

2013-01

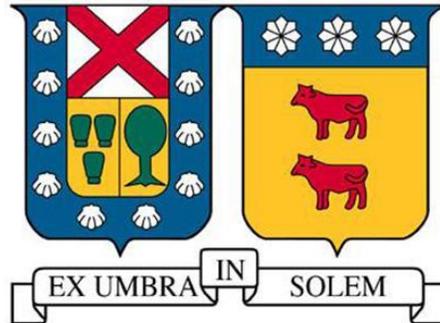
AMBIENTE DE ANALISIS COMPORTAMENTAL PARA ROEDORES DE LABORATORIO

CACERES LECAROS, LUIS ALONSO

<https://hdl.handle.net/11673/49691>

Repositorio Digital USM, UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



«AMBIENTE DE ANÁLISIS
COMPORTAMENTAL PARA ROEDORES DE
LABORATORIO»

LUIS ALONSO CÁCERES LECAROS

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL
ELECTRÓNICO

PROFESOR GUÍA
MARÍA JOSÉ ESCOBAR

PROFESOR CORREFERENTE
DANIEL RODRIGUEZ

PROFESOR CORREFERENTE
ADRIAN PALACIOS

ENERO - 2013

*Dedicado a
mi familia, amigos
y a todos quienes
me acompañaron durante
este recorrido*

Resumen

El desarrollo de esta memoria tiene como objetivo elaborar un ambiente de análisis comportamental para ser utilizado con roedores de laboratorio tipo *Octodon Degus*. Este ambiente de análisis se compone de hardware y software diseñado a medida para presentar estímulos visuales en dos pantallas de tamaño reducido a un roedor que se encuentra confinado a un espacio determinado. Estos estímulos visuales son presentados simultáneamente existiendo un estímulo correcto y otro incorrecto. A disposición del roedor, se encuentran dos palancas, una por cada pantalla para señalar su respuesta y en caso de acertar, se entrega una recompensa en alimento y en caso contrario, se advierte del error con una señal de luz y sonido. Cada sesión de medición presenta una cantidad de estímulos determinada por el usuario. Para cumplir estos objetivos es necesario contar con dos pantallas uOled, dos palancas metálicas, un dispensador de alimento, una cámara web, una tarjeta adquisidora, un computador, entre otros.

Para elaborar el software de control se utiliza LabVIEW, de propiedad de National Instruments. El software de control programado a medida proporciona una interfaz gráfica que permite la configuración de diversos parámetros como por ejemplo, el tiempo de presentación de estímulos, tiempo de recompensa/castigo y más. Además de proporcionar una interfaz gráfica, al finalizar la serie de estímulos presentados se genera un reporte de resultados y un video registro de la sesión correspondiente para posteriores análisis.

Palabras Clave: Análisis comportamental, roedor, Octodón Degus, LabVIEW, Ambiente de análisis

Abstract

The objective of the development of this thesis is to build a behavioral analysis chamber to be used with lab rodents of a specific species, *Octodon Degus*. This chamber is composed by hardware and custom designed software to present visual stimuli through two small screens to one rodent at a time inside this chamber. These visual stimulus are simultaneously presented, showing a correct and an incorrect stimulus. Inside the chamber are two swivels, one for each screen, where the rodent can choose its answer. If it is correct, reward is given but if not, corrective sound and light signal is deployed. Each analysis session presents as many images the user determines. To achieve these objectives specific hardware is needed. Those are two small Oled screens, two metallic levers, a pellet dispenser, a webcam, an acquisition card and a computer among others. The control software is developed with LabVIEW, by National Instruments. This software provides a graphic interface which allows the configuration of different parameters such as stimulus showing time, reward/corrective time and more. As a complementary feature, at the end of a session, a full report and a video recording is given to be used for further analysis.

Key words: Behavioral analysis, rodent, *Octodon Degus*, LabVIEW, analysis chamber.

Índice general

1. Introducción	1
1.1. Inicios del estudio comportamental	1
1.2. Definición de características y objetivos del proyecto	2
2. Consideraciones principales para el desarrollo de proyecto	4
2.1. Estimación de diseño	4
2.1.1. Planteamiento de problemática y solución preliminar	4
2.1.2. Pantalla	7
2.1.3. Dispensador de Alimento	7
2.1.4. Cámara de video	7
2.1.5. Señal auditiva y luminosa	8
2.1.6. Tarjeta adquisidora de datos	8
2.1.7. Computador	8
2.1.8. Software de control	8
2.2. Trabajos previos en el área	9
2.3. Alternativas de Solución	13
2.3.1. Pantalla para imágenes	13
2.3.2. Dispensador de alimento	15
2.3.3. Cámara de video	15
2.3.4. Iluminación de refuerzo/rechazo y señal auditiva	16
2.3.5. Tarjeta programable	16
2.3.6. Software de Control	17
2.4. Solución Escogida	17
2.4.1. Pantalla	17
2.4.2. Cámara	19
2.4.3. Dispensador de Alimento	19
2.4.4. Tarjeta Adquisidora de Datos	23
2.4.5. Herramienta de desarrollo de software	24
2.4.6. Computador	25
2.4.7. Diagrama de Conexiones	25

3. Desarrollo de Software	27
3.1. Panel Frontal	27
3.1.1. Indicadores	27
3.1.2. Controles	31
3.2. Diagrama de Bloques	32
3.2.1. Primer Cuadro	32
3.2.2. Segundo Cuadro	33
3.2.3. Tercer Cuadro	36
3.2.4. Cuarto Cuadro	40
3.2.5. Quinto Cuadro	42
3.2.6. Sexto Cuadro	42
3.2.7. Séptimo Cuadro	52
3.2.8. Octavo Cuadro	52
3.2.9. Noveno Cuadro	53
3.2.10. Fuera de cuadro	53
4. Montaje de dispositivos en ambiente de análisis	56
5. Pruebas	58
5.1. Ejecución de pruebas y evaluación de resultados	59
5.1.1. Prueba 1	59
5.1.2. Prueba 2	60
5.1.3. Prueba 3	60
5.1.4. Prueba 4	61
5.2. Conclusión de pruebas	62
6. Mejoras potenciales al proyecto	63
6.1. Detección de proximidad del roedor a las pantallas	63
6.2. Determinación de nivel de estrés del roedor	65
6.3. Proyecciones futuras para este proyecto	66
7. Conclusiones	68
8. Anexos	70
8.1. Anexo A	70
8.2. Anexo B	76
8.2.1. Primer Cuadro	76
8.2.2. Segundo Cuadro	77
8.2.3. Tercer Cuadro	79
8.2.4. Cuarto Cuadro	81
8.2.5. Quinto Cuadro	85
8.2.6. Sexto Cuadro	86
8.2.7. Séptimo Cuadro	90
8.2.8. Octavo Cuadro	91

8.3.	Anexo C	93
8.4.	Anexo D	95
8.5.	Anexo E	98
8.6.	Anexo F	103

Índice de figuras

2.1. Diseño preliminar de ambiente de análisis	5
2.2. Diagrama de conexión de elementos	9
2.3. Etapas de entrenamiento	10
2.4. Patrón de desplazamiento de herramienta	11
2.5. Cámara de doble elección	12
2.6. Discriminación de patrones visuales	13
2.7. Pantalla uOLED 160-G1, 4D Systems	18
2.8. Cámara LifeCam Studio HD de Microsoft®	19
2.9. Dispensador de pellets	20
2.10. Conector Molex - 3 pines	21
2.11. Circuito relé	21
2.12. Circuito adaptado	22
2.13. Palanca accionadora	23
2.14. Tarjeta adquisidora de datos	23
2.15. Cable blindado y caja de conectores	24
2.16. Diagrama de Conexiones	26
3.1. Panel Frontal - Software de Control	28
3.2. Ventana de diálogo - Bienvenida	32
3.3. Ventana de diálogo - Parámetros Temporales	33
3.4. Ventana de diálogo - Configuración de dispositivos	34
3.5. Diagrama de bloques - Caso 0 (Default)	35
3.6. Diagrama de bloques - Caso 1	35
3.7. Diagrama de bloques - Caso 2	35
3.8. 4DGL Workshop 3	37
3.9. Diagrama de bloques - Inicialización uSD	37
3.10. Interfaz de Graphic Composer 3	39
3.11. Ventana de diálogo - Generación de Secuencia	40
3.12. Ventana de diálogo - Validador de Secuencia	41
3.13. Ventana de diálogo - Ingreso manual de Secuencia	41
3.14. Diagrama de bloques - Estímulos en pantallas	43
3.15. Diagrama de bloques - Censo de palancas a través de entradas digitales NI DAQ	44
3.16. Diagrama de estados - Muestra de 1 par de estímulos	45
3.17. Estado lógico - Listo	46

3.18. Estado lógico - Caso Correcto	47
3.19. Estado lógico - Caso Incorrecto	48
3.20. Estado lógico - Caso Intervalo	49
3.21. Estado lógico - Caso Alimento	50
3.22. Diagrama de bloques - Resguardo de palanca presionada	51
3.23. Gráfico de respuestas acumuladas en panel frontal	51
3.24. Ejemplo de gráfico de respuestas acumuladas para reporte final	51
3.25. Diagrama de bloques - Gráfico respuesta tiempo real	52
3.26. Ventana de diálogo - Fin de sesión y generación de reporte	52
3.27. Diagrama de bloques - Cámara	54
3.28. Diagrama de bloques - Sonido de refuerzo y rechazo	55
4.1. Diagrama diseño ambiente de análisis	56
4.2. Diagrama diseño lateral ambiente de análisis	57
6.1. Detección de proximidad	64
6.2. Vision Assistant	64
8.1. Diagrama de bloques - Primer y Segundo Cuadro y Diagrama de Sonido	71
8.2. Diagrama de bloques - Tercer, Cuarto y Quinto cuadro y Diagrama de Video	72
8.3. Diagrama de bloques - Sexto Cuadro (Incompleto)	73
8.4. Diagrama de bloques - Continuación Sexto Cuadro y Séptimo Cuadro	74
8.5. Diagrama de bloques - Octavo y Noveno Cuadro	75
8.6. Panel frontal - init message.vi	76
8.7. Diagrama de bloques - init message.vi	77
8.8. Diagrama de bloques - temporal parameters.vi	78
8.9. Diagrama de bloques - system configuration .vi	78
8.10. Diagrama de bloques - complete indicator.vi	79
8.11. Diagrama de bloques - image library.vi	80
8.12. Diagrama de bloques - sd init.vi	81
8.13. Diagrama de bloques - stimulus array.vi (caso 0 (Default))	82
8.14. Diagrama de bloques - random.vi	83
8.15. Diagrama de bloques - stimulus array.vi (caso 1)	83
8.16. Diagrama de bloques - stimulus array.vi (caso 2)	84
8.17. Diagrama de bloques - manual.vi	84
8.18. Panel frontal - start count.vi	85
8.19. Diagrama de bloques - start count.vi	85
8.20. Diagrama de bloques - UOled 160G1.vi	86
8.21. Diagrama de bloques - logic module.vi	87
8.22. Diagrama de bloques - pellet dispenser.vi	89
8.23. Diagrama de bloques - screen noise.vi	90
8.24. Diagrama de bloques - report info.vi	91
8.25. Diagrama de bloques - report times.vi	92
8.26. Diagrama de bloques - report resume.vi	93
8.27. Dimensiones físicas módulo pantalla uOLED	94
8.28. Primera página de reporte de sesión	96
8.29. Segunda página de reporte de sesión	97

