados al finalizar la misma en el panel de medidas. En caso de no existir la comunicación serial se emite un mensaje de error, por lo cual no se puede dar inicio a la ejecución de la prueba.
- b) **Guardar.-** Este botón permite guardar las medidas obtenidas en un archivo de Excel (.xls), del mismo modo guarda la gráfica de la trayectoria con una extensión .jpg, el usuario será capaz de indicar el directorio donde desea que se guarden los resultados así como indicar el nombre con el que se guardarán. En la *Figura 5-3* se puede observar el modo en que se presentan las ventanas para guardar los datos.





**Figura 5-3:** Ventana Emergente botón Guardar  
Realizado por: Adriano Erica, 2018.

- c) **Reset.-** Este botón borra la Gráfica de Desplazamientos, los *EditText* tanto del panel de duración de la prueba como los del Panel de Medidas, lo cual permite al usuario continuar con la realización de las pruebas respectivas.

**Panel 4: Gráfica de Trayectoria.-** En esta área se visualiza la trayectoria realizada por la rata durante el tiempo que permanece en el LDM o en el EPM. Esta gráfica se muestra una vez finalizada la prueba al igual que las medidas.

### 3.3. Pruebas de campo

Una vez finalizados con las pruebas de transmisión y recepción de datos del sistema de monitoreo, sabiendo que las mismas funcionan adecuadamente se procedió a realizar las respectivas pruebas de campo real. Estas pruebas se realizaron empleando 6 grupos de roedores, 5 por cada grupo, 15 machos serán usados para el Modelo Claro/Oscuro y 15 hembras para el Laberinto en Cruz Elevado, utilizando sus respectivos identificadores, los cuales se detallan en la *Tabla 1-3*.

**Tabla 1-3:** Nomenclatura de las ratas a utilizar

<i>Identificador</i>	<i>Detalle</i>
<i>k1</i>	Sin estrés y sin tratamiento
<i>k2</i>	Con estrés y sin tratamiento
<i>k3</i>	Sin estrés y con vehículo
<i>k4</i>	Con estrés con tratamiento
<i>k5</i>	No Manipulado

Realizado por: Adriano Erica, 2018

Las características de las especies para ambas pruebas se detallan en la *Tabla 2-3*.

**Tabla 2-3:** Conjunto de individuos para la evaluación ansiolítica.

GRUPOS	GRUPO 1	GRUPO 2	GRUPO 3	GRUPO 4	GRUPO 5	GRUPO 6
Número de roedores	5	5	5	5	5	5
Sexo	Machos			hembras		
Edad	8-12 semanas					
Peso promedio	250-300g					

Realizado por: Adriano Erica, 2018

### 3.3.1. Evaluación modelo claro/oscuro

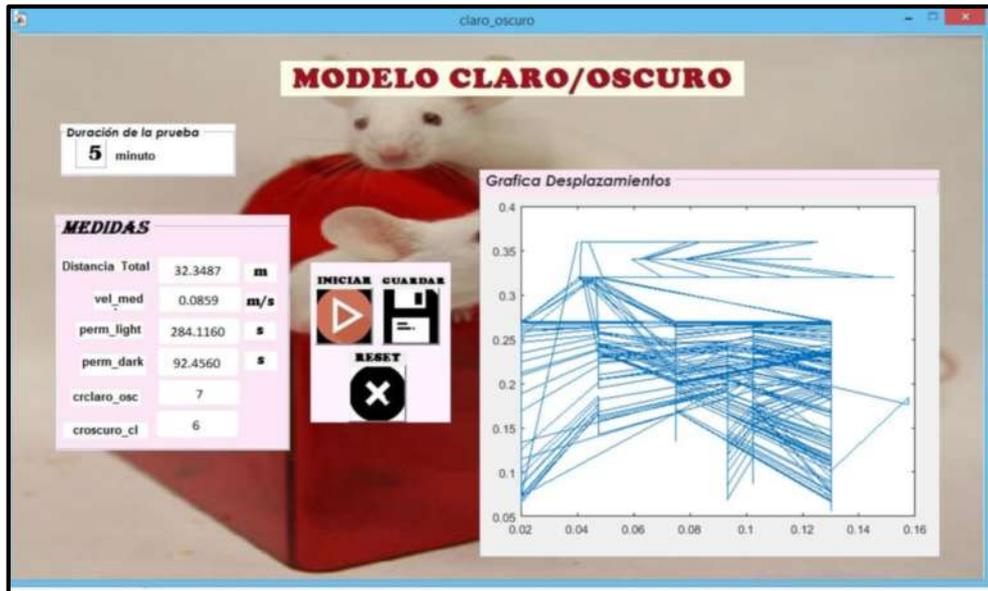
Para la validación de este sistema se empleó ejemplares machos, los mismos que fueron identificados tal como se indica en la *Tabla 1-3*. Para llevar a cabo esta prueba se debe disponer de un espacio totalmente aislado, en la *Figura 6-3*, se puede observar el escenario para llevar a cabo las pruebas ansiolíticas, donde se dispone de una cámara para posteriormente poder comparar los resultados obtenidos por el sistema y los observados en el video registrado.



**Figura 6-3:** Escenario para pruebas ansiolíticas.

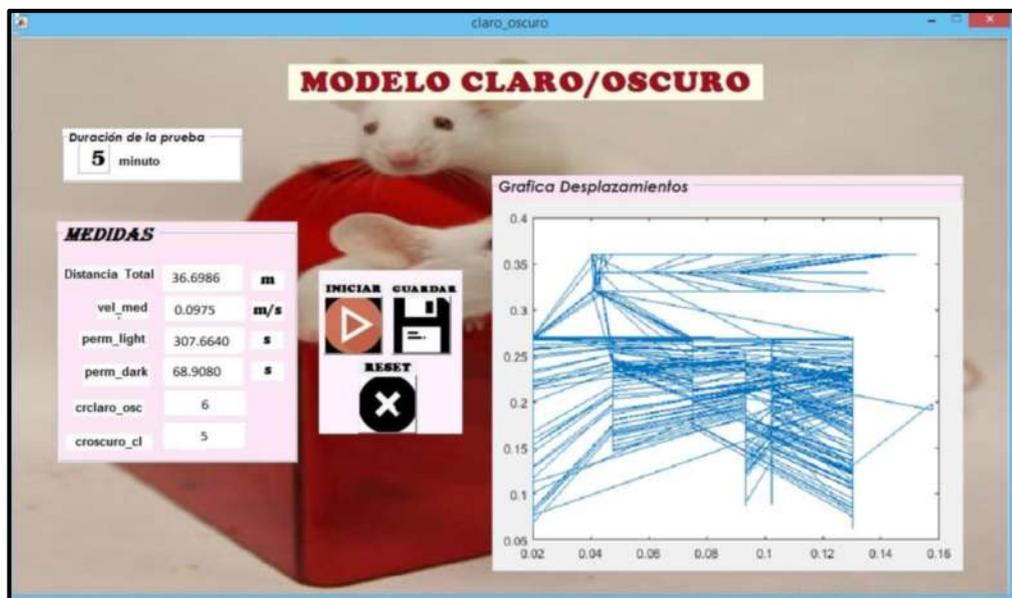
Realizado por: Adriano Erica, 2018.

Se procedió a realizar las pruebas en los individuos seleccionados, la duración de cada prueba es de cinco minutos, en la *Figura 7-3* se puede observar los resultados obtenidos en el individuo *k1*.



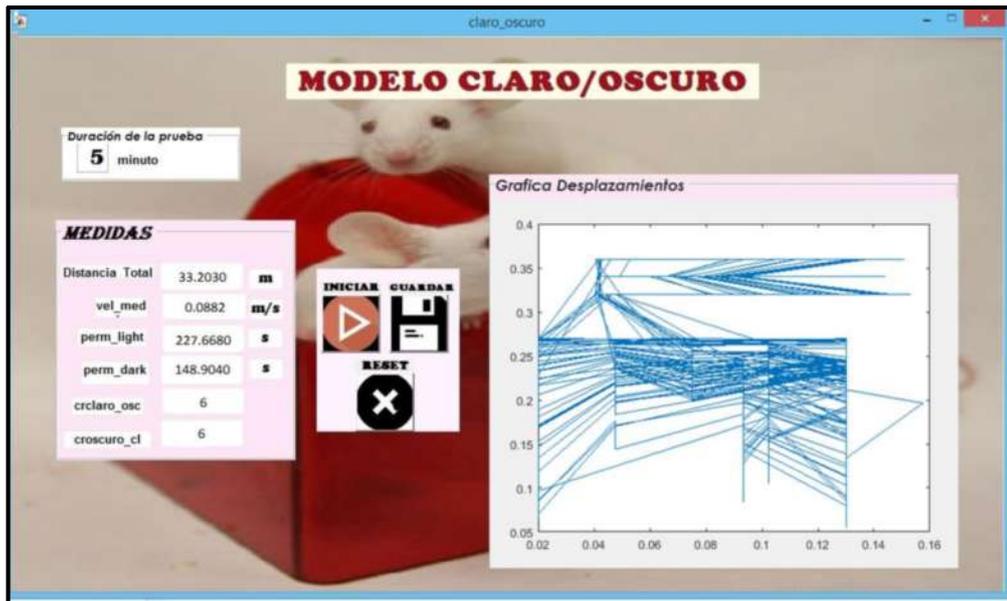
**Figura 7-3:** Resultados obtenidos en *k1*  
Realizado por: Adriano Erica, 2018.

En la *Figura 8-3* se visualiza los resultados obtenidos en el individuo *k2*.



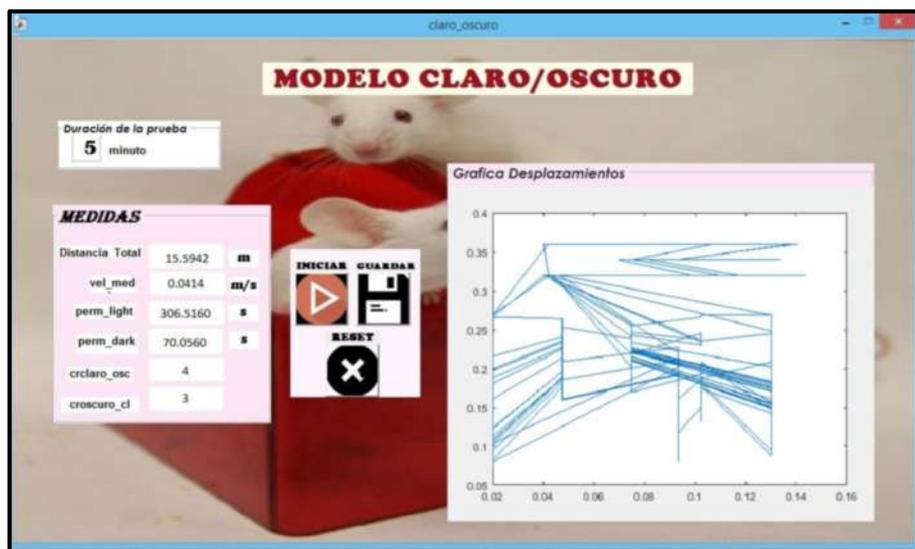
**Figura 8-3:** Resultados Obtenidos en *k2*  
Realizado por: Adriano Erica, 2018.

En la *Figura 9-3* se aprecia los resultados alcanzados por el individuo *k3*.



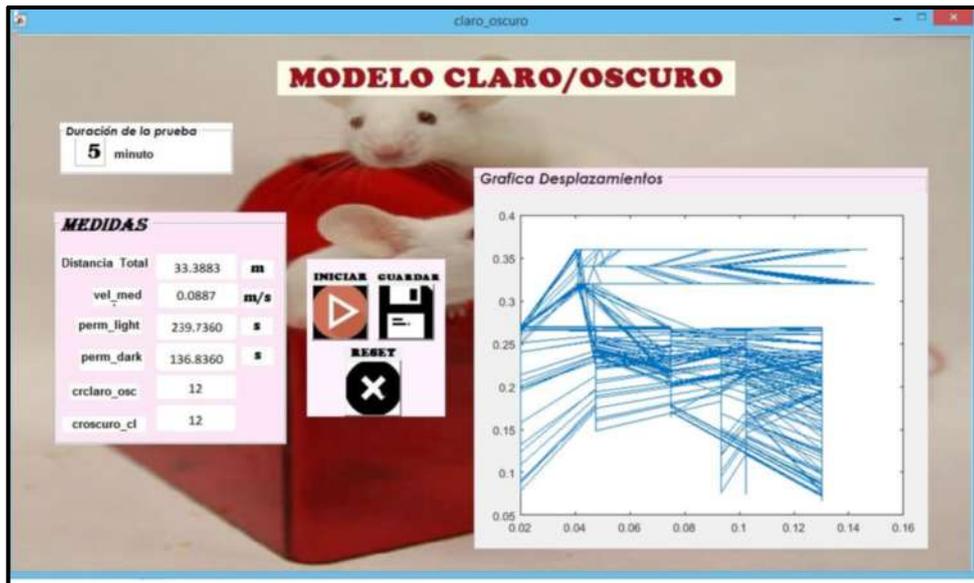
**Figura 9-3:** Resultados Obtenidos en k3  
Realizado por: Adriano Erica, 2018.

En la *Figura 10-3* se representa los valores finales en el individuo k4.