8.30. Graphic Composer - Pantalla inicial	98
8.31. Graphic Composer - Carga y previsualización de imágenes	99
8.32. Graphic Composer - Guardado de imágenes	99
8.33. Graphic Composer - Opciones de guardado	100
8.34. Graphic Composer - Abrir documento de texto	101
8.35. Graphic Composer - Documento de texto	101
8.36. Graphic Composer - image library.vi	102

Índice de cuadros

5.1. Parámetros de prueba - Prueba 1/2/3	59
5.2. Parámetros de prueba - Prueba 4	61
8.1. Lógica de estado - Correcto	88
8.2. Lógica de estado - Incorrecto	88
8.3. Lógica de estado - Listo	88

Capítulo 1

Introducción

1.1. Inicios del estudio comportamental

El uso de cámaras de condicionamiento operante remontan al inicio del siglo XX, con el pionero Edward L. Thorndike, quien extensivamente utilizó cajas de condicionamiento con gatos. Sus experimentos consistían en confinar un gato dentro de una caja en la cual debían encontrar la forma de escapar, motivados por el aroma de comida que los esperaba fuera de ella. El resultado obtenido demostró un aprendizaje condicionado de los gatos, llevándolos a buscar la forma efectiva de escapar y cada vez que eran sometidos a prueba, lograban el objetivo en menos tiempo. Posteriormente, es Burrhus Frederic Skinner quien formula un estilo de experimentación el cual más tarde dará origen a la conocida «Caja de Skinner». Esta cámara de condicionamiento originalmente usada en palomas, la cual consiste en una caja que contiene estímulos luminosos, una palanca de desición, un dispensador de alimento, entre otros dispositivos operados manualmente, entregó interesantes conclusiones sobre el comportamiento condicionado en función de la obtención de recompensas.

Así es como nace la exploración conductual en animales, dando origen a una interminable variedad de aplicaciones buscando resultados orientados a diversas áreas y es en esta dirección en la cual se desarrolla este proyecto de memoria.

En este proyecto se desarrolla un ambiente de análisis para roedores en el cual pueden interactuar con dispositivos mecánicos como palancas y estímulos visuales en pantallas, siempre

bajo condiciones controladas de iluminación, frecuencia de muestras, cantidad y forma de recompensa. La interacción del roedor examinado en el ambiente de análisis, debe ser grabada para poder ser analizada posteriormente por expertos en comportamiento de roedores, quienes buscan respuestas específicas y patrones de conducta. Para integrar y lograr todo lo mencionado anteriormente, se debe adquirir información de forma efectiva y fidedigna, por lo que es necesario contar con una interfaz gráfica de fácil comprensión y manipulación.

Cabe destacar que este proyecto de memoria se realiza en el contexto de la investigación «*Neurobiology of vision in the retina of the diurnal rodent Octodon Degus: A Psychophysical, Multi-electrode and Computational approach*», llevada a cabo por el Dr. Adrián Palacios, Profesor Referente, y su equipo multidisciplinario de trabajo.

1.2. Definición de características y objetivos del proyecto

El ambiente de análisis que se diseña y construye en este proyecto de memoria, cuenta con características específicas que han sido solicitadas por el grupo encargado de la investigación. Las necesidades básicas que debe cubrir este proyecto se presentan a continuación.

- Presentación de estímulos visuales.

La función medular de este proyecto consiste en presentar estímulos visuales a roedores, quienes deben discernir y escoger cual es el correcto. Esta presentación de estímulos, se llevará a cabo a través de pequeñas pantallas OLED. Se presentarán sets de pares de imágenes controlando tiempo de muestra y tiempo entre muestras, entre otras variables.

- Recompensa y Castigo.

Cada acierto o equivocación del roedor en escoger uno u otro estímulo visual, debe ser seguida por una acción de refuerzo o rechazo. La acción de refuerzo considera dar una recompensa en alimento, mientras que la acción de rechazo, es la negación de tal alimento y la emisión de un sonido y un parpadeo de luz que le indique al roedor que se ha equivocado. Es necesario poder controlar ciertos factores como por ejemplo, la cantidad de alimento, el tiempo que se designa al animal para alimentarse, el tiempo de castigo, entre otros. Esto se determina así dado que para enseñar al roedor a escoger un estímulo sobre otro, se debe tener un entrenamiento gradual. Por ejemplo, partir solo premiando y no castigando,

estimulando el uso de la palanca para generar un vínculo entre lo que el roedor ve en pantalla y lo que puede lograr si presiona la palanca, avanzando en dificultad hasta llegar a la prueba mencionada en un principio. La finalidad de esta interacción es conseguir que el roedor aprenda a discriminar un estímulo de otro a través de ensayo y error.

- Cámara de supervisión y control a tiempo real.

Para supervisar el comportamiento del roedor que está siendo entrenado o analizado, es necesario contar con una cámara que nos muestre a tiempo real como interactúa con los dispositivos a su alcance. Esto permite detectar cualitativamente el progreso del roedor y permite también la intervención del usuario en caso de que el progreso no se lleve a cabo como es esperado. Además de ser utilizada para supervisar al roedor, se debe generar un video que permita guardar un registro de la sesión llevada a cabo para posterior análisis por parte del usuario.

- Generación de reportes.

Al finalizar cada sesión de análisis, es primordial generar un registro de actividades. Este registro se presenta en un reporte que contiene toda la información relevante para el usuario que le permita realizar un estudio posterior. Este informe debe incluir tanto resultados finales como parámetros de entrada y comentarios agregados por el usuario durante la sesión de medición.

- Software de control.

La interfaz de interconexión entre el ambiente físico en el cual se examinan los roedores y el usuario que configura variables y lleva a cabo la sesión de análisis, se implementará a través de un software de control el cual debe contar con una interfaz gráfica intuitiva, amigable y fidedigna.

Entonces, conociendo todas las necesidades de este proyecto y estableciendo sus principales características, podemos definir el objetivo de este proyecto.

El objetivo de este proyecto, es el de establecer conexiones entre dispositivos, de hardware y software, para que estos funcionen en conjunto de forma coordinada y efectiva para permitir entrenar y analizar el comportamiento de roedores del tipo *Octodon Degus* en función de las necesidades definidas previamente.

Capítulo 2

Consideraciones principales para el desarrollo de proyecto

2.1. Estimación de diseño

2.1.1. Planteamiento de problemática y solución preliminar

Como fue mencionado anteriormente, los estudios con roedores de laboratorio han abarcado una amplia gama de investigaciones en diversos ámbitos de la ciencia, como por ejemplo, estudios conductuales, estudios fisiológicos, estudios genéticos, estudios sobre el efecto de drogas y así un número interminable de aplicaciones que han dado frutos creando soluciones a problemas escalables a humanos, como tratamientos de enfermedades, efectos adversos en medicamentos y mucho más.

Cada una de estas investigaciones, utiliza ambientes de análisis con necesidades propias para cada investigación por lo que cada ambiente de estudio es diseñado y creado para cada caso en particular. En este proyecto se diseña y elabora un ambiente de análisis para el estudio [1] a cargo del Dr. Adrián Palacios, External Researcher CINV y profesor de la Universidad de Valparaíso.

La figura 2.1 obtenida de [1], muestra un boceto de un ambiente de análisis para roedores. En esta figura se destaca el uso de un estímulo visual y una palanca, a una distancia establecida,

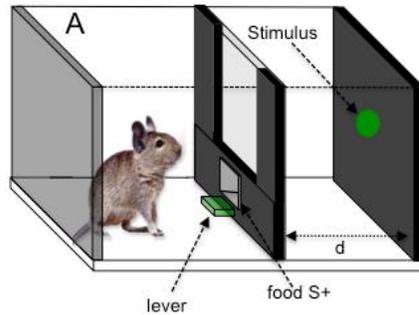


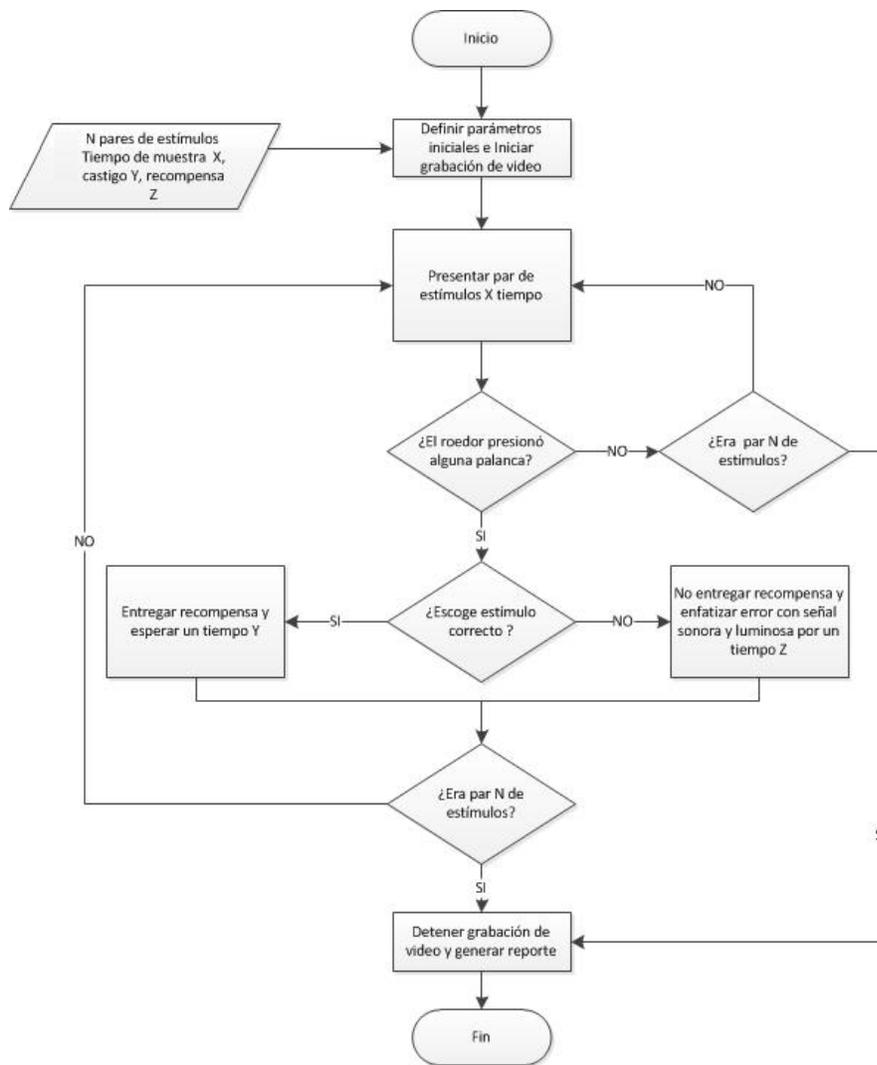
Figura 2.1: Diseño preliminar de ambiente de análisis

para que el roedor pueda interactuar y así obtener resultados orientados a lo que se desee evaluar.