**Figura 10-3:** Resultados Obtenidos en k4  
Realizado por: Adriano Erica, 2018.

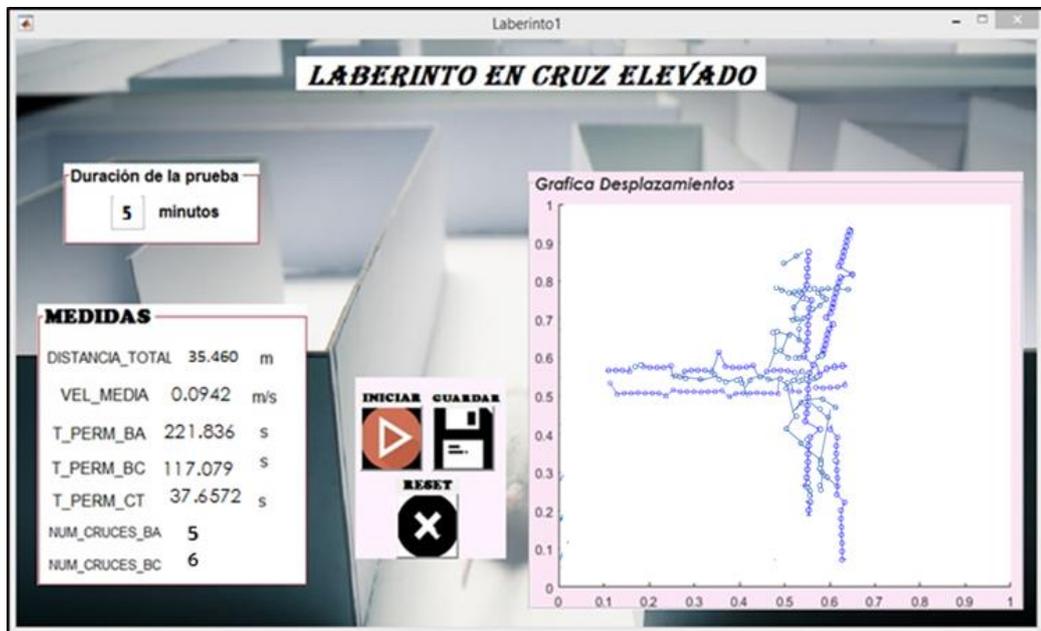
En la *Figura 11-3* se concibe las derivaciones emanadas por el individuo k5.



**Figura 11-3:** Resultados Obtenidos en k5  
 Realizado por: Adriano Erica, 2018.

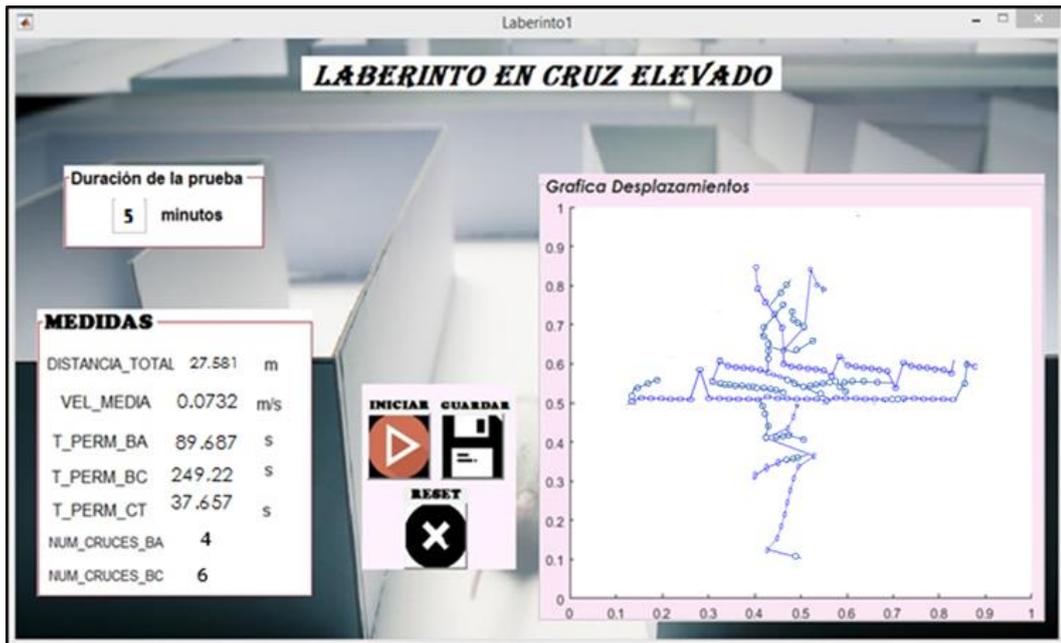
### 3.3.2. Evaluación laberinto cruz elevado

Para la validación de este equipo se utilizó hembras las mismas que al igual que la prueba anterior serán identificadas como se indica en la *Tabla 1-3*. La duración de esta prueba es de cinco minutos, en la *Figura 12-3* se puede observar los resultados obtenidos en el individuo k1.



**Figura 12-3:** Resultados obtenidos en k1 (EPM)  
 Realizado por: Adriano Erica, 2018.

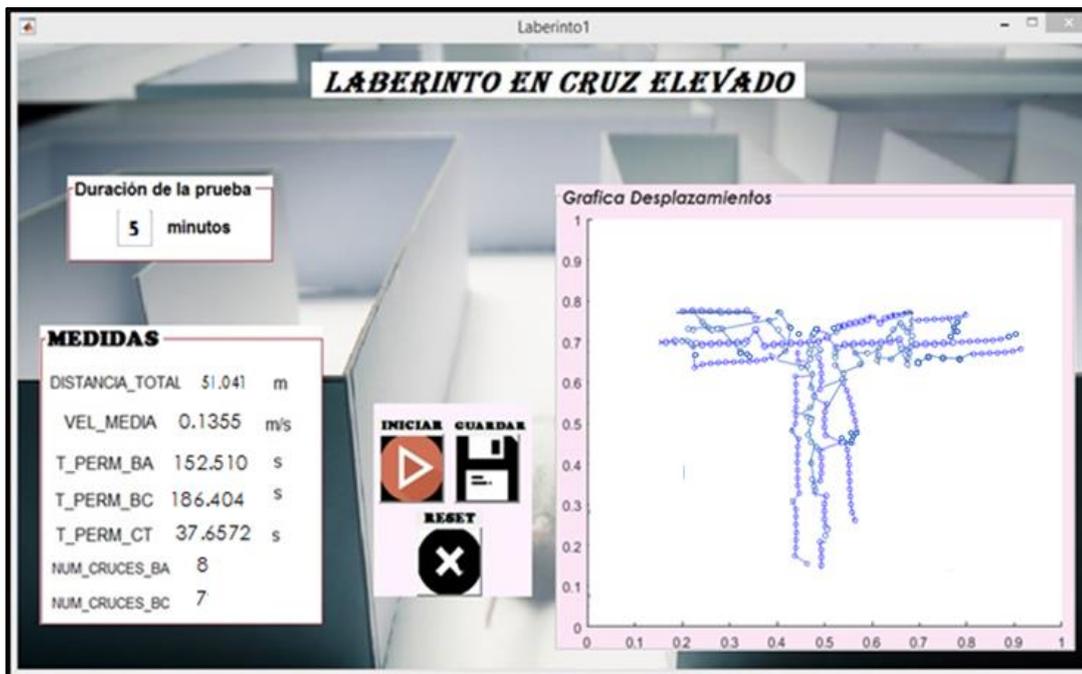
En la *Figura 13-3* se visualiza los valores emitidos por la ejemplar k2.



**Figura 13-3:** Resultados obtenidos en k2 (EPM)

Realizado por: Adriano Erica, 2018.

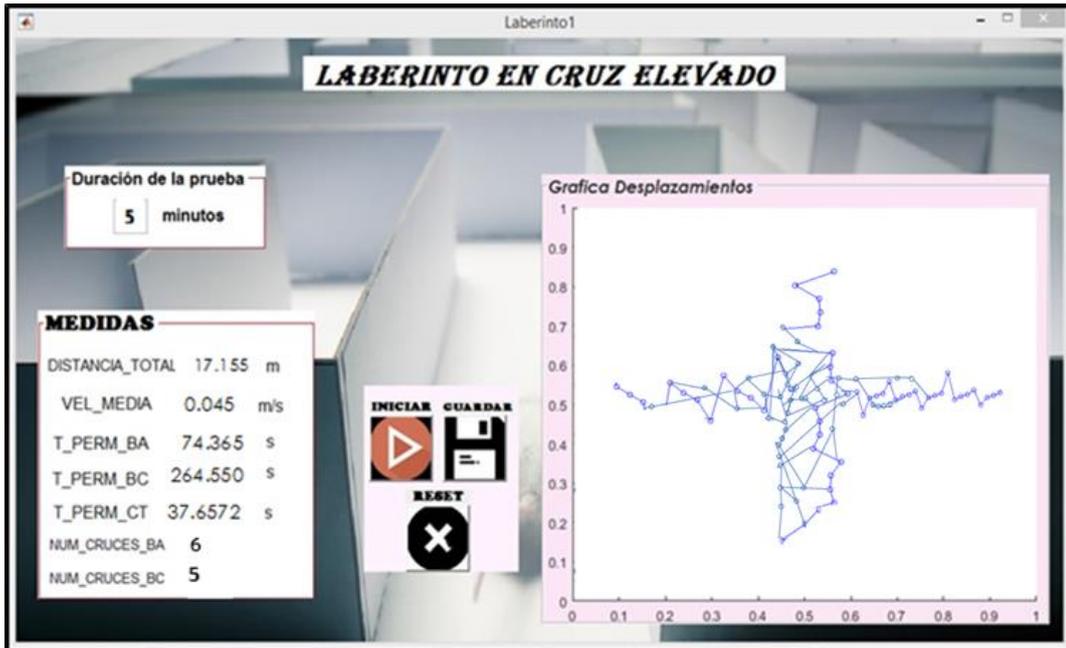
En la *Figura 14-3* se puede apreciar los resultados emanados por el individuo k3.



**Figura 14-3:** Resultados obtenidos en k3 (EPM)

Realizado por: Adriano Erica, 2018.

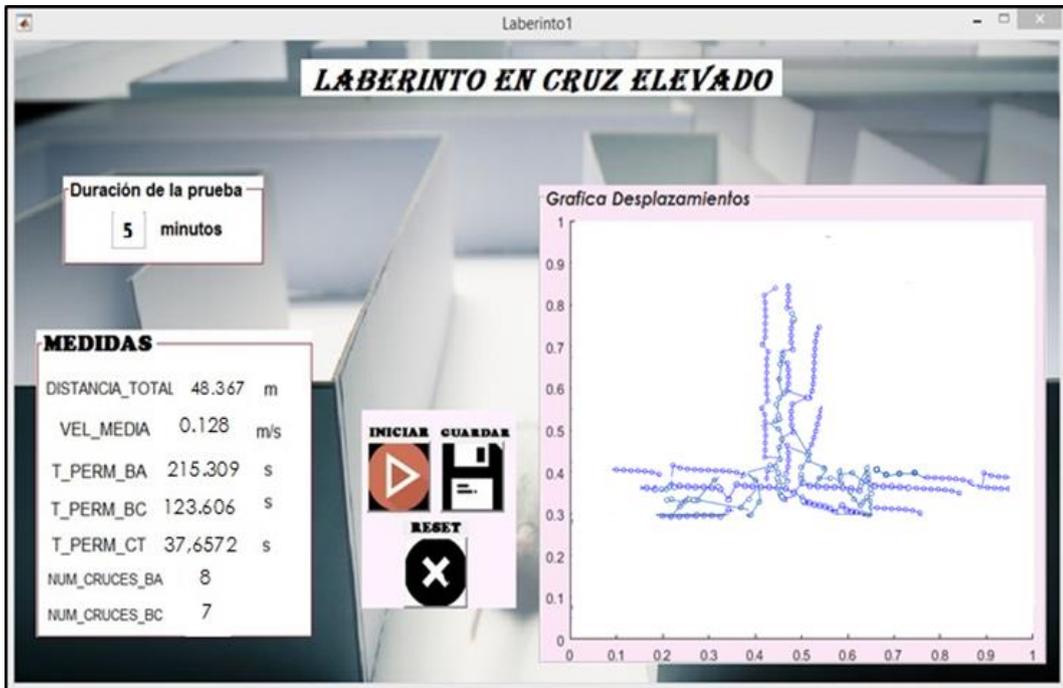
En la *Figura 15-3* se distingue las derivaciones conseguidas por k4.



**Figura 15-3:** Resultados obtenido en k4 (EPM)

Realizado por: Adriano Erica, 2018.

En la *Figura 16-3* se aprecia los valores presentados por k5.



**Figura 16-3:** Resultados obtenidos en k5 (EPM)

Realizado por: Adriano Erica, 2018.

### 3.4. Banco de pruebas del sistema de monitoreo

Para la validación de los equipos se realizó las pruebas en 30 especies, divididas 15 machos para el LDM y 15 hembras para el EPM. El tiempo de duración fue de tres semanas por cada equipo.

#### 3.4.1. Análisis del modelo claro oscuro

Para comprobar la confiabilidad de este sistema se comparó los resultados obtenidos mediante las pruebas manuales, es decir mediante la grabación de video con la ayuda de una cámara fotográfica, con las pruebas automatizadas. Por cada día se hizo uso de 5 especies, en horario diurno entre 10-11:30 am, con un intervalo de separación de aproximadamente 3 minutos por prueba. En la *Tabla 3-3*, se puede apreciar el análisis del LDM, además con la ayuda del *software R* se realizó el cálculo del promedio, desviación estándar y coeficiente de variación.

**Tabla 3-3:** Análisis Modelo Claro/Oscuro (Ligth/Dark Model)

PRUEBA MANUAL						
Número de Muestra	T. Claro (s)	T. Oscuro (s)	Cruces	ERROR CRUCES	E. T Claro	E. T Oscuro
1	254,000	62,000	13	0	0,116	0,456
2	278,000	39,000	11	0	0,336	0,092
3	198,000	119,000	12	0	0,332	0,096
4	277,000	40,000	7	0	0,484	0,056
5	210,000	107,000	25	1	0,264	0,164
6	216,000	101,000	26	0	0,412	0,016
7	170,000	147,000	19	1	0,080	0,348
8	140,000	176,000	24	1	0,240	0,332
9	347,000	0,000	0	0	0,428	0,000
10	275,000	41,000	18	0	0,172	0,400
11	70,000	247,000	22	1	0,348	0,080
12	216,000	100,000	22	1	0,484	0,088
13	139,000	177,000	48	3	0,456	0,116
14	53,000	264,000	22	0	0,372	0,056
15	209,000	107,000	33	1	0,232	0,340
<b>Promedio</b>	203,467	115,133	20,120			
<b>Desviación Estándar</b>	79,915	76,387	11,370			
<b>Coefficiente de Variación</b>	0,392767	0,663465	0,5647			
<b>MEDIA ERRORES</b>				0,600	0,317	0,176



PRUEBA AUTOMATIZADA						
Número de Muestra	Distancia (m)	Vel_ Media (m/s)	T. Claro (s)	T. Oscuro (s)	Cruces	ERROR CRUCES
1	32,348	0,0859	254,116	62,456	13	0
2	36,698	0,0975	277,664	38,908	11	0
3	33,203	0,0882	197,668	118,904	12	0
4	15,594	0,0414	276,516	40,056	7	0
5	33,388	0,0887	209,736	106,836	24	0
6	45,628	0,1212	215,588	100,984	26	0
7	28,367	0,0753	169,920	146,652	19	1
8	29,797	0,0791	140,240	176,332	24	1
9	20,865	0,0554	346,572	0,000	0	0
10	44,688	0,1187	275,172	41,400	18	0
11	27,581	0,0732	69,652	246,920	22	1
12	35,460	0,0942	216,484	100,088	23	0
13	51,041	0,1355	139,456	177,116	47	2
14	17,155	0,0456	52,628	263,944	23	1
15	48,367	0,1284	209,232	107,340	33	1
<b>Promedio</b>	33,345	0,0886	203	115	20	
<b>Desviación Estándar</b>	10,8187	0,02872	79,8789	76,334	11,1986	
<b>Coefficiente de Variación</b>	0,32444	0,32440	0,3927	0,6627	0,5562	
<b>Media Error cruces</b>						0,467

Realizado por: Adriano Erica, 2018

Una vez realizada la medición y con los datos conseguidos en la *Tabla 3-3*, indica que los resultados obtenidos son confiables con una tasa de error de 0,467 la cual es baja en comparación a la tasa de error de las pruebas manuales que es de 0,6. Los valores obtenidos por el sistema son confiables y pueden ser tomados como referencia.

### 3.4.2. Análisis del laberinto en cruz elevado

Para validar cada uno de los resultados obtenidos se realizó la comparación de los resultados proporcionados en forma manual, es decir mediante la grabación de video con la ayuda de una cámara fotográfica, con los valores emitidos por la forma automatizada. Estas pruebas se llevaron a cabo en un horario diurno de 9:30 – 11:30a.m, con un intervalo de separación de 3 minutos entre cada prueba. En la *Tabla 4-3* se aprecia el análisis del EPM, además con la ayuda del *software R* se realizó el cálculo del promedio, desviación estándar y coeficiente de variación.

**Tabla 4-3:** Análisis Laberinto en Cruz Elevado (EPM)

<b>PRUEBA MANUAL</b>								
Número de Muestra	TBAbierto (s)	TBCerrado (s)	TCentro (s)	Cruces	E.CRUCES	E.TAbierto	E.TCerrado	E.TCentro
1	90	249	38	21	1	0,313	0,22	0,343
2	222	117	38	22	1	0,164	0,079	0,343
3	153	186	38	25	2	0,49	0,404	0,343
4	74	265	38	23	0	0,365	0,45	0,343
5	215	124	38	33	2	0,309	0,394	0,343
6	208	126	43	37	2	0,068	0,213	0,148
7	189	145	43	20	0	0,331	0,166	0,068
8	227	107	43	30	1	0,145	0,21	0,366
9	105	228	44	23	0	0,062	0,499	0,008
10	237	99	40	28	0	0,163	0,086	0,321
11	256	83	38	13	0	0,296	0,21	0,343
12	182	159	36	19	0	0,073	0,013	0,342
13	157	190	30	23	0	0,38	0,175	0,125
14	221	126	30	24	0	0,443	0,111	0,125
15	229	67	18	11	0	0,19	0,254	0,128
<b>Promedio</b>	184,333	151,4	37	23,4666				
<b>Desviación Estándar</b>	56,7647	60,2539	6,71884	6,79145				
<b>Coefficiente de Variación</b>	0,30794	0,3979	0,1815	0,2894				
<b>MEDIA ERRORES</b>					<b>0,6</b>	<b>0,2528</b>	<b>0,232266</b>	<b>0,24593</b>
<b>PRUEBA AUTOMATIZADA</b>								
Número de Muestra	Distancia (m)	Vel_Media (m/s)	T_BAbierto (s)	T_BCerrado (s)	TCentro (s)	Cruces	ERROR CRUCES	
1	27,581	0,0732	89,687	249,220	37,657	11	0	
2	35,460	0,0942	221,836	117,079	37,657	10	0	
3	51,041	0,1355	152,51	186,404	37,657	15	1	
4	17,155	0,045	74,365	264,55	37,657	11	0	
5	48,367	0,128	215,309	123,606	37,657	15	1	
6	40,8819	0,1086	207,932	125,787	42,852	18	2	
7	35,3591	0,0939	188,669	144,834	43,068	20	0	
8	54,3007	0,1442	227,145	106,79	42,634	17	1	
9	25,0957	0,0666	105,062	227,501	44,008	23	0	
10	51,2506	0,1361	237,163	99,086	40,321	28	0	
11	32,3487	0,0859	255,704	83,21	37,657	13	0	
12	28,3671	0,0753	181,927	158,987	35,658	19	0	
13	29,7973	0,0791	156,62	189,825	30,125	23	0	
14	33,3883	0,0887	220,557	125,889	30,125	24	0	
15	15,5942	0,0414	229,19	67,254	18,128	11	0	
<b>Promedio</b>	35,06584	0,093046667	184,2450667	151,3348	36,8574	24,06666667		
<b>Desviación Estándar</b>	12,0603	0,03205	56,7606	60,1663	6,6424	7,36271	<b>0,3333</b>	

<b>Coefficiente de Variación</b>	0,3439	0,3445	0,3080	0,3975	0,18022	0,3059	
<b>Media Error Cruces</b>							

Realizado por: Adriano Erica, 2018

Una vez realizadas las pruebas y en base a la información en la *Tabla 4-3*, nos indica que los datos emanados en forma automatizada son confiables y pueden ser tomados como referencia, con una tasa de error de 0,333 en los cruces debido a la sensibilidad de los sensores, la cual es baja en comparación a la tasa de error en forma tradicional que es de 0.6.

### 3.4.3. Comparación de las variables del sistema automatizado con el sistema tradicional

Se procede a realizar la comparación de las variables medidas cuales son: tiempos de permanencia, número de cruces, con las mediciones tradicionales, se debe considerar que las variables de distancia total, velocidad media son exclusivas del sistema automatizado, en la *Tabla 5-3* se puede observar la comparación de estas variables con su respectiva valoración.

**Tabla 5-3:** Comparación de Variables. Forma automatizada vs Forma Tradicional.

<b>VARIABLES</b>	<b>SISTEMA AUTOMATIZADO</b>		<b>SISTEMA MANUAL</b>
	<b>LDM</b>	<b>EPM</b>	<b>FORMA TRADICIONAL</b>
<b>Distancia</b>	SI	SI	NO
<b>Velocidad Media</b>	SI	SI	NO
<b>Tiempos de Permanencia</b>	SI	SI	SI
<b>Número de Cruces</b>	SI	SI	SI
<b>Exportación de Resultados</b>	SI	SI	NO
<b>Optimización de Tiempo</b>	SI	SI	NO
<b>Margen de Error</b>	BAJO	BAJO	ALTO

Realizado por: Adriano Erica, 2018

En base a la *Tabla 5-3*, se puede apreciar que el sistema automatizado ofrece mayores beneficios en comparación a la forma tradicional de ejecutar las pruebas de actividad ansiolítica. El único error que presentan estos sistemas está en el número de cruces, esto se debe a que los sensores al ser sensibles a la reflexión del objeto toman como cruce el momento en que el roedor acerque su

cabeza al límite existen entre la parte clara y oscura, tomando en cuenta que un cruce es válido cuando el roedor ha cruzado todo su cuerpo, la cola se desprecia. Del mismo al poder exportar los resultados obtenidos esto ahorra tiempo a los expertos para poder utilizarlos en sus respectivos análisis.

### 3.5. Presupuesto

En la *Tabla 6-3* se detalla el presupuesto de costos invertido para los sistemas de monitoreo, refiriéndose tanto a hardware como a software.

**Tabla 6-3:** Presupuesto Sistemas de Monitoreo.

	Dispositivo	Cant.	P.Unitario	P.Total
<b>Hardware</b>	Tarjeta Arduino Uno	3	\$ 35,00	\$ 105,00
	Tarjeta Arduino Mega2560	1	\$ 20,00	\$ 20,00
	Módulo NRF24L01	4	\$ 20,00	\$ 80,00
	Sensor Sharp Gp2y0a21yk0f	6	\$ 17,00	\$ 102,00
	Sensor Sharp Gp2y0a60szlf	4	\$ 20,00	\$ 80,00
	Placas PCB	4	\$ 25,00	\$ 100,00
	Cargador para Arduino	2	\$ 20,00	\$ 40,00
	Plancha de acrílico	1	\$ 130,00	\$ 130,00
	Cajas de protección	4	\$ 10,00	\$ 40,00
	Soporte para sensores	2	\$ 1,00	\$ 2,00
<b>Software</b>	Matlab Home Suit	1	\$ 61,60	\$ 61,60
<b>Otros</b>	Barras de Silicona	3	\$ 0,15	\$ 0,45
	Cloroformo (oz)	5	\$ 4,00	\$ 20,00
	Cable de conexión JST	6	\$ 3,50	\$ 21,00
	Cable de conexión de 4 hilos (m)	6	\$ 2,00	\$ 12,00
	Spray Negro	1	\$ 3,50	\$ 3,50
	Patas de aluminio	5	\$ 3,00	\$ 15,00
	Tapones para aluminio	10	\$ 1,50	\$ 15,00
	Estaño (m)	3	\$ 0,75	\$ 2,25
	Pegamento	8	\$ 0,25	\$ 2,00
<b>Total:</b>				\$ 851,80

Realizado por: Adriano Erica, 2018.

De acuerdo con los valores obtenidos en la *Tabla 6-3* de los costos de producción, el sistema implementado tiene un costo de 851,80 dólares americanos (USD), que en relación a las utilidades que brinda, las características operativas del sistema y la ayuda que brinda para evaluar la actividad ansiolítica, llega a tener un bajo costo.

## CONCLUSIONES

1. Se realizó la implementación de dos sistemas de monitoreo de dispositivos electrónicos de bajo costo cuya operatividad está basada en cuatro bloques: entrada, procesamiento, almacenamiento y usuario, que en conjunto a más de ahorrar recursos optimizan el tiempo de desarrollo de cada prueba, permitiendo al usuario optimar su tiempo para poder dedicarse exclusivamente a la validación de nuevos extractos naturales para ansiolíticos que procuren mejorar la vida de los seres humanos.
2. Se optó por el uso de los sensores Sharp GP2Y0A60SZLF y GP2Y0A21YK0F, por ser elementos asequibles y de tamaño reducido que los hacen ideales para la ejecución y la portabilidad del sistema. De las pruebas se determinó que el sensor Sharp GP2Y0A21YK0F tiene una tasa de actualización entre medidas de 32 ms, por su parte el sensor Sharp GP2Y0A60SZLF posee una alta tasa de actualización de 16 ms, haciéndolo más eficiente en la toma de muestras del sistema.
3. Arduino Uno es la principal plataforma de comunicación entre el sistema y la aplicación, cumple con todas las características fundamentales para la implantación del sistema además de adaptarse a la placa por su uso de espacio reducido. La sincronización entre las dos plataformas (Arduino y Matlab) facilita sin gran cantidad de esfuerzo el procesamiento de las señales emitidas, para poder generar los resultados deseados, el tiempo de sincronización fue de 2 segundos.
4. Del banco de pruebas realizadas en el Modelo Claro/Oscuro se determinó que existe una variación de 0,467 en el conteo de los cruces en relación con la prueba de forma tradicional que fue de 0,600, lo cual nos indica que el sistema implementado emite resultados que pueden ser tomados como referencia.
5. De las pruebas elaboradas en el Laberinto en Cruz Elevado se comprobó que existe una variación de 0,333 en el conteo de cruces en comparación a la prueba de forma tradicional que fue de 0,600, por lo tanto el sistema implementado emite resultados que pueden ser tomados como referencia.

## RECOMENDACIONES

1. Es recomendable que antes de empezar a utilizar el sistema se calibre los sensores, es decir que se ejecute la aplicación sin la existencia de algún objeto dentro del mismo, esto con el fin de procurar no emitir medidas erróneas.
2. Para ampliar el rango de visión de los sensores se puede hacer uso de un servomotor para mapear toda el área consiguiendo la reducción de los elementos a utilizar.
3. Con el fin de mejorar las funcionalidades de los sistemas de monitoreo implementados se sugiere hacer uso de visión artificial para llevar a cabo el procesamiento de imagen, con la ayuda de un software especializado como Matlab o algún otro existente en el mercado como es el caso de Aphelion™.
4. Para futuras mejoras del sistema de monitoreo ahorrando recursos se podría hacer uso de una sola plataforma de Hardware con mayor potencia, procesamiento, capacidad de almacenamiento y transmisión para que no exista interrupciones al momento de ser almacenadas las distancias medidas, como es el caso de Raspberry Pi el cual es compatible incluso con varios sistemas operativos.

## BIBLIOGRAFÍA

**ANTUÑEZ.** *Pinguino*. [En línea].Junta de Extremadura: Proyecto Pingüino. [Consulta: 07 Octubre 2017]. Disponible en: <http://cprjaraiz.juntaextremadura.net/aaa/caparra/17%20CPR%20CACERES%20proyecto%20Pinguino/Articulo.doc>.

**BARRERA, O; & ROS, J.** *Sistemas eléctricos y de seguridad y confortabilidad: Sensores y actuadores*, España: Paraninfo, 2016, p.204.