Concretamente, el proyecto a desarrollar poseerá dos pantallas y dos palancas. El flujo de ejecución de tareas para lograr el objetivo de este proyecto será llevado a cabo como se muestra en el diagrama de flujo en la página siguiente. Los parámetros mencionados en este diagrama de flujo son definidos a continuación:

- N par de estímulos: En cada sesión de medición se presentarán N pares de imágenes en las pantallas, de las cuales siempre habrá una asociada a una respuesta correcta y otra a una respuesta incorrecta.
- Tiempo de muestra X: Corresponde al tiempo de presentación en pantalla de cada par de estímulos.
- Tiempo de castigo Y: En el caso que el roedor se equivoque, se emite una señal sonora y luminosa durante un tiempo Y.
- Tiempo de recompensa Z: Corresponde al tiempo otorgado para que el roedor se alimente luego de haber respondido correctamente.

El inicio de una sesión de medición comienza ingresando los parámetros solicitados. Luego, se presenta el primer par de estímulos durante un tiempo X esperando una respuesta del roedor. En caso que no se obtenga respuesta del roedor, se pasa al siguiente par de estímulos. En caso contrario se evalúa la respuesta adquirida y en caso de ser correcta, se premia al roedor y se espera un tiempo Z y en caso contrario, se castiga al roedor en un tiempo Y. Luego, se pasa al siguiente



par de estímulos y así sucesivamente hasta llegar al par N de estímulos con el cual se concluye la sesión. Paralelamente, al inicio de la sesión se comienza a grabar un video que servirá para posterior análisis por parte de los usuarios, que finaliza precisamente con la adquisición de la última respuesta del par N. Además, al culminar la sesión, se genera un reporte que contiene toda la información relevante adquirida así como también los parámetros ingresados.

Ya definido el flujo de acciones, es necesario determinar como y con qué elementos se llevará a cabo el desarrollo de este ambiente de análisis. En las siguientes secciones se determinará que componentes de hardware y software serán necesarios para construir este ambiente.

2.1.2. Pantalla

Este elemento será utilizado para presentar los estímulos visuales al roedor examinado. Los estímulos serán presentados en pares, por lo que se evaluará la opción de utilizar dos pantallas o una de mayor tamaño para presentar dos estímulos a la vez. Los estímulos proyectados serán escogidos para cada fase de la investigación [1] por lo que tendrán diferentes características, tanto como en colores, formas y frecuencia en la que serán proyectados. Por lo tanto, los parámetros de interés para seleccionar una pantalla adecuada para este proyecto son contraste, frecuencia de muestreo, tamaño y complejidad de implementación.

2.1.3. Dispensador de Alimento

Cada vez que el roedor examinado responda correctamente, será recompensado con alimento (pellet). La entrega de este alimento será accionada por el mismo roedor a través de una palanca que estará en frente de cada pantalla, en cambio, si la respuesta es errónea, no se entregará alimento. Las características relevantes de este dispositivo residen principalmente en su velocidad de respuesta y eficiencia, ya que la entrega tardía de la recompensa perjudica el normal desarrollo de la medición, al igual que si esta simplemente no se entrega en la totalidad de las veces que deba hacerse.

2.1.4. Cámara de video

Para obtener un registro de la sesión de medición, es necesario contar con una cámara de video. Además, utilizando esta misma cámara, se necesita obtener una imagen a tiempo real

de lo que ocurre dentro de este ambiente de análisis. Al contar con imágenes a tiempo real es posible determinar por parte del usuario cualquier anomalía que ocurra durante la sesión y así intervenir cuando sea necesario.

2.1.5. Señal auditiva y luminosa

Como medida correctora se emitirá una señal luminosa y auditiva cada vez que el roedor se equivoque. Tanto la señal luminica como la auditiva deben poder ser activadas o desactivadas segun el usuario lo determine.

2.1.6. Tarjeta adquisidora de datos

Para controlar los dispositivos, es necesario contar con una tarjeta adquisidora de datos que debe ser compatible con las especificaciones técnicas de los demás elementos seleccionados. Este elemento es la interfaz entre el ambiente de análisis y el computador que aloja el software de control. La función de esta tarjeta es la de realizar un intercambio de información en ambos sentidos, ambiente y computador, para controlar factores en el ambiente y para recopilar resultados. Se debe buscar que los elementos escogidos sean compatibles con la tarjeta que se utilizará, contando con la cantidad de entradas/salidas necesarias, frecuencia de muestreo suficiente, y todo otro parámetro relevante que permita un funcionamiento en conjunto de la forma mas eficiente.

2.1.7. Computador

La interfaz entre el usuario y el ambiente de análisis es el computador que aloja el software de control. Este computador debe cumplir con las especificaciones técnicas necesarias para asegurar el normal desarrollo de las mediciones.

2.1.8. Software de control

El software de control que se utilizará debe realizar las siguientes funciones como mínimo:

- Visualización de ambiente de análisis a tiempo real
- Configuración de parámetros temporales

- Contabilización de errores y aciertos
- Generación de reporte final

El dilema en este ítem, es la elección de un entorno de programación que sea compatible con las pretenciones que se tienen para este proyecto, escogiendo el adecuado en base a su complejidad de operación y prestaciones.

De todas las consideraciones de hardware y de software realizadas, no se descarta la inclusión de nuevos elementos en base a requerimientos que surjan en el proceso de la investigación.

En la Figura 2.2, podemos ver un esquema básico de la ubicación de los elementos que compondrán este ambiente de análisis.

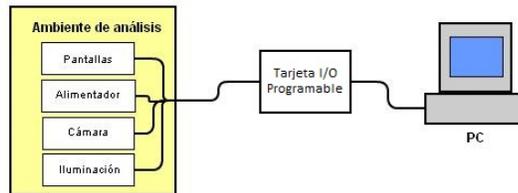


Figura 2.2: Diagrama de conexión de elementos

2.2. Trabajos previos en el área

El ambiente de análisis que se desarrollará, si bien no es un concepto nuevo, es un concepto adaptado que atiende a las necesidades específicas de esta investigación.

Primero, se debe considerar que esta investigación será llevada a cabo con roedores del tipo *Octodon Degus* y una de las interrogantes que puede surgir es: ¿son capaces estos roedores de lograr tales tareas planteadas y aprender a descubrir el estímulo correcto?. Si, lo son. Y para responder a esta interrogante, las habilidades de aprendizaje de estos roedores han sido evaluadas y registradas por investigadores en distintas publicaciones.

Uno de los casos mas interesantes es el trabajo de Tokimoto y Okanoya [4] donde plantea y comprueban la habilidad de estos roedores para aprender a utilizar una herramienta tipo rastrillo para conseguir alimento fuera de su alcance, detrás de una rejilla plastica. Esta tarea

es controlada y registrada variando factores como el largo y forma del rastrillo, y también la ubicación del alimento que debe alcanzar.

En la Figura 2.3, columna derecha, pueden verse los distintos escenarios montados en los que se prueba la capacidad de estos roedores. La secuencia de imágenes en la columna central, muestra la destreza de un roedor en la etapa de entrenamiento 2a, y la columna a la derecha, muestra el desplazamiento de la herramienta en estos instantes.

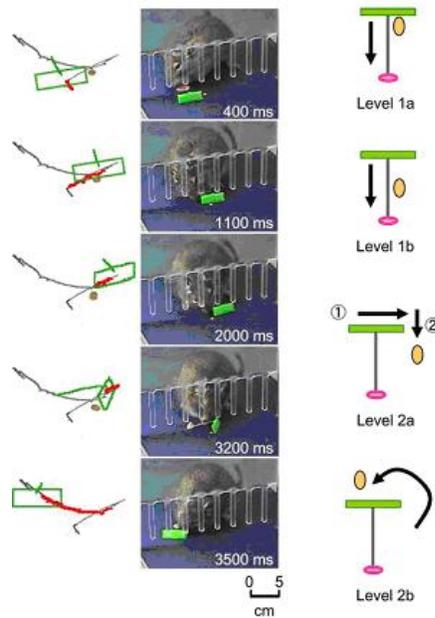


Figura 2.3: Etapas de entrenamiento

También, como se puede ver en la Figura 2.4 se analiza el patrón de desplazamiento de la herramienta a través de seguimiento con una cámara.

El estudio realizado por los autores de [4] permite considerar la adaptación y aprendizaje de este tipo de roedor a los distintos desafíos a los que podría someterse en el desarrollo de este proyecto de memoria.

Otro de los principales estudios, podríamos decir que el más cercano en comparación a la investigación [1], explora las capacidades y características visuales del roedor *Octodon Degus* es el estudio del Dr. Adam Jacobs [2]. Este estudio se concentra en comprender el funcionamiento de la visión de este roedor, llevando a cabo varias pruebas. Entre estas, se encuentran mediciones

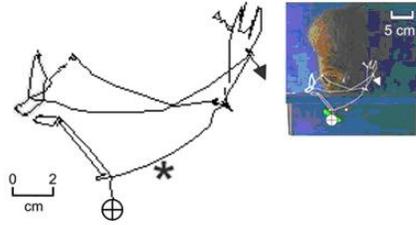


Figura 2.4: Patrón de desplazamiento de herramienta

de absorción lumínica de la cornea, distribución de fotorreceptores sobre la retina y propiedades espectrales de los fotorreceptores.

En relación al trabajo de proyecto de título, en [2], Jacobs realiza un estudio psicofisiológico basado en el comportamiento del animal en un ambiente controlado el cual se presentan tres paneles translúcidos que reciben luz proyectada desde ampollitas y un monocromador, que en conjunto con un espejo es posible direccionar la luz emitida por el monocromador a alguno de los paneles. El roedor es entrenado para discriminar en cual panel se encuentra la luz dirigida desde el monocromador, reforzando las respuestas correctas suministrándole alimento en recompensa. Sobre esta misma configuración de pantallas, se pone a prueba distintas características de la visión del roedor como lo es la visión a diferentes horas del día, controlando la luminosidad del entorno del roedor; la percepción de la intensidad lumínica, modificando la intensidad de las luces proyectadas sobre los paneles translucidos y descubrir el espectro visible de este roedor, realizando un barrido de frecuencia en la luz emitida del monocromador probando hasta que punto el roedor puede acertar en la elección del panel iluminado. El estudio de Jacobs [2] muestra que este tipo de roedor es capaz de reaccionar y aprender sobre estímulos visuales, por lo que el ambiente que se desarrollará, es compatible con las capacidades que este tipo de roedor posee y es de esperar que en base a entrenamiento, después de un tiempo, el roedor pueda responder correctamente al ambiente en el que será inmerso.

El uso de pantallas para desplegar estímulos visuales es recurrente en los estudios con roedores, como por ejemplo, otro estudio de Jacobs [3] en el que se someten a roedores a estímulos visuales con determinados patrones. Los roedores son entrenados para reconocer patrones y luego son puestos a prueba en un ambiente como el que se muestra en la Figura 2.5. La recompensa de esta prueba se encuentra en frente de la pantalla que muestra el patrón visual correcto.