**BUSTAMANTE ZAPATA, Wilson.** *Ecuador ya cuenta con un biomódulo para investigaciones biomédicas*. [En línea]. Ecuador: Luis Rivadeneira Játiva, Diego Fernando Zapata Garzón, 2014. [Consulta: 17 Julio 2017]. Disponible en: <http://ecuadoruniversitario.com/noticias-universitarias/ecuador-ya-cuenta-con-un-biomodulo-para-investigaciones-biomedicas/>

**CHERREZ ESCOBAR, Geovanni Daniel.** Diseño e implementación de un sistema de localización, medición de velocidad y aceleración de un vehículo para determinar rutas alternas, utilizando tecnologías GPS Y GPRS. [En línea]. (Tesis) (Pregrado) Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, 2017, pp.20, 44, 55. [Consulta: 07 Octubre 2017]. Disponible en: <http://dspace.espace.edu.ec/handle/123456789/3556>

**DYREFACILITETER VED INSTITUT FOR BIOMEDICIN.** [En línea]. Dinamarca: Aarhus Universitet. Open field, 2017. [Consulta: 30 Enero 2017]. Disponible en: <http://dyrefaciliteter.au.dk/institut-for-biomedicin/phenotyping/behavioural-phenotyping/>

**FERNÁNDEZ, Hernando.** *Open Hardware*. [En línea]. Universidad Católica., 2014 [Consulta: 07 octubre 2017]. Disponible en: [http://jeuazarru.com/wp-content/uploads/2014/10/Open\\_Hardware.pdf](http://jeuazarru.com/wp-content/uploads/2014/10/Open_Hardware.pdf)

**FERNÁNDEZ, Roberto; et al.** *Redes inalámbricas de sensores: teoría y aplicación práctica*. [En línea]. España: Universidad de la Rioja, 2009, p.28. [Consulta: 06 marzo 2018]. Disponible en: <https://publicaciones.unirioja.es/catalogo/monografias/mdi26.shtml>

**FUENTES, M.** *Sensores y Transductores*. [En línea]. Chile: Universidad de Tarapacá. Clasificación de los sensores según la magnitud a medir, p.5 [Consulta: 06 octubre 2017]. Disponible en: [http://www.eudim.uta.cl/files/5813/2069/8949/fm\\_Ch03\\_mfuentesm.pdf](http://www.eudim.uta.cl/files/5813/2069/8949/fm_Ch03_mfuentesm.pdf)

**HAU, J.** *Animal models. Handbook of Laboratory Animal Science*. EEUU-CRC Press: Hau, J. & VanHoosier G.L (eds.), 2003, pp. 8-16.

**IDROBO BERMEO, Tatiana Paola.** Evaluación ansiolítica comparativa por solventes de los extractos de hojas flores de *Passiflora manicata* y *Passiflora tripartita* mediante administración vía oral en ratones *Mus musculus*. [En línea]. (Tesis) (Pregrado) Escuela Superior Politécnica de Chimborazo, Facultad de Ciencias, Escuela de Bioquímica y Farmacia, Riobamba, Ecuador, 2016, pp. 29-31, 38. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/5720>

**JEFFUS, Larry.** *Welding Principle and Applications*. 7ª ed. New York-US: Delmar Celgage Learning, 2011, p. 908.

**JÜRGENS, Björn; et al.** *Vigilancia Tecnológica Estudio Sectorial*. [En línea]. Andalucía-España: Agencia de Innovación y Desarrollo de Andalucía IDEA, 2008. pp. 20-23. [Consulta: 08 marzo 2018]. Disponible en: [https://www.agenciaidea.es/c/document\\_library/get\\_file?uuid=9e84835b-e749-4b6c-a01d-f60d3399fae9&groupId=10157](https://www.agenciaidea.es/c/document_library/get_file?uuid=9e84835b-e749-4b6c-a01d-f60d3399fae9&groupId=10157)

**LABEQUIM.** : *Laberinto en Cruz Elevado* [En línea]. México. [Consulta: 20 Diciembre 2017]. Disponible en: [http://labequim.com.mx/html/STOELTING/html/Laberinto\\_en\\_cruz\\_elevado.htm](http://labequim.com.mx/html/STOELTING/html/Laberinto_en_cruz_elevado.htm)



**LLAMAS, Luis.** *Medir distancias con Arduino y sensor Sharp GP2Y0A02YK0F.* [En línea]. Zaragoza-España: ¿Qué es un sensor Sharp gp2y0a02yk0f?, 2016. [Consulta: 08 Marzo 2018]. Disponible en: <https://www.luisllamas.es/arduino-sharp-gp2y0a02yk0f/>

**LORENZO, P; et al.** *Velázquez: Farmacología Básica y Clínica.* 18th ed. Madrid-España: Panamericana, 2015, p.275.

**MATURAMA, Jesús.** *Intel Galileo, placa de desarrollo fruto de la colaboración entre Intel y Arduino.* [En línea]. Xataka: ¿Qué es Intel Galileo?, 2013. [Consulta: 17 Febrero 2018]. Disponible en: <https://www.xataka.com/makers/intel-galileo-placa-de-desarrollo-fruto-de-la-colaboracion-entre-intel-y-arduino>

**MEJÍA, Alfredo.** *Ecuador, entre los países con más casos de depresión en Latinoamérica.* [En línea]. Ecuador: Redacción Médica, 2017. [Consulta: 02 Marzo 2018]. Disponible en: [https://www.redaccionmedica.ec/secciones/salud-publica/ecuador-entre-los-paises-con-m-s-casos-de-depresi-n-en-latinoam-rica-89705](https://www.redaccionmedica.ec/secciones/salud-publica/ecuador-entre-los-paises-con-mas-casos-de-depresi-n-en-latinoam-rica-89705)

**MIRANDA, David.** “Modelado del Comportamiento de Ratas en Laberinto en Cruz Elevado Basado en Redes Neuronales Artificiales”. *Revista de la Sociedad Colombiana de Física*, ISSN 0120-2650, vol. 41 n°2 (2009), (Colombia) pp. 406-408.

**MORA GALLEGOS, Andrea; & SALAS CASTILLO, Sofía.** “Modelos animales de miedo y ansiedad: descripciones neuro-conductuales.” *Actualidades en Psicología.* [En línea]. 2014, (Costa Rica), 28(117), p.5. [Consulta: 07 Marzo 2018]. ISSN 2215-3535. Disponible en: <https://revistas.ucr.ac.cr/index.php/actualidades/article/view/14595/16703>

**NAYLAMPMECHATRONICS.** *Tutorial Sensor de Distancia SHARP* [Blog]. Trujillo-Perú: 2016. [Consulta: 08 Marzo 2018]. Disponible en: [http://www.naylampmechatronics.com/blog/55\\_tutorial-sensor-de-distancia-sharp.html](http://www.naylampmechatronics.com/blog/55_tutorial-sensor-de-distancia-sharp.html)

**NFC SUR MOBILE.** *Une technologie sans fils.* [En línea]. Francia: [Consulta: 07 Marzo 2018]. Disponible en: [http://monge.univ-mlv.fr/~dr/XPOSE2010/NFC\\_sur\\_mobile/carac\\_rsf.html](http://monge.univ-mlv.fr/~dr/XPOSE2010/NFC_sur_mobile/carac_rsf.html)

**OPINIÓN.** *UTMACH inauguró Biomódulo – Bioterio* [En línea]. Machala-Ecuador: 2016. [Consulta: 17 Julio 2017]. Disponible en: <https://www.diariopinion.com/local/verArticulo.php?id=938258>

**PARTICLE DOCS.** *Particle Kits Overview: Core Spark.* [En línea]. [Consulta: 20 Noviembre 2017]. Disponible en: <https://docs.particle.io/datasheets/kits-and-accessories/kits/>

**POLANCO, Lady, Vargas-Irwin, Cristina; & Góngora, Manuel.** “Modelos Animales: Una revisión desde tres pruebas utilizadas en Ansiedad”. *Scientific Electronical Library Online*, [En línea], 2011, (Colombia) 18(2), pp.142-146. [Consulta: 04 Marzo. 2018]. ISSN 0121-4381. Disponible en: <http://www.scielo.org.co/pdf/sumps/v18n2/v18n2a11.pdf>

**PROMETEC.** *Los Módulos de Radio NRF2401.* [En línea]. México: Diagrama De Conexión De Un Módulo Nrf2401. [Consulta: 25 Febrero 2018]. Disponible en: <https://www.prometec.net/nrf2401/>

**REYES MENA, Diego Vinicio.** Diseño de la red de sensores inalámbricos mediante la evaluación del desempeño de las plataformas Arduino y Waspote bajo el protocolo de comunicaciones Zigbee para el monitoreo de los embalses de Agoyán y Pisayambo. [En línea]. (Tesis) (Pregrado) Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, 2014. [Consulta: 07 Octubre 2017]. Disponible en: <http://dspace.espoch.edu.ec/handle/123456789/3556>

**RIBADENEIRA, Jefferson** .*Tema 4 Introducción a GSM.* Riobamba: ESPOCH, 2016, pp. 3, 71.

**SALAMANCA, Fabián.** *Electrónica - Desarrollo en Software y Hardware Libre* [blog]. 2016. [Consulta: 25 Enero 2018]. Disponible en: <https://cc-electrognu.blogspot.com/2016/11/pre.html>

**SALAZAR, Jordi.** *Redes Inalámbricas*. [En línea]. Barcelona-España: European Virtual Learning Platform for Electrical and Information Engineering, 2016. [Consulta: 22 Febrero 2018]. Disponible en: <http://hdl.handle.net/2117/100918>

**SENSORES.** *Sensores Infrarrojos*. [En línea]. España: Ministerio de Educación, Cultura y Deporte. [Consulta: 15 Diciembre 2017]. Disponible en: [http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena11/4quincena11\\_contenidos\\_3g.htm](http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena11/4quincena11_contenidos_3g.htm)

**SENSORES & CONTROLADORES.** *Sensores Fotoelectrónicos* [En línea]. México: 2009, pp.4-7. [Consulta: 10 Noviembre 2017]. Disponible en: <http://dominion.com.mx/descargas/sensores-fotoelectricos.pdf>

**SERNA, A., ROS, F; & RICO, J.** *Los Sensores y su Clasificación. En Guía Práctica de Sensores*. España: Creaciones Copyright. 2010, p.3.

**SHANKS, Niall, GREEK, Ray; & GREEK, Jean.** “Are animal models predictive for humans?” *BioMed Central*. [En línea]. 2009, (Estados Unidos) 4, pp.1-20. [Consulta: 17 Julio 2017]. DOI: 10.1186/1747-5341-4-2. Disponible en: <https://pubmed.biomedcentral.com/articles/10.1186/1747-5341-4-2>

**TBEM.** Sensor Sharp de 10-150CM 5V, GP2Y0A60SZ. [En línea]. Puebla-Mexico: Características del Producto. [Consulta: 25 Noviembre 2017]. Disponible en: <https://teslabem.com/index.php/catalog/product/view/id/316/s/sensor-sharp-de-10-150-cm-5v-gp2y0a60sz/>

**VÁZQUEZ, Carmen; & GARCÍA, Ernesto.** *Tema 8: Sensores Ópticos*. [En línea]. Madrid-España: Universidad Carlos III de Madrid, 2015. [Consulta: 06 Octubre 2017]. Disponible en: [http://ocw.uc3m.es/tecnologia-electronica/instrumentacion-electronica-i/material-de-clase-1/tema8\\_MOD.pdf/view](http://ocw.uc3m.es/tecnologia-electronica/instrumentacion-electronica-i/material-de-clase-1/tema8_MOD.pdf/view)

**VASQUEZ, G.** *Sensores, Características Estáticas y Dinámicas*. [En línea]. Cauca-Colombia, Universidad del Cauca, 2012. [Consulta: 05 Octubre 2017] Disponible en: <http://artemisa.unicauca.edu.co/~gavasquez/res/Sensores.pdf>



**GP2Y0A60SZ0F  
GP2Y0A60SZLF**

Distance Measuring Sensor Unit  
Measuring distance : 10 to 150 cm  
Analog output type



■Description

GP2Y0A60SZ0F/GP2Y0A60SZLF is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector), IR-LED (infrared emitting diode) and signal processing circuit.  
The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method.  
This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

■Features

1. Distance measuring sensor is united with PSD, infrared LED and signal processing circuit
2. Distance measuring range : 10 to 150 cm
3. Compact size (22.0 × 8.0 × 7.2mm)
4. Long distance measuring type  
(No external control signal required)
5. Analog output type

■Agency approvals/Compliance

1. Compliant with RoHS directive (2002/95/EC)

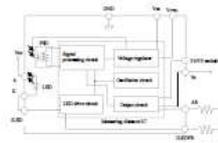
■Applications

1. Touch-less switch  
(Sanitary equipment, Control of illumination, etc)
2. Sensor for energy saving  
(ATM, Copier, Vending machine, Laptop computer, LCD monitor, etc)
3. Amusement equipment  
(Robot, Arcade game machine, etc)

Note: The content of this sheet is subject to change without prior notice.  
In the absence of confirmation for device specifications sheets, SHARP takes no responsibility for the use defects that may occur in equipment using any SHARP device shown in catalogs, data books, etc. Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device.

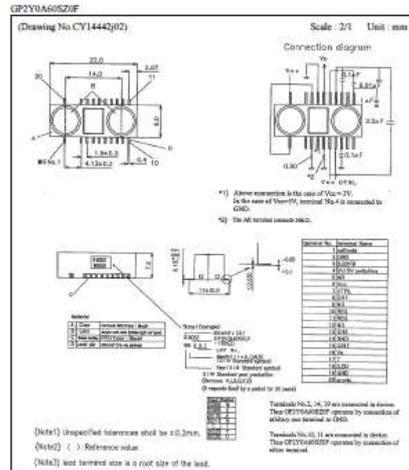
Sheet No.: CP1405501-A

■Schematic



- Showing main terminal
- Please use an electric source with an output current of 400mA or more because LED pulse current is more than 300mA.

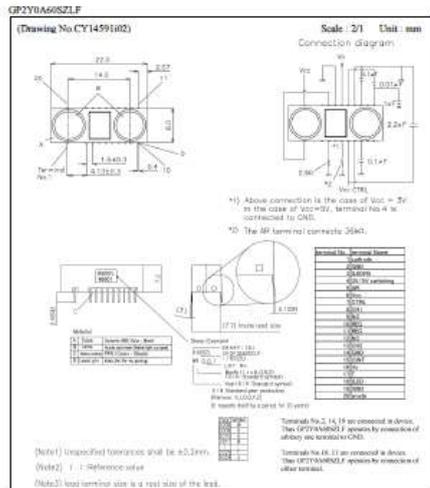
■Outline



Terminal No.2, 14, 19 are connected in device.  
This GP2Y0A60SZLF operates by connection of allbody are connected to GND.  
Terminal No. 10, 11 are connected in device.  
This GP2Y0A60SZLF operates by connection of allbody are connected to GND.

2

Sheet No.: CP1405501-A



3

Sheet No.: CP1405501-A

■Absolute maximum ratings

Parameter	Symbol	Rating	Unit	Remark
Supply voltage	Vcc	-0.3 to +5.5	V	-
Output terminal voltage	V <sub>o</sub>	-0.3 to Vcc+0.3	V	-
Input terminal voltage 1	-	-0.3 to Vcc+0.3	V	Terminal (NC, CH1, CH2, CNT, AR, T)
Input terminal voltage 2	-	-0.3 to Vcc+0.3	V	Terminal (CTRL)
Operating temperature	T <sub>opr</sub>	-10 to +60	°C	-
Storage temperature	T <sub>stg</sub>	-40 to +70	°C	-
Soldering temperature	T <sub>sld</sub>	260	°C	5s or less/time up 2 times

(T<sub>a</sub>=25°C)

■Recommended operating conditions

Parameter	Symbol	Rating	Unit	Remark
Supply voltage	Vcc	4.5 to 5.5	V	5V
Supply voltage	Vcc	3.7 to 3.6	V	3V
High level input voltage	V <sub>in,H</sub>	3.3 to Vcc	V	Openning
Low level input voltage	V <sub>in,L</sub>	MAX 0.2	V	Stand-by state

■Electro-optical Characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Measuring distance range	ΔL	(Note 1)	10	-	150	cm
Output terminal voltage	V <sub>o</sub>	L=150cm (Note 1)	0.05	0.35	0.65	V
Output voltage difference	ΔV <sub>o</sub>	Output change at L change (10cm → 150cm) (Note 1)	1.35	1.60	1.85	V
Average supply current	I <sub>cc1</sub>	Vcc=3V, V <sub>in,H</sub> =3V	-	33	50	mA
Stand-by supply current	I <sub>cc2</sub>	Vcc=3V, V <sub>in,H</sub> =0V	-	-	5	μA

(T<sub>a</sub>=25°C, Vcc=3V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Measuring distance range	ΔL	(Note 1)	10	-	150	cm
Output terminal voltage	V <sub>o</sub>	L=150cm (Note 1)	0.15	0.65	1.15	V
Output voltage difference	ΔV <sub>o</sub>	Output change at L change (10cm → 150cm) (Note 1)	2.75	3.00	3.25	V
Average supply current	I <sub>cc1</sub>	Vcc=5V, V <sub>in,H</sub> =5V	-	33	50	mA
Stand-by supply current	I <sub>cc2</sub>	Vcc=5V, V <sub>in,H</sub> =0V	-	-	5	μA

(T<sub>a</sub>=25°C, Vcc=5V)

Note 1. Distance to reflective object  
(Note 1) Using reflective object : White paper  
(Made by Kodak Co., Ltd. gray cards R-27 : white face, reflective ratio : 90%)

4

Sheet No.: CP1405501-A

# nRF24L01+

## Single Chip 2.4GHz Transceiver

### Preliminary Product Specification v1.0

#### Key Features

- Worldwide 2.4GHz ISM band operation
- 250kpbs, 1Mbps and 2Mbps on air data rates
- Ultra low power operation
- 11.3mA TX at 0dBm output power
- 13.5mA RX at 2Mbps air data rate
- 900nA in power down
- 25µA in standby-I
- On chip voltage regulator
- 1.9 to 3.6V supply range
- Enhanced ShockBurst™
- Automatic packet handling
- Auto packet transaction handling
- 6 data pipe MultiCeiver™
- Drop-in compatibility with nRF24LD1
- On-air compatible in 250kpbs and 1Mbps with nRF2401A, nRF2402, nRF24E1 and nRF24E2
- Low cost BOM
- ±60ppm 16MHz crystal
- 5V tolerant inputs
- Compact 20-pin 4x4mm QFN package

#### Applications

- Wireless PC Peripherals
- Mouse, keyboards and remotes
- 3-in-1 desktop bundles
- Advanced Media center remote controls
- VoIP headsets
- Game controllers
- Sports watches and sensors
- RF remote controls for consumer electronics
- Home and commercial automation
- Ultra low power sensor networks
- Active RFID
- Asset tracking systems
- Toys

All rights reserved.  
Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.  
March 2008

#### 1.1 Features

Features of the nRF24L01+ include:

- Radio
  - Worldwide 2.4GHz ISM band operation
  - 126 RF channels
  - Common RX and TX interface
  - GFSK modulation
  - 250kpbs, 1 and 2Mbps air data rate
  - 1MHz non-overlapping channel spacing at 1Mbps
  - 2MHz non-overlapping channel spacing at 2Mbps
- Transmitter
  - Programmable output power: 0, -6, -12 or -18dBm
  - 11.3mA at 0dBm output power
- Receiver
  - Fast AGC for improved dynamic range
  - Integrated channel filters
  - 13.5mA at 2Mbps
  - 32dBm sensitivity at 2Mbps
  - 85dBm sensitivity at 1Mbps
  - 94dBm sensitivity at 250kpbs
- RF Synthesizer
  - Fully integrated synthesizer
  - No external loop filter, VCO varactor diode or resonator
  - Accepts low cost ±60ppm 16MHz crystal
- Enhanced ShockBurst™
  - 1 to 32 bytes dynamic payload length
  - Automatic packet handling
  - Auto packet transaction handling
  - 5 data pipe MultiCeiver™ for 1.5 star networks
- Power Management
  - Integrated voltage regulator
  - 1.9 to 3.6V supply range
  - Idle modes with fast start-up times for advanced power management
  - 25µA Standby-I mode, 900nA power down mode
  - Max 1.5ms start-up from power down mode
  - Max 130µs start-up from standby-I mode
- Host Interface
  - 4-pin hardware SPI
  - Max 10Mbps
  - 3 separate 32 bytes TX and RX FIFOs
  - 5V tolerant inputs
- Compact 20-pin 4x4mm QFN package

#### 1.2 Block diagram

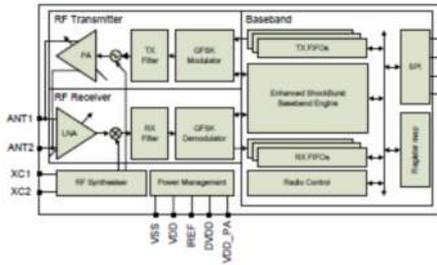


Figure 1. nRF24L01+ block diagram

#### 2 Pin Information

##### 2.1 Pin assignment

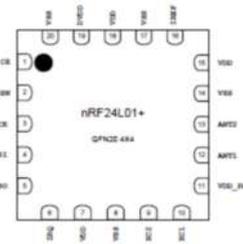


Figure 2. nRF24L01+ pin assignment (top view) for the QFN20 4x4 package

#### 2.2 Pin functions

Pin	Name	Pin function	Description
1	CE	Digital Input	Chip Enable Activates RX or TX mode
2	CAN	Digital Input	SPI Chip Select
3	SCK	Digital Input	SPI Clock
4	MOSI	Digital Input	SPI Slave Data Input, with tri-state option
5	MISO	Digital Output	SPI Slave Data Output, with tri-state option
6	IRQ	Digital Output	Maskable interrupt pin. Active low
7	VDD	Power	Power Supply (+1.9V - +3.6V DC)
8	VSS	Power	Ground (0V)
9	XC2	Analog Output	Crystal pin 2
10	XC1	Analog Input	Crystal pin 1
11	VDD_PA	Power Output	Power Supply Output (+1.9V) for the internal nRF24L01+ Power Amplifier. Must be connected to ANT1 and ANT2 as shown in Fig. 19a 29.
12	ANT1	RF	Antenna interface 1
13	ANT2	RF	Antenna interface 2
14	VSS	Power	Ground (0V)
15	VDD	Power	Power Supply (+1.9V - +3.6V DC)
16	IREF	Analog Input	Reference current. Connect a 22kΩ resistor to ground. See Figure 29.
17	VSS	Power	Ground (0V)
18	VDD	Power	Power Supply (+1.9V - +3.6V DC)
19	DVDD	Power Output	Internal digital supply output for de-coupling purposes. See Figure 29.
20	VSS	Power	Ground (0V)

Table 1. nRF24L01+ pin function

#### 3 Absolute maximum ratings

Note: Exceeding one or more of the limiting values may cause permanent damage to nRF24L01+.

Operating conditions	Minimum	Maximum	Units
<b>Supply voltages</b>			
VDD	-0.3	3.6	V
VSS		0	V
<b>Input voltage</b>			
V <sub>I</sub>	-0.3	5.25	V
<b>Output voltage</b>			
V <sub>O</sub>	VSS to VDD	VSS to VDD	
<b>Total Power Dissipation</b>			
P <sub>D</sub> (T <sub>A</sub> =85°C)		60	mW
<b>Temperatures</b>			
Operating Temperature	-40	+85	°C
Storage Temperature	-40	+125	°C

Table 2. Absolute maximum ratings

5 Electrical specifications

Conditions: VDD = +3V, VAS = 0V, TA = -40°C to +85°C

5.1 Power consumption

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
<b>Idle mode</b>						
I <sub>VDD_PD</sub>	Supply current in power down			900		nA
I <sub>VDD_ST1</sub>	Supply current in standby-I mode	a		26		µA
I <sub>VDD_ST2</sub>	Supply current in standby-II mode			320		µA
I <sub>VDD_SU</sub>	Average current during 1.5ms crystal oscillator startup			400		µA
<b>Transmit</b>						
I <sub>VDD_TX0</sub>	Supply current @ 0dBm output power	b		11.3		mA
I <sub>VDD_TX6</sub>	Supply current @ -6dBm output power	b		9.0		mA
I <sub>VDD_TX12</sub>	Supply current @ -12dBm output power	b		7.5		mA
I <sub>VDD_TX18</sub>	Supply current @ -18dBm output power	b		7.0		mA
I <sub>VDD_AVG</sub>	Average supply current @ -6dBm output power, ShockBurst™	c		0.12		mA
I <sub>VDD_TXS</sub>	Average current during TX setting	d		8.0		mA
<b>Receive</b>						
I <sub>VDD_RX</sub>	Supply current 2Mbps			13.5		mA
I <sub>VDD_RX</sub>	Supply current 1Mbps			13.1		mA
I <sub>VDD_RX</sub>	Supply current 250kpps			12.6		mA
I <sub>VDD_RXS</sub>	Average current during RX setting	e		8.9		mA

- a. This current is for a 12pF crystal. Current when using external clock is dependent on signal swing.
- b. Antenna load impedance = 150+j88Ω.
- c. Antenna load impedance = 150+j88Ω. Average data rate 10kpps and max. payload length packets.
- d. Average current consumption during TX startup (130µs) and when changing mode from RX to TX (130µs).
- e. Average current consumption during RX startup (130µs) and when changing mode from TX to RX (130µs).

Table 4. Power consumption



4 Operating conditions

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
VDD	Supply voltage		1.9	3.0	3.6	V
VDD	Supply voltage if input signals >3.6V		2.7	3.0	3.3	V
TEMP	Operating temperature		-40	+27	+85	°C

Table 3. Operating conditions



5.2 General RF conditions

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
f <sub>op</sub>	Operating frequency	a	2450		2525	MHz
PLL <sub>RES</sub>	PLL programming resolution			1		MHz
f <sub>crystal</sub>	Crystal frequency			16		MHz
Δf <sub>250</sub>	Frequency deviation @ 250kpps			±160		kHz
Δf <sub>1M</sub>	Frequency deviation @ 1Mbps			±160		kHz
Δf <sub>250</sub>	Frequency deviation @ 250kpps			±120		kHz
N <sub>ch</sub>	AF CSFS size	b	250		2000	bits
f <sub>channel, 2M</sub>	Non-overlapping channel spacing @ 250kpps 1Mbps	c			1	MHz
f <sub>channel, 2M</sub>	Non-overlapping channel spacing @ 2Mbps	c			2	MHz

- a. Regulatory standards determine the same range you can use.
- b. Data size in each burst only.
- c. The minimum channel spacing is 1MHz.

Table 5. General RF conditions

5.3 Transmitter operation

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
P <sub>max</sub>	Maximum Output Power	a		14	18	dBm
P <sub>max, RF</sub>	RF Power Control Range			15	20	dB
P <sub>max, RF</sub>	RF Power Accuracy				±4	dB
B <sub>250B</sub>	250B Bandwidth for Modulated Carrier (2Mbps)			1800	2000	kHz
B <sub>1M</sub>	250B Bandwidth for Modulated Carrier (1Mbps)			900	1000	kHz
B <sub>250B</sub>	250B Bandwidth for Modulated Carrier (250kpps)			700	800	kHz
P <sub>EP1.2</sub>	1 <sup>st</sup> Adjacent Channel Transmit Power 2MHz (2Mbps)				-50	dBc
P <sub>EP1.2</sub>	2 <sup>nd</sup> Adjacent Channel Transmit Power 4MHz (2Mbps)				-50	dBc
P <sub>EP1.1</sub>	1 <sup>st</sup> Adjacent Channel Transmit Power 1MHz (1Mbps)				-50	dBc
P <sub>EP1.1</sub>	2 <sup>nd</sup> Adjacent Channel Transmit Power 2MHz (1Mbps)				-45	dBc
P <sub>EP1.20</sub>	1 <sup>st</sup> Adjacent Channel Transmit Power 1MHz (250kpps)				-50	dBc
P <sub>EP1.20</sub>	2 <sup>nd</sup> Adjacent Channel Transmit Power 2MHz (250kpps)				-45	dBc

- a. Antenna load impedance = 150+j88Ω.