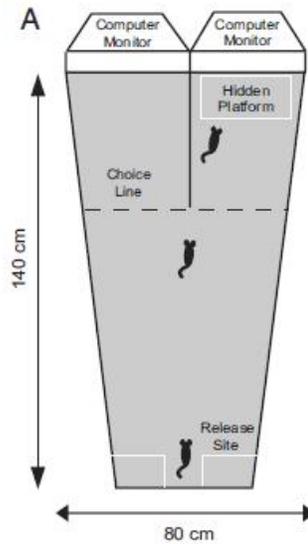


Figura 2.5: Cámara de doble elección

Los autores de [3], utilizaron una pantallas de computador tipo CRT, que son pantallas de mayor tamaño y de menor tecnología en comparación a lo que se busca utilizar en este proyecto de título, pero es una aproximación real en la utilización de pantallas para desplegar estímulos visuales en estudios con roedores.

Otra interesante utilización de estímulos visuales sobre roedores, es llevada a cabo en [5] dirigido por Pamela Reinagel, Phd en bioquímica y profesor asociado en USCD. Parte de la investigación realizada en [5] esta orientada a descubrir el comportamiento neuronal en la codificación de patrones visuales variantes en el tiempo, pero con características constantes. En el caso de la Figura 2.6 se presentan estímulos horizontales y verticales, uno en cada pantalla, esperando que el animal escoga el patrón horizontal, el cual es la característica constante, mientras que la característica variable es el ancho de estos patrones.

En base a todos estos antecedentes, la realización de este proyecto es posible y es cercana en el aspecto técnico a previas investigaciones previas. El tipo de roedor que será analizado en esta investigación, responde de buena forma a estímulos visuales y a elementos con los cuales debe interactuar, en este caso palancas, por lo que se espera obtener un ambiente de análisis funcional y capaz de cubrir las necesidades de esta investigación.



Figura 2.6: Discriminación de patrones visuales

2.3. Alternativas de Solución

Los procesos que se deben llevar a cabo para cumplir los objetivos propuestos, utilizan elementos de hardware y software. Estos elementos presentan características posibles de encontrar en una amplia gama de opciones, por lo tanto, es necesario estudiar estas opciones para posteriormente escoger los elementos adecuados en función de sus características técnicas para llevar a cabo estos procesos.

2.3.1. Pantalla para imágenes

Este elemento será utilizado para desplegar imágenes que serán vistas por el roedor, quien deberá escoger entre dos imágenes y acertar para obtener recompensa. Estas imágenes serán desplegadas en dos pantallas planas o bien, en una pantalla de mayor tamaño. Las imágenes mostradas en las pantallas serán pequeñas, llegando incluso en la etapa final de esta investigación [1] a tener que proyectarse en la retina del roedor en un área no mayor a $480 \times 480 \mu\text{m}$, área que corresponde al mayor tamaño que puede ser examinado en un MEA (Multi Electrode Array) disponible en el laboratorio del Dr. Palacios. Otro aspecto a considerar, es el contraste que ofrecen los diferentes tipos de pantalla y en este caso, lo mejor es utilizar pantallas que ofrezcan el mayor contraste posible para que el roedor centre su atención inmediatamente en los estímulos presentados y no dude o sienta atracción a mirar otras secciones de la pantalla. Los diferentes tipos de pantallas consideradas como alternativas posibles son:

- LCD

- LED

Entre las distintas tecnologías disponibles, la tecnología LED es mejor en ciertas características que la tecnología LCD, por ejemplo, porque tiene mejor calidad de colores, mayor contraste y menor consumo. Por otro lado, las pantallas LCD tienen una vida útil mucho mayor a una pantalla LED, pero este factor es irrelevante, ya que la vida útil de ambas tecnologías supera la expectativa de realización de este proyecto. Dentro de las distintas tecnologías, las pantallas LED, han evolucionado en diversas ramas una de ellas, la tecnología OLED, la cual es muy utilizada actualmente para pantallas de tamaños reducidos. Las pantallas OLED se caracterizan por tener mejor contraste, más brillo, mejores colores y menor consumo que las pantallas LED tradicionales, y como principal característica, las pantallas OLED son flexibles debido a la utilización de partículas plásticas. Aun mejor, es la tecnología AMOLED, que cuenta con mejor eficiencia energética que las pantallas OLED tradicionales, mejores colores y mejor contraste. Estas pantallas son de reducido tamaño por su complejidad y costos de producción.

Todas estas tecnologías en pantallas cuentan con características suficientes para ser utilizadas en el proyecto, por lo que los aspectos determinantes serán su precio en base a su desempeño y disponibilidad.

Entre los fabricantes de este tipo de pantallas, destacamos a *4DSystems*, quien ofrece una amplia gama de pantallas de pequeño tamaño, entre las cuales encontramos los siguientes modelos como posibles alternativas:

- μ LCD-144(GFX)
- μ OLED-96-G1(GFX)
- μ LED-160-G1(GFX)
- μ OLED-32024-P1(GFX)

Todas estas alternativas son suficientes para poder ser utilizadas en este proyecto, ya que son de similares características, variando principalmente en el precio y tamaño. Para mayores detalles de cada pantalla, se recomienda visitar el sitio:

www.4dsystems.com.au/products.php.

2.3.2. Dispensador de alimento

Para dar una recompensa al roedor, es necesario contar con un dispensador de pellets y una palanca, los cuales bien pueden parecer elementos bastante simples, realmente no lo son ya que deben cumplir con ciertos requerimientos.

Durabilidad

Esta palanca será accionada cada vez que sea necesario, pudiendo llegar fácilmente a los cientos de veces y cada vez debe entregar recompensa. La solidez de este elemento debe permitir funcionamiento continuo sin alteraciones. Otro elemento a considerar es la durabilidad de la palanca, ya que es muy probable que el roedor puesto a prueba intente morder la palanca y no sería apropiado que esta se rompa en medio de una medición y haya que intervenir para repararla.

Velocidad de respuesta

Cuando el roedor accione la palanca, la recompensa debe ser entregada lo mas rápido posible y siempre que sea correcto, en virtud de eliminar posibles alteraciones a las mediciones e impedir que los roedores se confundan, perjudicando su aprendizaje.

Las dos alternativas para disponer de este elemento, son adquirir el sistema de un fabricante especializado o fabricar este sistema desde cero. En el caso de adquirir el alimentador y palanca, se asegura funcionamiento efectivo y solo será necesario adaptar este sistema a los demás elementos que componen el proyecto, en cambio, si se fabrica este sistema desde cero, se requiere de mas tiempo y esfuerzo, rescatando si, que probablemente se necesitaría menos recursos que la opción de compra a un fabricante especializado.

Un referente para palacas y dispensadores de pellet para roedores es Med Associates (<http://www.med-associates.com>), que además de estos elementos, ofrecen diversos dispositivos para estudios con todo tipo de animales.

2.3.3. Cámara de video

Tanto para desplegar una imagen a tiempo real en el software de control como para generar un video al final de la sesión de medición, es necesario contar con una cámara que cuente con la

capacidad necesaria para poder desempeñar ambas tareas sin inconvenientes. Efectivamente una cámara web conectada por usb, es apropiada para esta tarea ya que no se necesitan videos de gran calidad de imagen. Solo se debe tomar en cuenta que la cámara escogida debe ser compatible con la herramienta de desarrollo de software que se escoja para implementar el software de control.

2.3.4. Iluminación de refuerzo/rechazo y señal auditiva

En caso de que el roedor elija una opción correcta, se puede reforzar con algún patrón de iluminación y cuando se equivoque, también es posible utilizar variaciones en la iluminación para acentuar el error. Esta señal puede ser acompañada por la emisión de un sonido que advierta y refuerce el acierto u error. En un comienzo se espera que estos cambios de iluminación sean utilizados en la etapa de entrenamiento del roedor. Como dato adicional, se conoce que roedor escogido para esta investigación, Octogon Degus, es un roedor diurno que posee visión dicromática, con peaks de sensibilidad en luz UV y color verde. Es por esta razón, que la luz ambiente incluirá luz UV (300-400nm) y luz verde (510nm). La iluminación será provista a través de tubos fluorescentes o a través de leds de alta intensidad. La señal auditiva será provista por parlantes comunes conectados al computador y situados en las cercanías del ambiente de análisis.

2.3.5. Tarjeta programable

Entre el computador y el ambiente de análisis existe un intercambio de información, por lo que la tarjeta programable corresponde a hardware de adquisición de datos, el cual se encuentra en diferentes formatos, como módulos externos conectados por puerto paralelo, serie y usb. También están disponibles en tarjetas internas para pc, conectadas en puerto pci y pci-e.

Un fabricante reconocido de tarjetas adquisidoras de datos es *National Instruments* quien ofrece una variada gama de tarjetas, internas y externas, y con distintos puertos de conexión. Para conocer mas sobre las tarjetas adquisidoras de *National Instruments*, se recomienda visitar el sitio www.ni.com/dataacquisition/multifunction/.

Es posible también pensar en otras alternativas como FPGA, MSP u otro como por ejemplo Arduino, los cuales corresponden a interfaces de medición y control entre ambientes físicos y digitales, pero que en esta ocasión no serán considerados, ya que porque están orientados a otros usos, especialmente para sistemas embebidos o autónomos que necesiten de procesamiento de

baja/media complejidad sin la necesidad de un computador para ejecutar comandos.

2.3.6. Software de Control

El software de control debe poseer características definidas como se mencionó en la sección anterior, dejando como interrogante la plataforma de programación de este software.

Habiendo considerado a *National Instruments* para escoger una tarjeta adquisidora de datos, la primera y única opción considerada para desarrollar este software de control, es LabVIEW, de propiedad del mismo fabricante, *National Instruments*.

LabVIEW es una herramienta de desarrollo de amplio uso en muchos ámbitos, ya que su simplicidad de manejo, acerca su uso a diversas áreas de la ingeniería. Este software se basa en lenguaje G, o lenguaje gráfico, permitiendo simplificar el proceso de desarrollo al programar con módulos o bloques, evitando tener que programar en código. Los programas creados en LabVIEW se llaman instrumentos virtuales, o VI los cuales pueden ser ejecutados en un computador u operar en sistemas embebidos.

Este entorno de programación posee los módulos, o Toolkits, necesarios para controlar los distintos dispositivos que componen este ambiente de análisis y mucho más. Además es prudente mencionar, que si bien este es un software de pago, se cuenta con una licencia a disposición.

Para mayor información y detalles de LabVIEW es posible ver <http://www.ni.com/labview/>, donde se encuentran especificaciones técnicas y aplicaciones reales para ejemplificar el alcance y poder de aplicación este software.

2.4. Solución Escogida

Para llevar a cabo las distintas funciones para lograr los objetivos propuestos, es necesario contar con el equipamiento necesario. Como se vió en la sección anterior, existen varias alternativas disponibles, de las cuales se escogieron las siguientes:

2.4.1. Pantalla

La pantalla escogida para usar en este proyecto, es una pantalla uOled, modelo 160-G1(SCG), fabricada por *4DSystems*.