Table 6. Transmitter operation



Datarate	Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
2Mbps	C <sub>100</sub>	C/I Co-channel			-11		dBc
	C <sub>115T</sub>	1 <sup>st</sup> ACS C/I 2MHz			-4		dBc
	C <sub>120</sub>	2 <sup>nd</sup> ACS C/I 4MHz			-18		dBc
	C <sub>130</sub>	3 <sup>rd</sup> ACS C/I 6MHz			-24		dBc
	C <sub>140</sub>	N <sup>th</sup> ACS C/I, f = 12MHz			-40		dBc
1Mbps	C <sub>100</sub>	C/I Co-channel			-12		dBc
	C <sub>115T</sub>	1 <sup>st</sup> ACS C/I 1MHz			-8		dBc
	C <sub>120</sub>	2 <sup>nd</sup> ACS C/I 2MHz			-21		dBc
	C <sub>130</sub>	3 <sup>rd</sup> ACS C/I 3MHz			-28		dBc
	C <sub>140</sub>	N <sup>th</sup> ACS C/I, f = 6MHz			-40		dBc
250kpps	C <sub>100</sub>	C/I Co-channel			-7		dBc
	C <sub>115T</sub>	1 <sup>st</sup> ACS C/I 1MHz			-13		dBc
	C <sub>120</sub>	2 <sup>nd</sup> ACS C/I 2MHz			-24		dBc
	C <sub>130</sub>	3 <sup>rd</sup> ACS C/I 3MHz			-28		dBc
	C <sub>140</sub>	N <sup>th</sup> ACS C/I, f = 6MHz			-40		dBc

- a. Narrow Band (in Band) Blocking measurements: 0 to +40dBm, 16kHz step size
- Wide Band Blocking measurements: 30kHz to 2000kHz, 10kHz step size; 2000kHz to 2300kHz, 30kHz step size; 2400kHz to 3000kHz, 30kHz step size; 30MHz to 12.75GHz, 20kHz step size
- Wanted signal for wideband blocking measurements: -47dBm in 1Mbps and 2Mbps mode; -77dBm in 250kpps mode
- For interferer frequency offsets n\*250kHz, blocking performance is degraded by approximately 5dB compared to adjacent figures.
- If the wanted signal is 3dB or more above the sensitivity level then, the carrier-to-interferer ratio is independent of the wanted signal level for a given frequency offset.

Table 8. RX selectivity with nRF24L01+ equal modulation on interfering signal. Measured using P<sub>in</sub> = -47dBm for wanted signal.



5.4 Receiver operation

Datarate	Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
2Mbps	R <sub>X<sub>min</sub></sub>	Maximum received signal at -0.1% BER			0		dBm
	R <sub>X<sub>min</sub></sub>	Sensitivity (0.1%BER) @2Mbps			-82		dBm
1Mbps	R <sub>X<sub>min</sub></sub>	Sensitivity (0.1%BER) @1Mbps			-85		dBm
250kpps	R <sub>X<sub>min</sub></sub>	Sensitivity (0.1%BER) @250kpps			-84		dBm

Table 7. RX Sensitivity

Datarate	Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
2Mbps	C <sub>100</sub>	C/I Co-channel			-7		dBc
	C <sub>115T</sub>	1 <sup>st</sup> ACS (Adjacent Channel Selectivity) C/I 2MHz			-3		dBc
	C <sub>120</sub>	2 <sup>nd</sup> ACS C/I 4MHz			-17		dBc
	C <sub>130</sub>	3 <sup>rd</sup> ACS C/I 6MHz			-21		dBc
	C <sub>140</sub>	N <sup>th</sup> ACS C/I, f = 12MHz			-40		dBc
1Mbps	C <sub>100</sub>	C/I Co-channel			-8		dBc
	C <sub>115T</sub>	1 <sup>st</sup> ACS C/I 1MHz			-6		dBc
	C <sub>120</sub>	2 <sup>nd</sup> ACS C/I 2MHz			-20		dBc
	C <sub>130</sub>	3 <sup>rd</sup> ACS C/I 3MHz			-28		dBc
	C <sub>140</sub>	N <sup>th</sup> ACS C/I, f = 6MHz			-40		dBc
250kpps	C <sub>100</sub>	C/I Co-channel			-8		dBc
	C <sub>115T</sub>	1 <sup>st</sup> ACS C/I 1MHz			-12		dBc
	C <sub>120</sub>	2 <sup>nd</sup> ACS C/I 2MHz			-23		dBc
	C <sub>130</sub>	3 <sup>rd</sup> ACS C/I 3MHz			-28		dBc
	C <sub>140</sub>	N <sup>th</sup> ACS C/I, f = 6MHz			-40		dBc

- a. Narrow Band (in Band) Blocking measurements: 0 to +40dBm, 16kHz step size
- For interferer frequency offsets n\*250kHz, blocking performance is degraded by approximately 5dB compared to adjacent figures.

Table 8. RX selectivity according to ETSI EN 300 440-1 V1.3.1 (2001-06) page 27



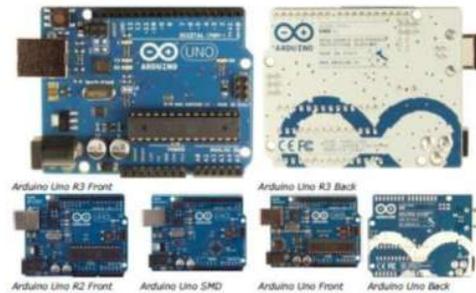
Datarate	Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
2Mbps	P <sub>IM(6)</sub>	Input power of IM interferers at 6 and 12MHz offset from wanted signal			-42		dBm
	P <sub>IM(8)</sub>	Input power of IM interferers at 8 and 16MHz offset from wanted signal			-38		dBm
1Mbps	P <sub>IM(10)</sub>	Input power of IM interferers at 10 and 20MHz offset from wanted signal			-37		dBm
	P <sub>IM(3)</sub>	Input power of IM interferers at 3 and 6MHz offset from wanted signal			-36		dBm
250kpps	P <sub>IM(4)</sub>	Input power of IM interferers at 4 and 8MHz offset from wanted signal			-36		dBm
	P <sub>IM(5)</sub>	Input power of IM interferers at 5 and 10MHz offset from wanted signal			-36		dBm

- Note: Wanted signal level at P<sub>in</sub> = -64 dBm. Two interferers with equal input power are used. The interferer closest in frequency is unmodulated, the other interferer is modulated equal with the wanted signal. The input power of interferers where the sensitivity equals BER = 0.1% is reported.

Table 10. RX intermodulation test performed according to Bluetooth Specification version 2.0



## Arduino Uno



### Overview

The Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega16U2 (ATmega32U2 up to version R3) programmed as a USB-to-serial converter.

Version 1 of the Uno board has a resistor pulling the BU2 HWB line to ground, making it easier to put into [DIP mode](#).

- Version 1 of the board has the following new features:
  - 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the I2CREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
  - Stronger RESET circuit.
  - ATmega 16U2 replace the BU2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

### Summary

Microcontroller ATmega328P  
Operating Voltage 5V  
Input Voltage (recommended) 7-12V

- **Digital I/O:** 14 (of which 6 provide PWM output)
- **Analog Input Pins:** 6
- **DC Current per I/O Pin:** 40 mA
- **DC Current per 3.3V Pin:** 50 mA
- **Flash Memory:** 32 KB (ATmega328P) of which 0.5 KB used by bootloader
- **SRAM:** 2 KB (ATmega328P)
- **EEPROM:** 1 KB (ATmega328P)
- **Clock Speed:** 16 MHz

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **DW:** A4 or A5A pin and A8 or A9A pins. Support PWM generation with the [PCA9685](#) library.

There are a couple of other pins on the board:

- **AREF:** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **RESET:** Bring the line LOW to reset the microcontroller. Typically used to add a reset button to shields which lock the boot on the board.

See also the [pinouts between Arduino boards and ATmega328P boards](#). The mapping for the ATmega328P, 168, and 256 is identical.

### Communication

The Arduino Uno has a number of options for communication with a computer, another Arduino, or other microcontrollers. The ATmega328P provides UART (TX, RX) serial communication, which is available as digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels the serial communication from USB and appears as a virtual COM port to software on the computer. The 16U2 firmware uses the standard USB COM driver, and no external driver is needed. However, [on Windows, a driver file is required](#). The Arduino software includes a serial monitor which allows simple terminal data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transferred via the USB-to-serial chip and USB connection to the computer (but not for serial communication by pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins. The ATmega328P also supports I2C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I2C bus, see the [documentation](#). For details for SPI communication, see the [SPI library](#).

### Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno" from the [Tools > Board menu](#) (according to the microcontroller on your board). For details, see the [reference and tutorials](#).

The ATmega328P on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original 5V/500 protocol ([reference](#), [C header file](#)). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header, see [pin headers for details](#).

- The ATmega16U2 (or 32U2 in the rev 1 and rev 2 boards) firmware source code is available. The ATmega16U2/32U2 is loaded with a DFU bootloader, which can be activated by:
  - On Rev 1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then pressing the BU2.
  - On Rev 2 or later boards: there is a resistor that pulling the BU2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then see [Arduino's DFU software](#) (DFUwin) or the [DFU bootloader](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [Arduino's DFU bootloader](#) for more information.

### Automatic (Software) Reset

Input Voltage (limited)	5-12V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current per 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz

### Schematic & Reference Design

**BAGLE Max** [reference design](#) [Pin 2 to ATmega328P](#) (NOTE: works with Eagle 6.0 and newer)  
Schematic: [Arduino Uno R3 Schematic.pdf](#)  
Notes: The Arduino reference design's main use an ATmega328P, 168, or 256. Current models use an ATmega328P, but the ATmega328P is shown in the schematic for reference. The pin configuration is identical on all these processors.

### Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The regulator can be connected by plugging a 2.1 mm center positive plug into the board's power jack. Leads from a battery can be inserted in the GND and VIN pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

- **VIN:** The input voltage to the Arduino board when it is using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or a regulating voltage via the power jack, access it through a pin header. The pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **5V:** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND:** Ground pin.

### Memory

The ATmega328P has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

### Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Pin 0 (RX) and 1 (TX):** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega328P's USB-to-TTL Serial chip.
- **External Interrupts 0 and 2:** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **Pins 3, 4, 5, 6, 9, 10, and 11:** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

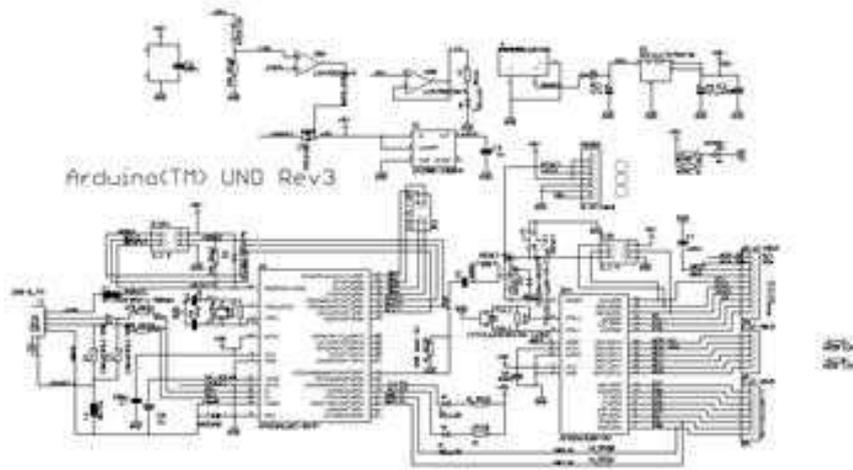
Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega328P/16U2 is connected to the reset line of the ATmega328P via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

### USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

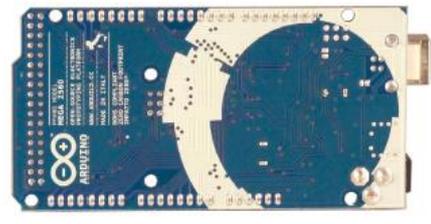
### Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes along the board are attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



Reference Design: WE PROVIDE THE IS\* AND \*WITH ALL RIGHTS RESERVED. ARDUINO INCLUDES ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ARDUINO MAY MAKE CHANGES TO SPECIFICATIONS AND PRODUCT DESIGN/TOPOLOGY AT ANY TIME, WITHOUT NOTICE. THE CUSTOMER MUST NOT RELY ON THE ABSENCE OF CHARACTERISTICS OR WARNINGS OR PRECAUTIONS MARKED "TYPICAL" OR "SUGGESTED" UNLESS SPECIFICALLY STATED IN THE PRODUCT SPECIFICATION AND SHALL HAVE NO RESPONSIBILITY THEREAFTER FOR CONDITIONS OR INCORPORATIONS ARISING FROM FUTURE CHANGES TO THEM. THE PRODUCT INFORMATION ON THE WEB SITE OR PLATFORM IS SUBJECT TO CHANGE WITHOUT NOTICE. DO NOT REPRODUCE A DESIGN WITH THIS INFORMATION. ARDUINO IS A REGISTERED TRADEMARK. Use of the ARDUINO name must be compliant with <https://www.arduino.cc/en/Main/Policy>

### Arduino Mega 2560



The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, a UART (hardware serial port), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

**Schematic & Reference Design**  
 Eagle file: [arduino-mega2560-reference-design.zip](#)

Schematic: [arduino-mega2560-schematic.pdf](#)

**Summary**

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	0-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

**Power**  
 The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.5mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vcc pins headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FT232RL USB-to-serial driver chip. Instead, it features the ATmegaU2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN**: The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, sense it through this pin.
- **5V**: The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **5V2**: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND**: Ground pins.