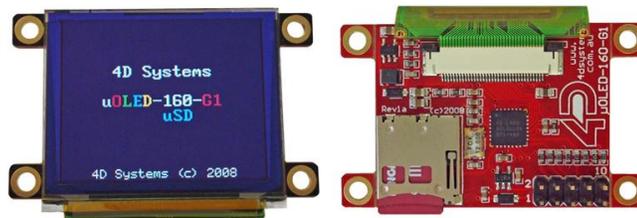


Figura 2.7: Pantalla uOLED 160-G1, 4D Systems

Esta pantalla, Figura 2.7, es de tamaño reducido, versátil y de mediana complejidad, lo que la hace ideal para ser utilizada en este proyecto. Dentro de sus características técnicas, podemos mencionar algunas:

- Tamaño de 1,7 pulgadas.
- Lector de tarjeta microSD con memoria de 2Gb incorporada.
- 65k colores.
- Función de *Auto Baud Rate*, para sincronizar tasas de muestreo entre dispositivo y controlador.
- Conector serial de 5 pines (VCC, TX, RX, GND, RESET).

El conector serial de la pantalla posee un adaptador USB, el cual necesita de un controlador que debe ser instalado en el computador *Host*.

El microcontrolador de esta pantalla posee una característica dual de funcionamiento, que permite operar en modo *Stand Alone* o como *Slave*. Es el segundo modo el escogido para desarrollar este proyecto. En este modo, se utiliza a la pantalla como esclavo y el computador utilizado, como *Host*, quien envía comandos de forma serial en formato de 8 bits, sin paridad y un bit de detención.

Para escoger esta modalidad de operación, el fabricante provee un programa para cargar el controlador específico a la pantalla, llamado *Pmmc Loader*. Además de este programa, el fabricante provee otros programas para cargar imágenes en la tarjeta de memoria, para realizar pruebas básicas para comprender el funcionamiento, para cambiar la fuente de texto y más. Y

en función de prestar soporte completo al usuario, se dispone también de IDE *4GDL Workshop*, que permite programar funciones y realizar proyectos entorno a estas pantallas.

Para conocer más sobre este entorno de desarrollo para este tipo de pantallas, visitar www.4dsystems.com.au/prod.php?id=111. Para obtener más especificaciones técnicas de la pantalla escogida, ver Anexo C.

2.4.2. Cámara

La cámara escogida para ser utilizada en este proyecto es una cámara LifeCam Studio de Microsoft® HD. Esta cámara, Figura 2.8, es *DirectShow Compliant*, es decir, utiliza específicamente este framework multimedia para proporcionar una interfaz apta para ser manipulada por distintos lenguajes de programación. Este aspecto es importante, ya que LabVIEW, el cual es el entorno de desarrollo que se utilizará, trabaja con este framework para manipular imágenes desde cámaras web. Además, esta cámara posee un sensor HD que permite realizar grabaciones y tomar fotografías en alta definición de hasta 1080p y también un micrófono con botón de encendido y apagado.



Figura 2.8: Cámara LifeCam Studio HD de Microsoft®

2.4.3. Dispensador de Alimento

Este elemento entrega recompensa al roedor en caso de que este acierte al estímulo correcto. El dispensador de pellets es adquirido a Med-Associates, quien produce artículos especializados para la investigación con animales, ofreciendo desde palancas, como las que usaremos en es este proyecto, hasta estaciones de mediciones autónomas de mayor complejidad. Este dispositivo

opera con un voltaje de entrada de 28V continuo el cual será proporcionado por una fuente regulable de voltaje. La corriente máxima del dispositivo, no supera los 0,3 mA, valor que ha sido determinado a través de mediciones.

El dispensador de pellets, Figura 2.9, provee durabilidad, rapidez y efectividad. Posee un sensor infrarrojo para verificar que se entrega el pellet de recompensa y su procesador interno permite intentar hasta 5 veces la entrega en caso de falla y tan solo en un periodo de 150 milisegundos. En caso de falla, y luego del quinto intento, emite una señal de falla.



Figura 2.9: Dispensador de pellets

Este dispensador, posee un pulsador en su estructura que permite entregar un pellet en cuanto se necesite y también posee un conector tipo *mini Molex*®, de tres pines, el cual se puede conectar a un interruptor externo y accionar su función. Este conector posee una configuración según se muestra en la Figura 2.10.

Para accionar este alimentador y entregar un pellet de recompensa, es necesario conectar el pin n°2 (blanco) a tierra (negro). Esta acción debe ser realizada a través del software de control, que puede determinar si una respuesta adquirida es correcta o incorrecta. Para ejecutar esta tarea desde la tarjeta adquisidora, se necesita un circuito de adaptación, Figura 2.11.

El relé del circuito se activa con 5[V] de entrada, admitiendo 24[V] cuando el relé cierra. Se incluye un diodo de seguridad para proteger la tarjeta y transistor de un posible peak de voltaje que se produce por el campo magnético generado en la desactivación del relé.

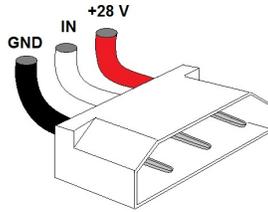


Figura 2.10: Conector Molex - 3 pines

Los componentes seleccionados para construir este circuito son:

- Resistencia de $4.7K\Omega$
- Transistor NPN BC337
- Diodo 1N4007
- Relé JQC 3FL

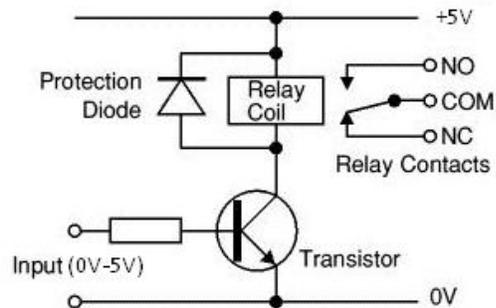


Figura 2.11: Circuito relé

El transistor funciona en corte y saturación. Cuando la salida TTL de la tarjeta de adquisición a la cual está conectado el circuito es de $0[V]$, no hay corriente de colector. Mientras que cuando hay $5[V]$ de entrada a la base del transistor, la corriente desde colector a emisor es aproximadamente $1[mA]$. El tiempo que el transistor se encuentra saturado cuando es utilizado es mínimo, cercano a los 10 milisegundos.

Finalmente, el circuito palanca y dispensador de pellet adaptado queda como se muestra en la Figura 2.12.

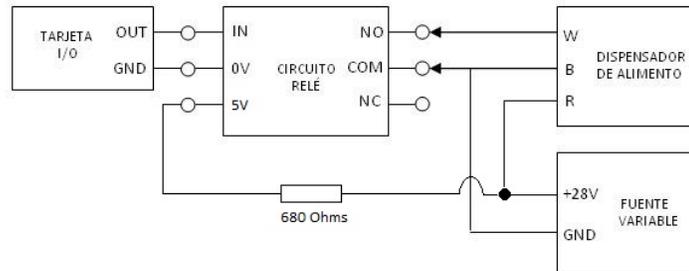


Figura 2.12: Circuito adaptado

La salida de la tarjeta adquisidora provee una fuente de 5[V], pero no es suficiente para alimentar el circuito de relé, por lo que se utiliza la misma fuente variable, con una resistencia de 680 Ohms en serie, que permite que el voltaje de 28[V] que alimenta el dispensador, pase a ser alrededor de 5[V] a la entrada del circuito con la corriente utilizada. Esta resistencia fue determinada luego de obtener un promedio de la corriente entre el pin blanco y el pin negro cuando se acciona el dispensador de pellets. La corriente en promedio medida fue de 30[mA]. La corriente que circula por esta resistencia solo circula durante el tiempo que el transistor se encuentra saturado, cerca de 10 milisegundos, y el resto del tiempo termina en circuito abierto.

Por último, la palanca de acción es una palanca metálica, Figura 2.13, redondeada en el extremo para evitar que el roedor se sienta atraído por morderla. En el extremo posterior posee una resistencia de peso equivalente a 35 gr.

Esta palanca también posee un conector Molex, pero solo utiliza dos de tres pines. La palanca lleva montado un switch del cual solo se utilizan los contactos COM y NO (común y normalmente abierto). Por lo tanto, al momento de presionar la palanca, el contacto NO se cierra. Esta palanca no interactúa con el circuito de adaptación o con el dispensador de pellet directamente, sino que se conecta a la tarjeta adquisidora de datos para ingresar un 1 o 0 indicando si ha sido presionado o no, donde posteriormente se realizan las operaciones necesarias para determinar si ha sido presionada en el momento correcto o no.



Figura 2.13: Palanca accionadora

2.4.4. Tarjeta Adquisidora de Datos

Entre las opciones que se tienen para escoger una tarjeta adquisidora de datos, *National Instruments* es referente mundial. La variedad de tarjetas, internas y externas, es bastante amplia y para variados presupuestos. Una excelente alternativa es la tarjeta adquisidora NI PCI 6221, Figura 2.14.

Como principal característica, destacan las 24 entradas/salidas digitales, dos fuentes de 5 volts y sus respectivas tierras, para trabajar con lógica TTL. Esto es adecuado a las necesidades del proyecto desarrollado, sin contar las demás características que posee esta tarjeta que excede con creces las necesidades básicas de este proyecto. Para conocer mayores detalles técnicos de esta tarjeta, visitar el sitio <http://sine.ni.com/nips/cds/view/p/lang/en/nid/14132>.



Figura 2.14: Tarjeta adquisidora de datos

Además de la tarjeta adquisidora, es necesario contar con un cable blindado a la salida de la tarjeta adquisidora y posteriormente, una caja de conectores, como se muestra en la figura 2.15.



Figura 2.15: Cable blindado y caja de conectores

2.4.5. Herramienta de desarrollo de software

LabVIEW es una plataforma de programación que se basa en programación gráfica. Al contrario de utilizar lenguaje de programación escrito, este estilo de programación permite una elaboración en menor tiempo de proyectos que requieren programar interfaces, dispositivos y más, ya que la implementación de instrucciones se realiza a través de la unión de bloques a través de líneas, encerrando grupos de bloques dentro de estructuras lógicas, declarando variables a través de bloques y más características que permiten una programación eficiente incluso por profesionales que no sean expertos en la materia.

Para este proyecto se necesita contar con LabVIEW, versión Full Profesional y los siguientes módulos de extensión:

- Vision Development Module

Este módulo permite la adquisición de imagen a tiempo real a través de una cámara web conectada a puerto USB del computador que se utiliza en este proyecto. Además de adquirir y registrar imagen y video, permite realizar análisis de diversa complejidad, como por ejemplo, filtrar colores de imágenes, reconocer patrones repetitivos en un video, contar objetos, reconocer objetos por su forma geométrica, medir distancia y mucho más.

- VISA Module Para establecer una comunicación entre las pantallas escogidas para desplegar imágenes y el computador que aloja el VI generado, se necesita establecer una

comunicación serial a través del puerto USB. Como mencionamos en una sección anterior, las pantallas cuentan con una interfaz USB a serial y esta comunicación se concreta enviando comandos en formato hexadecimal y gracias a este módulo, es posible sincronizar y establecer comunicación entre las partes.

- Report Generation Toolkit Este módulo permite generar el reporte final para recopilar la información necesaria para ser analizada posterior al proceso de medición en el ambiente de análisis.

La versatilidad y efectividad de LabVIEW se encuentra mas que demostrada en diversos ámbitos de la ciencia. En el sitio www.ni.com/labview/ es posible conocer aplicaciones reales y de mayor escala, como también sus características técnicas.