**Memory**

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

**Input and Output**

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial 0 (RX) and 1 (TX)**; **Serial 11 (RX) and 10 (TX)**; **Serial 17 (RX) and 16 (TX)**; **Serial 3: 15 (RX) and 14 (TX)**. Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega48U2 USB-to-TTL serial chip.
- **External Interrupts 2 (Interrupt 0), 3 (Interrupt 1), 18 (Interrupt 3), 19 (Interrupt 4), 20 (Interrupt 3), and 21 (Interrupt 3)**. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13**. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 20 (MISO), 21 (MOSI), 22 (SCK), 23 (SS)**. These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **LED: 13**. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I<sup>2</sup>C: 40 (SDA) and 41 (SCL)**. Support I<sup>2</sup>C (TWI) communication using the [Wire library](#) (documentation on the [Wiring website](#)). Note that these pins are not in the same location as the I<sup>2</sup>C pins on the Duemilanove or Diecimila.

The Mega2560 has 40 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the [AREF](#) pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF**: Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset**: Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the use on the board.

**Communication**

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega48U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega48U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I<sup>2</sup>C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I<sup>2</sup>C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, see the [SPI library](#).

**Programming**

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preprogrammed with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header file](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega48U2 firmware source code is available in the [Arduino repository](#). The ATmega48U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the MCU. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

**Automatic (Software) Reset**

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega48U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can

have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 100 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

**USB Overcurrent Protection**

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

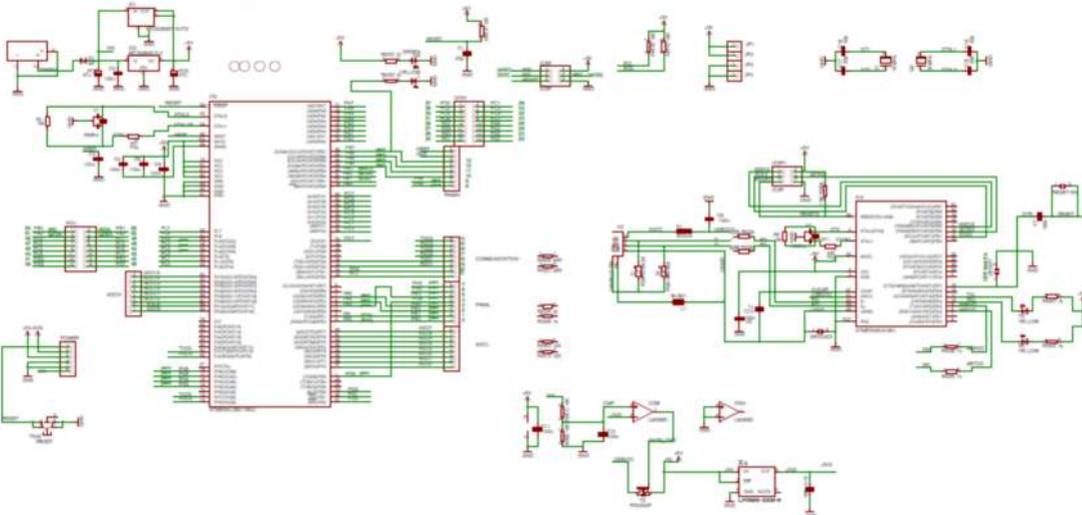
**Physical Characteristics and Shield Compatibility**

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (4.06 mm), not an even multiple of the 100 mil spacing of the other pins.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Duemila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Duemila. *Please note that I<sup>2</sup>C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Duemila (analog inputs 4 and 5).*

**Arduino™ Mega 2560 Reference Design**

Revision: 1.0  
 Date: 2010-05-20  
 Author: Arduino Team  
 License: CC BY-NC-SA



## **Anexo B: Código Arduino**

```
/* -----  
  
    TRABAJO DE TITULACIÓN  
  
    “DISEÑO E IMPLEMENTACIÓN DE DOS SISTEMAS DE MONITOREO PARA LA  
    EVALUACIÓN ANSIOLÍTICA IN VIVO EN RATAS (RATTUS NOVERGICUS)”  
  
    Realizado por: Erica Carolina Adriano Campoverde  
  
-----  
  
*/  
  
// Transmisor Mega  
#include <DistanceGP2Y0A21YK.h>  
#include <nRF24L01.h>  
#include <RF24.h>  
#include <RF24_config.h>  
#include <SPI.h>  
  
int ledon=24;  
int ledtx=22;  
int msg[1] ;           // Array a transmitir  
RF24 radio(9,53);     // Creamos un objeto radio del tipo RF24$  
const uint64_t pipe = 0xE8E8F0F0E1LL; // Usamos este canal  
    double Distancia1,Distancia2,val1,val2,val3.val4;  
void setup(void){  
    Serial.begin(9600);  
    radio.begin();  
    radio.openWritingPipe(pipe); // Abrir para escribir  
    pinMode(ledon, OUTPUT);  
    pinMode(ledtx, OUTPUT);  
    digitalWrite(ledon, HIGH);  
}  
void loop(void){  
  
    int VA0 = analogRead(A0);  
    int VA1 = analogRead(A1);
```

```
int VA2 = analogRead(A2);
int VA3 = analogRead(A3);
Distancia1 = 187754 * pow(VA0, -1.51);
Distancia2 = 187754 * pow(VA1, -1.51);
Distancia3 = 187754 * pow(VA2, -1.51);
Distancia4 = 187754 * pow(VA3, -1.51);
int val1=Distancia1;
int val2=Distancia2;
int val3=Distancia3;
int val4=Distancia4;
```

```
//delay(500); //make it readable
// msg[0] = 100+val1;
val1=val1+100;

radio.write( &val1, sizeof(double) );
val2=val2+300;
radio.write( &val2, sizeof(double) );
val3=val3+400;
radio.write( &val3, sizeof(double) );
val4=val4+500;
radio.write( &val4, sizeof(double) );
digitalWrite(ledtx, HIGH);
delay(10);
digitalWrite(ledtx, LOW);
```

```
Serial.print("Distancia1: ");
Serial.print(val1);
Serial.print(" Distancia2: ");
Serial.println(val2);
```

```

Serial.print("Distancia3: ");
Serial.print(val3);
  Serial.print(" Distancia4: ");
Serial.println(val4);
  // radio.write(msg, 3);
}
/*
Configuración Tx Arduino Uno
*/
// Transmisor Uno
#include <DistanceGP2Y0A21YK.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <RF24_config.h>
#include <SPI.h>
DistanceGP2Y0A21YK Dist;
int ledon=D2;
int ledtx=D3;
int distance;
int msg[1] ;           // Array a transmitir
RF24 radio(9,10);     // Creamos un objeto radio del tipo RF24
const uint64_t pipe = 0xE8E8F0F0E1LL; // Usamos este canal
double
Distancia1,Distancia2,Distancia3,Distancia4,Distancia5,Distancia6,val1,val2,val3,val4,val5,val
6;
void setup(void){
  Serial.begin(9600);
  radio.begin();
  Dist.begin(0);
  radio.openWritingPipe(pipe); // Abrir para escribir
  pinMode(ledon, OUTPUT);
  pinMode(ledtx, OUTPUT);
  digitalWrite(ledon, HIGH);

```

```

}
void loop(void){

int VA0 = analogRead(A0);
int VA1 = analogRead(A1);
int VA2 = analogRead(A2);
int VA3 = analogRead(A3);
int VA4 = analogRead(A4);
int VA5 = analogRead(A5);
Distancia1 = (6762/(VA0 -9))-4;
Distancia2 = (6762/(VA1 -9))-4;
Distancia3 = (6762/(VA2 -9))-4;
Distancia4 = (6762/(VA3 -9))-4;
Distancia5 = (6762/(VA4 -9))-4;
Distancia6 = (6762/(VA5 -9))-4;

//distance = (Dist.getDistanceCentimeter()*2);
//Serial.print("\nDistance in centimeters: ");
//Serial.print(distance);
//delay(500); //make it readable

//msg[0] = distance;
//radio.write(msg, 1);
val1=val1+200;
radio.write( &val1, sizeof(double) );
val2=val2+600;
radio.write( &val2, sizeof(double) );
val3=val3+700;
radio.write( &val3, sizeof(double) );
val4=val4+800;
radio.write( &val4, sizeof(double) );

```

```
val5=val5+900;
radio.write( &val5, sizeof(double) );
val6=val6+950;
radio.write( &val6, sizeof(double) );
```

```
digitalWrite(ledtx, HIGH);
    delay(10);
digitalWrite(ledtx, LOW);
    //Serial.print("Distancia1: ");
Serial.print(val1);
    //Serial.print(" Distancia2: ");
Serial.println(val2);
//Serial.print("Distancia3: ");
Serial.print(val3);
    // Serial.print(" Distancia4: ");
Serial.println(val4);
Serial.println(val5);
Serial.println(val6);
```

```
    //delay(1000);
```

```
}
```

```
/*
```

```
Configuración Rx Arduino Uno
```

```
*/
```

```
// Receptor Mega
```

```
#include <nRF24L01.h>
```

```
#include <RF24.h>
```

```
#include <RF24_config.h>
```

```
#include <SPI.h>
```

```
int dato;
```

```
int msg[1];
```



```
int S1=0;
int S2=0;
int S3=0;
int S4=0;
String trama="";
RF24 radio(9,10);
int ledon=6;
int ledrx=5;

const uint64_t pipe = 0xE8E8F0F0E1LL;

void setup(void){

  Serial.begin(9600);
  radio.begin();
  radio.openReadingPipe(1,pipe);
  radio.startListening();
  pinMode(ledon, OUTPUT);
  pinMode(ledrx, OUTPUT);
  digitalWrite(ledon, HIGH);
}

void loop(void){
  if (radio.available())
  {
    radio.read( &dato, sizeof(int) );
    digitalWrite(ledrx, HIGH);
    delay(10);
    digitalWrite(ledrx, LOW);

    if(dato>100 && dato<=300)
    {
```

```
S1=dato-100;
}

if(dato>300 && dato<=400)
{
S2=dato-300;
}
if(dato>400 && dato<=500)
{
S3=dato-400;
}
if(dato>500)
{
S4=dato-500;
}
trama=S1;
trama+=",";
trama+=S2;
trama+=",";
trama+=S3;

//lastmsg = msg[0];

Serial.print(S1);
Serial.print(",");
Serial.println(S2);
Serial.print(S3);
Serial.print(",");
Serial.println(S4);
//Serial.print(" TRAMA: ");*/

// Serial.println(trama);
```

```

    }
    /*
    Configuración Rx1 Arduino Uno
    */
    // Receptor Uno
    #include <nRF24L01.h>
    #include <RF24.h>
    #include <RF24_config.h>
    #include <SPI.h>
    int dato;
    int msg[1];
    int S1=0;
    int S2=0;
    int S3=0;
    int S4=0;
    int S5=0;
    int S6=0;
    String trama="";
    RF24 radio(9,10);
    int ledon=6;
    int ledrx=5;

    const uint64_t pipe = 0xE8E8F0F0E1LL;

    void setup(void){

        Serial.begin(9600);
        radio.begin();
        radio.openReadingPipe(1,pipe);
        radio.startListening();
        pinMode(ledon, OUTPUT);
        pinMode(ledrx, OUTPUT);
    }

```

```
digitalWrite(ledon, HIGH);
}

void loop(void){
  if (radio.available())
  {
    radio.read( &dato, sizeof(int) );
    digitalWrite(ledrx, HIGH);
    delay(10);
    digitalWrite(ledrx, LOW);

    if(dato>200 && dato<=600)
    {
      S1=dato-200;
    }

    if(dato>600 && dato<=700)
    {
      S2=dato-600;
    }
    if(dato>700 && dato<=800)
    {
      S3=dato-700;
    }
    if(dato>800 && dato<=900)
    {
      S4=dato-800;
    }
    if(dato>900 && dato<=950)
    {
      S5=dato-900;
    }
  }
}
```

```
if(dato>950)
{
S6=dato-950;
}

trama=S1;
trama+=" ";
trama+=S2;
trama+=" ";
trama+=S3;
//lastmsg = msg[0];
Serial.print(S1);
Serial.print(",");
Serial.println(S2);
Serial.print(S3);
Serial.print(",");
Serial.println(S4);
Serial.print(S5);
Serial.print(",");
Serial.println(S6);
//Serial.print(" TRAMA: ");*/

// Serial.println(trama);
}
}
```

### **Anexo C:** Código Guide Matlab R2017a

```
function varargout = claro_oscuro(varargin)

gui_Singleton = 1;

gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @claro_oscuro_OpeningFcn, ...
                  'gui_OutputFcn', @claro_oscuro_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% End initialization code - DO NOT EDIT

% --- Executes just before claro_oscuro is made visible.
function claro_oscuro_OpeningFcn(hObject, eventdata, handles, varargin)

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

% Update handles structure
guidata(hObject, handles);

a=imread('raton-caja.jpg'); %Imagen de fondo
image(a);

axis off

function varargout = claro_oscuro_OutputFcn(hObject, eventdata, handles)

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

function edit1_Callback(hObject, eventdata, handles)

function edit1_CreateFcn(hObject, eventdata, handles)
```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Iniciar.