2.4.6. Computador

El computador que será utilizado para el desarrollo de esta experiencia posee las siguientes características:

- Pantalla LED de 19 pulgadas, marca LG.
- Procesador Intel Core i7. Cuatro núcleos (3,0 GHz).
- 4Gb Memoria RAM
- Tarjeta gráfica NVIDIA 9600, 1Gb de RAM

Este computador se encuentra disponible en el laboratorio de conducta en la facultad de ciencias de la Universidad de Valparaiso y cuenta con capacidades suficientes para poder ejecutar sin problemas el software de control.

2.4.7. Diagrama de Conexiones

Todos los componentes escogidos para desarrollar este proyecto se relacionan según se puede ver en la Figura 2.16

El flujo de datos es bidireccional, desde un extremo al otro. Desde el extremo derecho, las palancas que actúan como interruptores, llevan a una entrada digital (DI) un pulso de 5 volts al

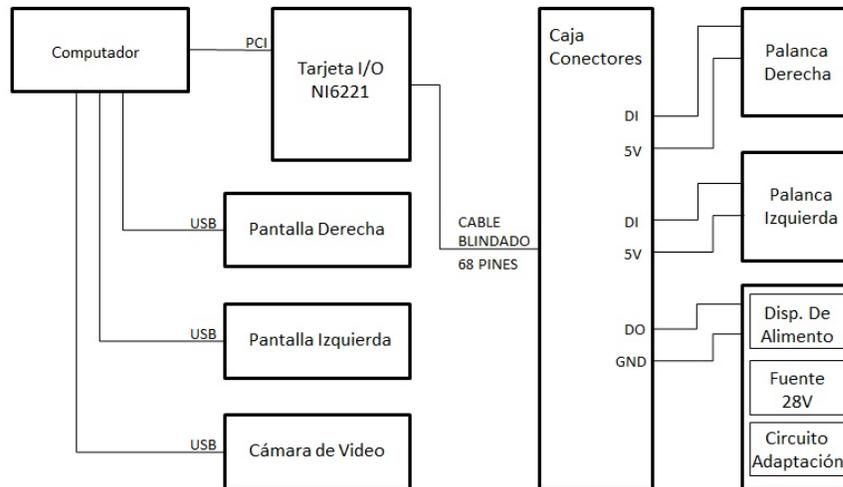


Figura 2.16: Diagrama de Conexiones

ser presionadas. Este pulso pasa desde la caja de conectores hacia la tarjeta adquisidora NI 6221 instalada en el computador en un puerto PCI a través de un cable blindado de 68 pines provisto por el fabricante. La tarjeta adquisidora recibe este pulso asociándolo con un 1 booleano, el cual ingresa al software de control interactuando con la lógica programada. En el caso que la respuesta asociada sea correcta, se envía una señal digital a una línea de salida, la cual realiza el mismo recorrido a través de los componentes. Esta señal es enviada a la línea de salida (DO), la cual es entrada al dispensador de alimento. Concretamente ingresa al circuito de adaptación (Figura 2.12), que adapta la señal a un voltaje apto para interactuar con el dispensador de alimento, como fue previamente explicado en la sección correspondiente.

Los periféricos restantes, pantallas y cámara de video, se encuentran conectados directamente al computador de control a través de puertos USB.

Capítulo 3

Desarrollo de Software

Para controlar y supervisar el ambiente de análisis, es necesario contar con una interfaz para el usuario. Esta interfaz debe contar con todos los elementos necesarios para modificar parámetros de medición y también, elementos para registrar constantemente el correcto desarrollo de la medición. Como fue mencionado anteriormente, el estilo de programación de LabVIEW se denomina programación G, o programación gráfica, que permite unir bloques creando un diagrama de bloques (DB), cuyo funcionamiento programado se refleja en los indicadores en el panel frontal (PF), utilizando constantes y controles de entrada, también algunos de ellos presentes en el PF.

En las siguientes secciones se presentará en detalle el PF y el DB generado para este proyecto.

3.1. Panel Frontal

El PF de la interfaz creada presenta distintos elementos, los cuales pueden ser indicadores o controles, según se puede ver en la Figura 3.1.

A continuación, se define cada uno de estos elementos presentes en el PF.

3.1.1. Indicadores

Los indicadores, como su nombre lo indica, presentan datos durante una medición. Por lo tanto, la interacción con estos elementos se remite a la visualización y supervisión.

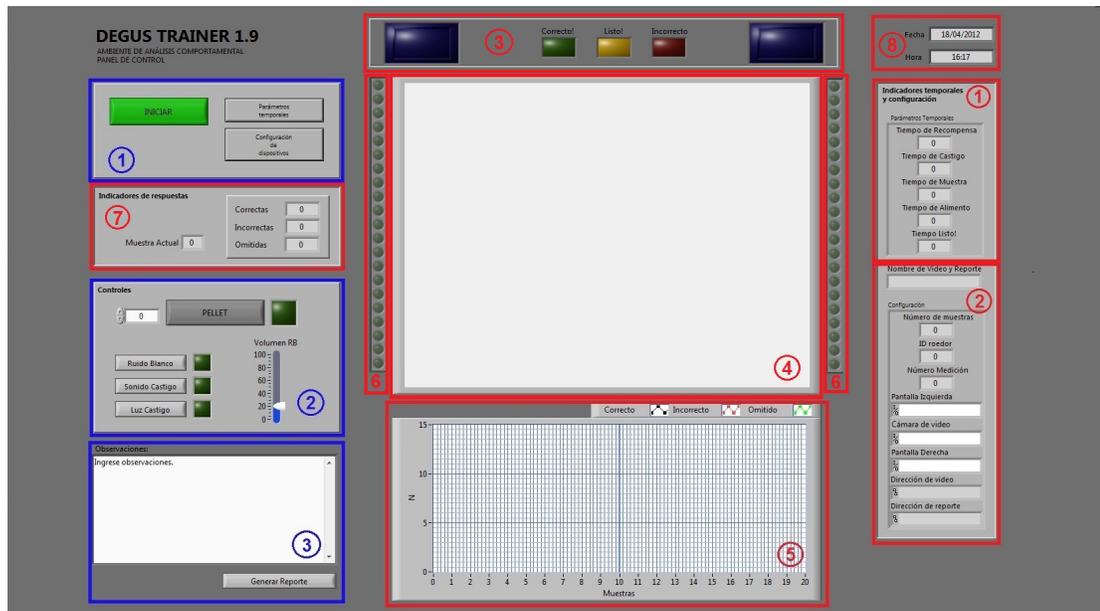


Figura 3.1: Panel Frontal - Software de Control

Según la enumeración propuesta en la Figura 3.1, destacados en rojo se tienen los siguientes INDICADORES:

1. Parámetros temporales

Estos parámetros definen los tiempos de acciones de las pantallas, recompensa y castigo sobre el roedor analizado.

- Tiempo de Muestra: Corresponde al tiempo de presentación de los estímulos en pantalla.
- Tiempo Listo!: Es el tiempo que separa el término de presentación de un par de estímulos del comienzo del par siguiente. Durante este tiempo, se espera que el roedor de su respuesta a través de las palancas disponibles.
- Tiempo de Recompensa: Como su nombre lo dice, es el tiempo que se le da al roedor para recibir y alimentarse de su recompensa al responder correctamente.
- Tiempo de Castigo: Similar al tiempo de recompensa, el tiempo de castigo es un tiempo de separación entre imágenes que se da cuando el roedor escoge erróneamente

su respuesta.

- **Tiempo Alimento:** Este tiempo corresponde al intervalo que se proporciona para que el roedor se alimente luego de que el usuario a cargo presione el botón PELLET en el PF, que será visto en la sección de controles siguiente.

2. Configuración de sistema

Para comenzar una medición se debe definir previamente algunos parámetros, los cuales son los siguientes:

- **Número de muestras:** Es el número de pares que se mostrarán en total a lo largo de la medición.
- **ID roedor:** Cada roedor posee un número identificador de tres dígitos.
- **Número medición:** Como es probable que se realice más de una medición por roedor, este número registra el número de medición.
- **Pantalla Derecha/Izquierda:** Este control despliega una pantalla en la cual se debe elegir cuál es el puerto usb en el cual está conectada cada pantalla.
- **Cámara de video:** Se debe escoger una cámara entre las disponibles para grabar y mostrar imágenes.
- **Dirección de Video/Reporte:** Corresponde a la dirección de guardado del video y del reporte generado respectivamente.

3. Indicadores de estado

Para saber el estado en tiempo real en cual se encuentra el sistema, se dispone de cinco indicadores en el panel frontal.

- **Pantalla Izquierda/Derecha:** Para indicar cual pantalla es la que presenta el estímulo correcto, se utilizan dos indicadores cuadrados de color azul. Cuando se presenta un estímulo correcto, se enciende y cuando es incorrecto, se mantiene apagado.
- **Palanca Izquierda/Derecha:** Bordeando cada indicador de pantalla en azul, existe otro indicador, que refleja el estado de las palancas presionadas. Estos se encienden cada vez que una palanca es presionada y permanece así hasta que es liberada.

- Correcto/Incorrecto: Cuando el roedor analizado acierta o se equivoca, se enciende el indicador led correspondiente. El indicador de respuesta correcta es verde y el de respuesta incorrecta, rojo.
- Listo!: Este indicador señala cuando el estado del sistema se encuentra esperando por la respuesta del roedor. El tiempo que se espera en este estado, se define como Tiempo Listo!, indicado anteriormente.

4. Visor de cámara

Este indicador es el de mayor tamaño en el panel frontal. Permite visualizar las imágenes adquiridas por la cámara de video a tiempo real.

5. Gráfico de respuestas

Para ilustrar el desempeño del roedor examinado, se dispone de un grafico con tres curvas. Estas tres curvas corresponden a la acumulación de respuestas correctas, incorrectas y omitidas. Este mismo gráfico es incluido en el reporte generado al final de la sesión de medición.

6. Secuencia de muestras por pantalla

A los costados del visor de cámara, se encuentran dos columnas de indicadores, mostrando la secuencia de estados a seguir por cada una de las pantallas. Estos indicadores muestran cual pantalla contiene el estado correcto en cada par de estímulos estando encendido de color verde y al contrario, cuando es incorrecto, estando apagados.

7. Tabla de respuestas y muestra actual

Al igual que el gráfico de respuestas, este indicador muestra el desempeño instantaneo del roedor examinado. Mientras el gráfico entrega una percepción principalmente cualitativa (aun siendo cuantitativa), la tabla cuantifica la acumulación de respuestas del roedor. Y de forma complementaria, se incluye un indicador del número de muestra actual en el proceso de medición.

8. Fecha y Hora

De forma complementaria, se incluye un indicador de hora y fecha.

3.1.2. Controles

Además de indicadores, en el panel frontal se dispone de controles para ingresar o modificar datos que serán utilizados para el proceso de medición. Según la Figura 3.1, en color azul, podemos ver los siguientes controles:

1. Iniciar, Parámetros temporales y Configuración de sistema

El primer botón, INICIAR, como su nombre lo indica, inicia la medición luego de haber ingresado los parámetros necesarios. El botón PARÁMETROS TEMPORALES despliega una ventana de diálogo que solicita ingresar los parámetros temporales mencionados anteriormente y que posteriormente se mostrarán en el indicador correspondiente. Y finalmente, el botón CONFIGURACIÓN DE SISTEMA, similar al anterior, solicita el ingreso de los parámetros de configuración mencionados anteriormente, los cuales se presentan en los indicadores correspondientes en el PF.

2. Pellet, Ruido Blanco, Sonido Castigo y Luz Castigo.

El botón PELLETT entrega la cantidad de pellets indicada a la izquierda del botón, como recompensa cuando el usuario estime conveniente. Adicionalmente a lo solicitado, se incluye la posibilidad de emitir ruido blanco durante la sesión de medición utilizando los parlantes disponibles del computador. La finalidad del ruido blanco es la de enmascarar los sonidos de fondo que se puedan generar en el entorno y que puedan perturbar o distraer al roedor bajo análisis. El ruido blanco, el sonido de castigo y luz de castigo que se aplican cuando el roedor se equivoca, pueden ser activados y desactivados cuando se necesite. Todos estos botones poseen un led indicador de estado, incluyendo además un control de volumen para el ruido blanco.

3. Observaciones y Generar Reporte

En el PF se dispone de un espacio para ingresar texto durante o al final de cada medición. Este texto ingresado aparecerá en el reporte generado, el cual solo se generará una vez finalizada la medición y presionando el botón GENERAR REPORTE.

Una vez definido el PF que utilizará el usuario, se deben programar las funciones necesarias para hacer funcionar este sistema. La próxima sección presenta el DB construido en LabVIEW que toma estos indicadores y controles para generar un software de control.

3.2. Diagrama de Bloques

Como se mencionó previamente, en el DB se llevan a cabo las conexiones entre constantes y variables a través de bloques que realizan distintas funciones. El marco global considerado para construir este diagrama de bloques es un *Flat Sequence Structure*, marco que permite una ejecución secuencial de subdiagramas o cuadros asegurando un orden de ejecución en el programa desarrollado. Esto implica que para avanzar de un cuadro a otro es necesario completar la ejecución y tener todos los resultados que pueden ser entradas del cuadro siguiente. Esto es particularmente útil, ya que de esta forma se evitan carreras en la ejecución de códigos y se optimiza la carga de procesamiento en el computador. En las siguientes secciones se especificará el desarrollo y funcionamiento de cada cuadro de esta estructura programada, siendo en total nueve cuadros secuenciales.

3.2.1. Primer Cuadro

Al comenzar la ejecución de este programa, se despliega la ventana de diálogo mostrada en la Figura 3.2.

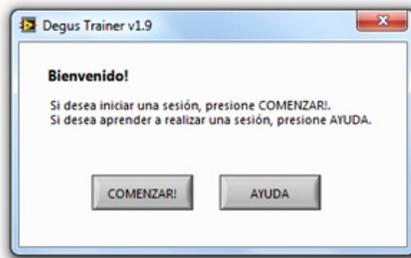


Figura 3.2: Ventana de diálogo - Bienvenida

Esta ventana permite continuar con el programa e iniciar una medición o solicitar ayuda, abriendo un documento PDF que explica paso a paso como se realiza una medición. Las instrucciones de uso del programa desplegadas cuando se presiona el botón AYUDA, se pueden encontrar en el Anexo F. En paralelo, los botones INICIAR, PARÁMETROS TEMPORALES, CONFIGURACIÓN DE DISPOSITIVOS, RUIDO BLANCO, LUZ CASTIGO, SONIDO CASTIGO y el control que establece la cantidad de pellet a entregar, se habilitan para ser utilizados.

Una vez ejecutado este cuadro, habiendo escogido continuar con la medición o habiendo solicitado ayuda, se continúa con la ejecución del segundo cuadro. Para referirse a la vista completa del primer cuadro, ver Anexo A, Figura 8.1.

3.2.2. Segundo Cuadro

El segundo cuadro, ejecutado inmediatamente luego de finalizar el primero, permite ingresar los parámetros necesarios para comenzar una medición.

Los botones del PF, PARÁMETROS TEMPORALES y CONFIGURACIÓN DE DISPOSITIVOS, son censados para detectar si han sido presionados. Si han sido presionados, se abre una ventana de diálogo como se ve en la Figura 3.3 y la Figura 3.4, respectivamente, permitiendo introducir los parámetros requeridos para iniciar una medición. Posteriormente, al cerrar la ventana de diálogo, los indicadores respectivos en el PF se actualizan mostrando los valores introducidos por el usuario.

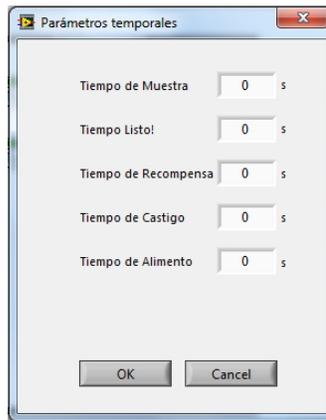


Figura 3.3: Ventana de diálogo - Parámetros Temporales

Para detectar si alguno de ambos botones fue presionado, los botones son censados dentro de un *While Loop*. La condición para salir del loop es que todos los parámetros solicitados sean ingresados y posteriormente, que el botón INICIAR sea presionado. Una respuesta de acción booleana de los botones PARÁMETROS TEMPORALES y CONFIGURACIÓN DE DISPOSITIVOS es emitida cada vez que alguno de estos botones es presionado, almacenándose en una matriz conectada a un *Case Structure*. Esta estructura permite tomar una acción específica

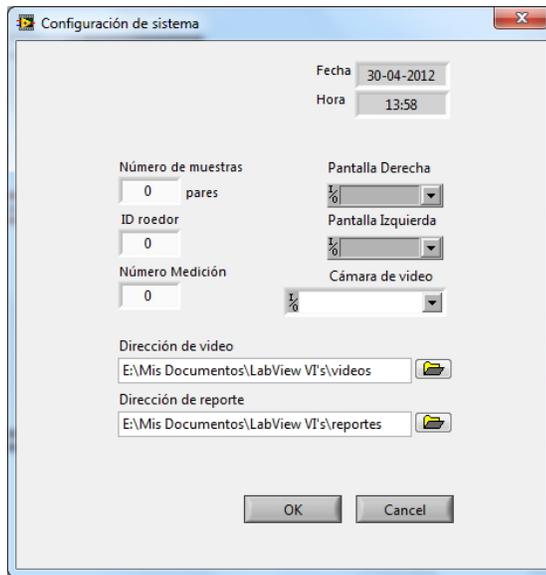


Figura 3.4: Ventana de diálogo - Configuración de dispositivos

por cada botón. En total, se tienen tres caso en el *Case Structure* los cuales serán explicados a continuación.

Antes de comenzar, cabe mencionar que se hace referencia a ciertas subVI creadas las cuales se presentan en detalle en el Anexo B.

- Caso 0 (Default)

El caso 0, Figura 3.5 corresponde al momento en que ambos botones representan un 0 booleano y no se toma acción alguna. Tanto los parámetros temporales como la configuración de los dispositivos se encuentran vacías y son leídas a través de variables locales asociadas a los indicadores correspondientes.

- Caso 1

Cuando el botón PARÁMETROS TEMPORALES es presionado, se ingresa al caso 1, Figura 3.6. Este caso ejecuta el subVI `temporal parameters.vi` desplegando la ventana de diálogo correspondiente a la Figura 3.3. Este subVI permite guardar los parámetros ingresados en un cluster. Además, el botón presionado es reinicializado y el cluster asociado a la configuración de sistema es leído a través de una variable local. Este cluster puede

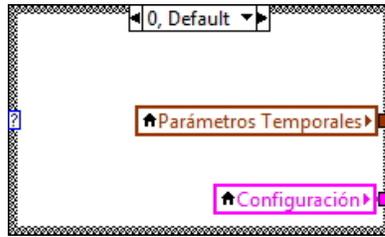


Figura 3.5: Diagrama de bloques - Caso 0 (Default)

estar vacío o ya completo, en el caso que se haya presionado nuevamente para corregir algún parámetro.

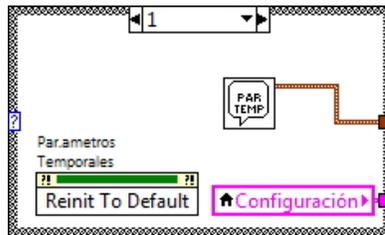


Figura 3.6: Diagrama de bloques - Caso 1

- Caso 2

En caso de presionar el botón CONFIGURACIÓN DE DISPOSITIVOS, se ingresa al caso 2, ejecutando el subVI `system configuration.vi` que despliega la ventana de diálogo, Figura 3.4. Al igual que en el caso anterior, los parámetros ingresados se almacenan en un cluster.

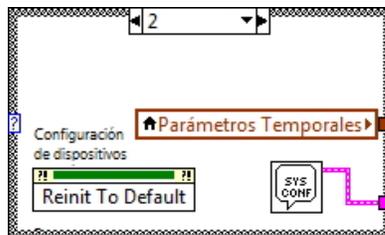


Figura 3.7: Diagrama de bloques - Caso 2

Una vez fuera del *Case Structure*, se comprueba que los parámetros ingresados sean correctos,

ejecutando el subVI `complete indicator.vi`. La salida de este subVI es booleana, la cual es 1 si todo se encuentra correctamente ingresado y un 0 en caso contrario. Junto a esta variable booleana en 1, más la verificación de que la cantidad de pellet a entregar sea positiva y distinta de cero y cuando el botón INICIAR sea presionado, la ejecución del *While Loop* es finalizada. En caso de que no se cumplan las condiciones mencionadas anteriormente y se presione el botón INICIAR, se despliega una ventana de advertencia, permitiendo corregir los valores ingresados. Este cuadro, al igual que el anterior, se encuentra en el Anexo A, Figura 8.1.

Una vez ejecutado este cuadro y habiendo recolectado los parámetros necesarios para iniciar una medición, se pasa al siguiente cuadro.

3.2.3. Tercer Cuadro

El tercer cuadro ejecutado en el DB inicializa la memoria microSD montada en la pantalla en la cual se incluye la librería de imágenes que se presentarán como estímulos visuales. También inicializa la cámara de grabación y se deshabilitan los controles del PF utilizados en el cuadro anterior.

- Inicialización de memoria microSD

Es en este cuadro se realizan las primeras operaciones entre dispositivos externos y el software de control. La pantalla lleva montado un lector de memoria microSD el cual debe ser inicializado a través de una secuencia de comandos seriales enviados por puerto USB a través del cable adaptador provisto por el fabricante.

Para comenzar a utilizar la pantalla, se debe enviar el comando `autobaud`, que corresponde a un 55 hexadecimal. Este comando permite sincronizar el *baud rate* de la pantalla a la del computador. Luego, se envía el comando `InitUSD`, el cual es un 4069 hexadecimal.

Para comprobar el envío de comandos a través *4DGL Workshop*, se envían los comandos en forma según instrucciones del fabricante como se muestra en la Figura 3.8. Este código establece el *baud rate* de la pantalla según el ofrecido por el computador con el comando `family autobaud`, luego inicializa la memoria microSD con `InitUSD`, para luego mostrar imágenes en pantalla con el comando `USDImage` con un intervalo de tiempo de 1 segundo con el comando `Delay`.