function Iniciar_Callback(hObject, eventdata, handles)

set(hObject, 'Enable','off');

global v1 v2 v3 v4 v12;

global v5 v6 v11 v13;

global s k v14 v15 v16;

global dt vm tclaro tobscuro contcl_obs contobs_cl;

delete(instrfind({'Port'},{'COM2'}));

s = serial('COM2','BaudRate',9600,'Terminator','CR/LF');

warning('off','MATLAB:serial:fscanf:unsuccessfulRead');

fopen(s);

ts=0.028;

dmax1=0.44;

dmax2=0.29;

dmax3=0.15;

r1=0.32;

disty=0.04;

rh=0.21;

distx=0.055;

r2=0.02;

t=str2double(get(handles.edit1,'string'))*2690; %en un 1 minuto 2690 muestras

v1=zeros(1,t);

v2=zeros(1,t);

v3=zeros(1,t);

v4=zeros(1,t);

v5=zeros(1,t);

v6=zeros(1,t);

for k=1:t

    valor=fscanf(s, '%d,%d');

```

```

valor1=fscanf(s,'%d,%d');
valor2=fscanf(s,'%d,%d');
v1(k)=valor(1);
v2(k)=valor(2);
v3(k)=valor1(1);
v4(k)=valor1(2);
v5(k)=valor2(1);
v6(k)=valor2(2);
end
v11=(30/24).*v1;
v12=(30/25).*v2;
v13=(30/25).*v3;
v14=(30/24).*v4;
v15=(21/24).*v5;
v16=(21/24).*v6;
v=v11;
n=30;
lim1=17;
lim2=32;

y1=limpiar(v,n,lim1+10,lim2);

v=v12;
y2=limpiar(v,n,lim1+10,lim2);

v=v13;
y3=limpiar(v,n,lim1+10,lim2);

v=v14;
y4=limpiar(v,n,lim1+10,lim2);

v=v15;

```



```

y5=limpiar(v,n,lim1,lim2-9);

v=v16;
y6=limpiar(v,n,lim1,lim2-9);
l1=[y1;y2;y3;y4;y5;y6];
l11=l1';
lt=l11/100;
fl=0;
contcl_obs=0;
contobs_cl=0;
[m n]=size(lt);
x=[];
y=[];
for i=1:m
    if sum(lt(i,:))==0
        x=[x x(length(x))];
        y=[y y(length(y))];
    else
        if lt(i,5)~=0||lt(i,6)~=0
            hor=rh*(lt(i,5:6)>0)-lt(i,5:6);
            x=[x sum(hor)/length(find(hor>0))];
            y=[y mean(find(lt(i,5:6)>0)-1)*disty+r1];
        else
            x=[x mean(find(lt(i,1:4)>0)-1)*distx+r2];
            y=[y sum(lt(i,1:4))/length(find(lt(i,1:4)>0))];
        end
    end
end
%plot(x(length(x)),y(length(y)),'*')
%hold on
%axis([0 0.21 0 0.44])
% pause(0.014)
if fl==0 && y(length(y))>0.32 % Conteo de cruces cuarto claro a oscuro

```

```

    contcl_obs=contcl_obs+1;

    fl=1;

end

if fl==1 && y(length(y))<0.30 %Conteo cruces cuarto oscuro a claro

    contobs_cl=contobs_cl+1;

    fl=0;

end

end

dist=[0];

for i=2:length(x)

    dist=[dist sqrt((abs(x(i)-x(i-1)))^2+(abs(y(i)-y(i-1)))^2)];

end

dt=sum(dist);

vel=dist/ts;

vm=mean(vel);

tclaro=ts*sum(y<0.31);

tobscuro=ts*sum(y>0.31);

axes(handles.axes3)

plot(handles.axes3,x,y)

contcl_obs;

contobs_cl;

plot(handles.axes3,x,y) %Gráfica de Trayectoria

set(handles.dis,'string',num2str(dt)); %Impresión de distancia

set(handles.velmed,'string',num2str(vm)); %Impresión de velocidad media

set(handles.ligth,'string',num2str(tclaro)); %Impresión de Tiempo de permanencia claro

set(handles.dark,'string',num2str(tobscuro)); %Tiempo de permanencia oscuro

set(handles.claro,'string',num2str(contcl_obs)); %Conteo cruces claro-oscuro

set(handles.oscuro,'string',num2str(contobs_cl)); %Conteo cruces oscuro-claro

fclose(s); %Cierre del puerto serial

delete(s); %elimina el puerto

clear s;

set(hObject, 'Enable','on');

```

```

% --- Executes on button press in pushbutton2. %Botón Reset
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.dis,'string',' ');
set(handles.edit1,'string',' ');
set(handles.ligth,'string',' ');
set(handles.dark,'string',' ');
set(handles.claro,'string',' ');
set(handles.velmed,'string',' ');
set(handles.oscuro,'string',' ');
cla(handles.axes3,'reset');

% --- Executes on button press in datos. %Botón guardar
function datos_Callback(hObject, eventdata, handles)
% hObject    handle to datos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global dt vm tclaro tobscuro contcl_obs contobs_cl;
A={'Distancia','VelMedia','tiempocl','tiempoOs','cruceClobs','cruceObscl'};
[filename, pathname]=uiputfile('* .xls', 'Choose a file name');
outname=fullfile(pathname, filename);
xlswrite(outname, A,1,'A1');
xlswrite(outname, dt,1,'A2');
xlswrite(outname, vm,1,'B2');
xlswrite(outname, tclaro,1,'C2');
xlswrite(outname, tobscuro,1,'D2');
xlswrite(outname, contcl_obs,1,'E2');
xlswrite(outname, contobs_cl,1,'F2');
rgb= getframe(gca,[-30 -25 510 412]); %guardar grafica
if isempty(rgb)

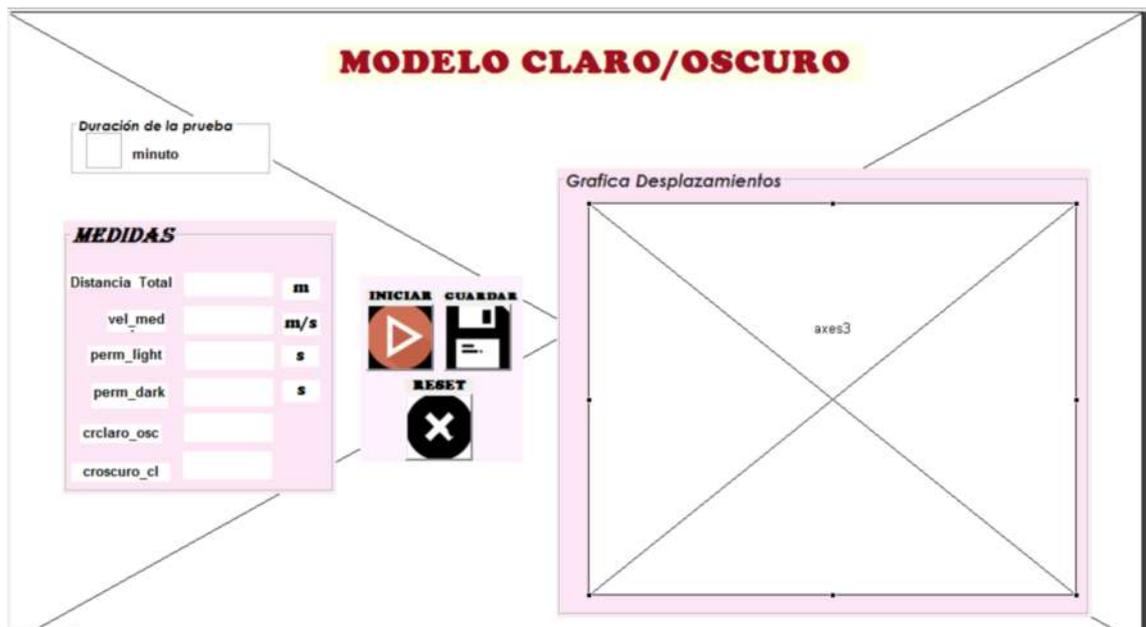
```

```

return
end
%formatos={'*.jpg;*.bmp;*.jpeg;*.fig'};
[nomb,ruta]=uiputfile('*.*','GUARDAR IMAGEN');
if nomb==0
return
end
fname=fullfile(ruta,nomb);
imwrite(rgb.cdata,fname);

```

### Archivo.fig



```

function varargout = Laberinto1(varargin)
% LABERINTO1 MATLAB code for Laberinto1.fig
%   LABERINTO1, by itself, creates a new LABERINTO1 or raises the existing
%   singleton*.
%
%   H = LABERINTO1 returns the handle to a new LABERINTO1 or the handle to
%   the existing singleton*.
%
%   LABERINTO1('CALLBACK', hObject,eventData,handles,...) calls the local
%   function named CALLBACK in LABERINTO1.M with the given input arguments.
%
%   LABERINTO1('Property','Value',...) creates a new LABERINTO1 or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before Laberinto1_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to Laberinto1_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 Laberinto1

% Last Modified by GUIDE v2.5 20-Feb-2018 22:16:54

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @Laberinto1_OpeningFcn, ...
                  'gui_OutputFcn', @Laberinto1_OutputFcn, ...

```

```

        'gui_LayoutFcn', [] , ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

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

% --- Executes just before Laberinto1 is made visible.
function Laberinto1_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 Laberinto1 (see VARARGIN)

% Choose default command line output for Laberinto1
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Laberinto1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
a=imread('lb.jpg');
image(a);

```

```

axis off

global s;

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

function varargout = Laberinto1_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

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

% --- Executes on button press in Iniciar.

function Iniciar_Callback(hObject, eventdata, handles)

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

global k s;

global v1 v2 v11 v12;

global v3 v4 v13 v14;

global sd sv tba tbc tct cba cbc;

delete(instrfind({'Port'}, {'COM2'}));

s = serial('COM2', 'BaudRate', 9600, 'Terminator', 'CR/LF');

warning('off', 'MATLAB:serial:fscanf:unsuccessfulRead');

fopen(s);

ts=0.016;

dmax1=101;

rh=0.43;

dx=0.46;

dy=0.52;

distx=0.06;

r2=0.03;

```

```

t=str2double(get(handles.edit1,'string'))*2690; %en un 1 minuto 2690 muestras
v1=zeros(1,t);
v2=zeros(1,t);
v3=zeros(1,t);
v4=zeros(1,t);
for k=1:t
    valor=fscanf(s, '%d,%d');
    valor1=fscanf(s, '%d,%d');
    v1(k)=valor(1);
    v2(k)=valor(2);
    v3(k)=valor1(1);
    v4(k)=valor1(2);
end
v11=(101/80)*v1;
v12=(101/90)*v2;
v13=(101/55)*v3;
v14=(101/65)*v4;
v=v11;
n=30;
lim1=98;
lim2=105;

y1=non(v,n,lim1,lim2);

v=v12;
y2=non(v,n,lim1,lim2);

v=v13;
y3=non(v,n,lim1,lim2);

v=v14;
y4=non(v,n,lim1,lim2);

```



```

l1=[y1;y2;y3;y4];
l11=l1';
lt=l11/100;
fl1=0;
cba=0;
cbc=0;
[m n]=size(lt);
x=[];
y=[];
for i=1:m
    if sum(lt(i,:))==0
        x=[x x(length(x))];
        y=[y y(length(y))];
    else
        if lt(i,1)~=0 || lt(i,2)~=0
            %hor=rh*(lt(i,1:2)>0)-lt(i,1:2);
            x=[x mean(find(lt(i,1:2)>0)-1)];
            y=[y sum(lt(i,1:2))/length(find(lt(i,1:2)>0)-1)*(dx+dy)/2];

        else
            if lt(i,3)~=0 || lt(i,4)~=0
                x=[x sum(lt(i,3:4))/length(find(lt(i,3:4)>0)-1)*(dx+dy)/2];
                y=[y mean(find(lt(i,3:4)>0)-1)];
            end
        end
    end
end
% plot(x(length(x)),y(length(y)),'*')
% hold on
% axis([0 0.99 0 0.99])
if fl1==0 && (x(length(x))>0.65 || (x(length(x))<0.45))
    cbc=cbc+1;
end

```

```

        fl1=1;
    else
    if fl1==1 && (y(length(y))<0.45 ||y(length(y))>0.75)
        cba=cba+1;
        fl1=2;
    else
    if fl1==2 && ((y(length(y))>0.45 || y(length(y))<0.6)
        contct=contct+1;
        fl1=0;
    end
    end
    end
end
sd=[0];
for i=2:length(x)
sd=[sd sqrt((abs(x(i)-x(i-1)))^2+(abs(y(i)-y(i-1)))^2)];
end
dt=sum(sd);
sv=sd/ts;
vm=mean(sv);
tba=ts*sum(y<0.43)+ts*sum(y>0.58)
tbc=ts*sum(x<0.43)+ts*sum(x>0.58)
tct=ts*sum(y>0.43)+ts*sum(y<0.58)
axes(handles.axes2)
plot(handles.axes2,x,y)
cbc;
cba;
plot(handles.axes2,x,y)
set(handles.dis,'string',num2str(dt));
set(handles.vel,'String',num2str(vm));
set(handles.ba,'String',num2str(tba));
set(handles.bc,'String',num2str(tbc));

```

```
set(handles.ct,'String',num2str(tct));
set(handles.ca,'String',num2str(cba));
set(handles.cc,'String',num2str(cbc));
fclose(s);
delete(s);
clear s;
guidata(hObject,handles);
```

```
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)
```

```
% 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 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 && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

```
end
```

```
% --- Executes on button press in Reset.
```

```
function Reset_Callback(hObject, eventdata, handles)
```

```
% hObject handle to Reset (see GCBO)
```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

set(handles.dis,'string',' ');
set(handles.vel,'String',' ');
set(handles.ba,'String',' ');
set(handles.bc,'String',' ');
set(handles.ct,'String',' ');
set(handles.ca,'String',' ');
set(handles.cc,'String',' ');
cla(handles.axes2,'reset');

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

global dt vm tba tbc tct cba cbc;
A={'Distancia','VelMedia','tiempoBa','tiempoBc','tiempoCt','cruceBa','cruceBc'};
[filename, pathname]=uiputfile('*.xls', 'Choose a file name');
outname=fullfile(pathname, filename);
xlswrite(outname, A,1,'A1');
xlswrite(outname, dt,1,'A2');
xlswrite(outname, vm,1,'B2');
xlswrite(outname, tba,1,'C2');
xlswrite(outname, tbc,1,'D2');
xlswrite(outname, tct,1,'E2');
xlswrite(outname, cba,1,'F2');
xlswrite(outname, cbc,1,'G2');

rgb= getframe(gca,[-30 -25 510 412]); %guardar grafica
if isempty(rgb)
    return
end

```

```

%formatos={'*.jpg;*.bmp;*.jpeg;*.fig'};
[nomb,ruta]=uiputfile('*.jpeg','GUARDAR IMAGEN');
if nomb==0
    return
end
fname=fullfile(ruta,nomb);
imwrite(rgb.cdata,fname);

```

**Archivo.fig**