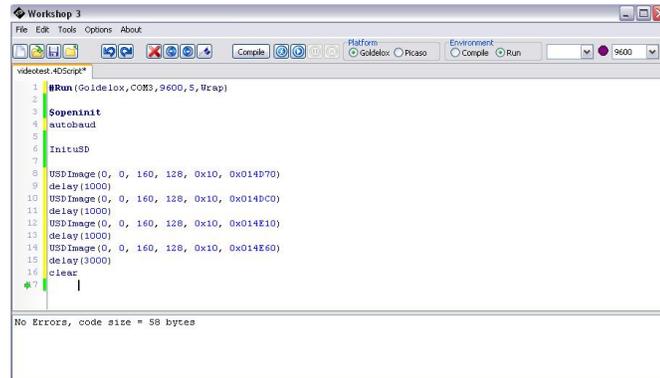


Figura 3.8: 4DGL Workshop 3

Entonces, lo que se debe lograr es emular esta transmisión de comandos utilizando LabVIEW. Para cumplir con esta tarea, se crea el subVI `sd init.vi`.

Usando las herramientas provistas por VISA Development Toolkit, módulo de LabVIEW, podemos enviar comandos seriales al puerto que escojamos. Para esclarecer el funcionamiento de este subVI, se presenta el diagrama de bloques elaborado en la Figura 3.9.

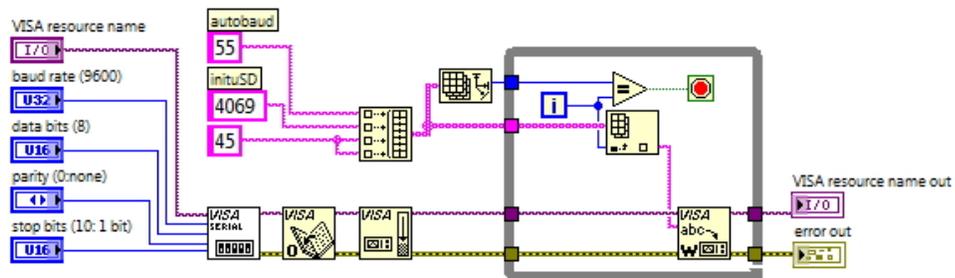


Figura 3.9: Diagrama de bloques - Inicialización uSD

Primero, se configura un puerto serial a través de controles que determinan el formato de transmisión. En este caso, se considera un *baud rate* de 9600 S/s, 8 bits de datos, sin paridad y 1 bit de detención tal cual es especificado en el manual de usuario de la pantalla utilizada (manual de referencia en sitio sugerido en Anexo C). Y el parámetro de control

VISA *Resource Name* permite seleccionar el puerto serial del computador a utilizar.

Paralelamente se crea un arreglo de string hexadecimales con comandos, partiendo por la función `Auto baud rate` (55 hex), luego el comando `InitSD` (4069 hex), que inicializa la memoria microSD y adicionalmente el comando `Clear Screen` (45 hex) que limpia la pantalla, dejandola preparada para comenzar.

Este arreglo de comandos y la configuración del puerto serial, son entrada a un *While Loop* en donde se envían los comandos hexadecimales secuencialmente por cada iteración del loop. Saliendo del loop en el momento en que la iteración sea igual al tamaño de la matriz.

Los parámetros de salida de este módulo son VISA *Resource Name Out* y *Error out*, que sirven de entrada para los módulos siguientes.

Para mayores especificaciones de este subVI, ver Anexo B.

- Inicialización de librería de imágenes

Los estímulos visuales que se presentan en ambas pantallas, son imágenes almacenadas en la memoria microSD montada en cada uno de los módulos de pantalla.

Estas imágenes son almacenadas en la memoria microSD con un formato y orden específico a través de un programa entregado por el fabricante de las pantallas utilizadas. Este programa es *Graphic Composer 3*, del cual podemos ver su interfaz en la Figura 3.10.

Este programa permite cargar tanto imagenes como videos de resolución acorde a la aplicación, que en este caso son imágenes de 160x128 pixeles. Una vez adjuntas todas las imágenes necesarias en *Graphic Composer 3*, se cargan las imágenes en la memoria microSD y se genera un archivo .txt que entrega la dirección en memoria de cada una de las imágenes.

Cada una de estas direcciones es tomada y copiada en el subVI `image library.vi`, la cual genera como salida principal, un arreglo de strings hexadecimales que contiene las direcciones de las imágenes y los comandos necesarios que permiten desplegar estas imágenes en la pantalla. Este subVI, `image library.vi`, se encuentra en detalle en Anexo B, sección 8.2.3.

El proceso detallado para cargar imágenes con el programa *Graphic Composer 3* se encuentra en el Anexo E.

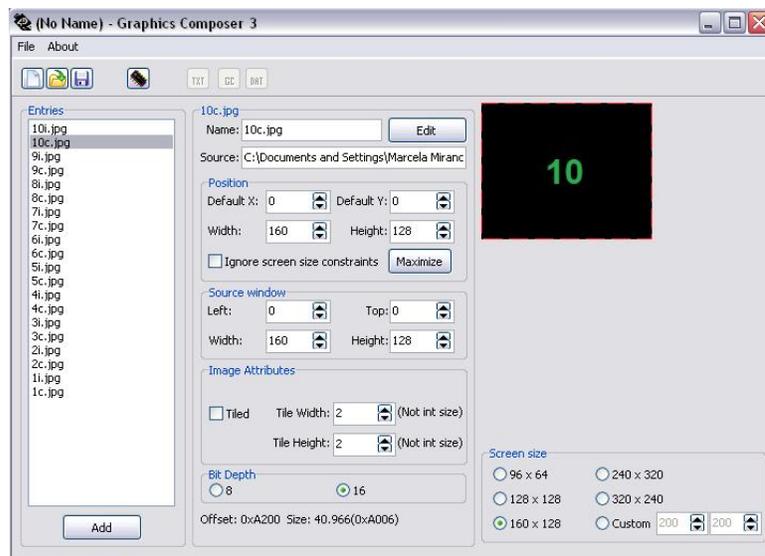


Figura 3.10: Interfaz de Graphic Composer 3

- Inicialización de Cámara

La cámara web utilizada para supervisar al roedor examinado y generar el video de reporte, debe ser inicializada utilizando las herramientas proporcionadas por IMAQ Development Toolkit. Primero, se realiza una conexión con la cámara que se utilizará, luego se genera una sesión IMAQ que permite guardar un video en formato .avi y finalmente, existiendo el modo captura continua y captura única, se escoge captura continua.

La inicialización de la cámara provee de parámetros necesarios a la función de adquisición continua de video que se realiza fuera de cualquier cuadro en el diagrama de bloques, por lo que será explicado con detalle al final de este capítulo.

El formato de nombre utilizado para el video generado se compone de tres elementos. El código de roedor (tres dígitos), la fecha actual (sin separadores) y el número de sesión con ese roedor (tres dígitos). Estos datos, exceptuando la fecha, son solicitados al usuario en la etapa de *Configuración de dispositivos* en el segundo cuadro.

- Deshabilitación de controles

De forma similar al primer cuadro, en este cuadro se utilizan nodos de propiedad para

dejar deshabilitados y en gris los mismos controles que fueron habilitados.

Para ver en detalle la distribución de bloques en este cuadro, ver Anexo A, Figura 8.2.

3.2.4. Cuarto Cuadro

Para mostrar estímulos en ambas pantallas, se necesita establecer cual pantalla mostrará el estímulo correcto, cual el incorrecto y cual imagen se presentará en cada pantalla. Es por este motivo que en este cuadro se generan cuatro matrices que contienen esta información para poder ser transportada al cuadro siguiente. Esta información es solicitada al usuario a través de una ventana de diálogo, mostrada en la Figura 3.11. Como se ve en la figura, se tienen dos opciones, ALEATORIA o MANUAL.

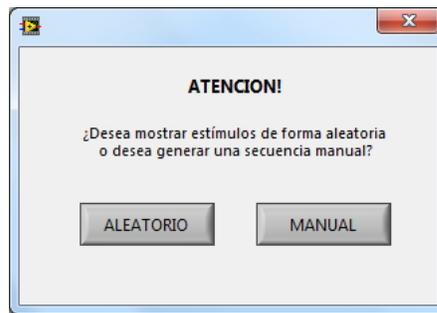


Figura 3.11: Ventana de diálogo - Generación de Secuencia

La diferencia entre escoger la opción ALEATORIA o MANUAL reside en que al presionar el primer botón, las matrices tendrán valores aleatorios, mientras que si se escoge la segunda opción, MANUAL, esta permite configurar manualmente por cada par de estímulos, cuales serán las imágenes a mostrar y cual pantalla presentará el estímulo correcto o el incorrecto.

Cuando la opción ALEATORIA es la escogida, se despliega una ventana que permite generar la secuencia aleatoria y luego permite que el usuario valide esta secuencia. Esta ventana emergente se muestra en la Figura 3.12, la cual además muestra como referencia el total de estímulos correctos e incorrectos que aparecerán en la pantalla izquierda. En este caso la secuencia que determina cual pantalla mostrará el estímulo correcto, el estímulo incorrecto y la imagen que se despliega en cada pantalla, es de origen aleatorio.

Al contrario, en el caso que se desee ingresar manualmente la configuración de la secuencia

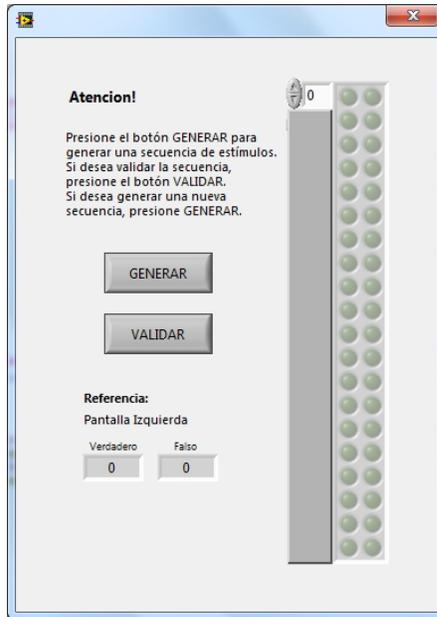


Figura 3.12: Ventana de diálogo - Validador de Secuencia

de estímulos, se presiona el botón MANUAL, que despliega la ventana de diálogo presente en la Figura 3.13.



Figura 3.13: Ventana de diálogo - Ingreso manual de Secuencia

El control numérico izquierdo, indica el número de par de estímulos, mientras que los controles numéricos dentro del recuadro, indican el número o código del estímulo que se desea mostrar. Y el switch deslizante en medio permite seleccionar cual pantalla mostrará el estímulo correcto. Sea una secuencia aleatoria o manual, en el PF se encuentran dos columnas que indican la secuencia de estados de cada pantalla. Para ejecutar estas funciones descritas, se utiliza

una variable local que lee el cluster de configuración de dispositivos, extrayendo el número de muestras. Este valor es una entrada a la subVI `stimulus_array.vi`. Dentro de esta subVI, se encuentra un *Event Structure* que permite ejecutar acciones diferentes en caso que se desee una secuencia ALEATORIA o MANUAL.

Para la ejecución de estas acciones, se utilizan dos subVI llamadas `manual_array.vi` y `random_array.vi` para el caso MANUAL y ALEATORIO respectivamente. El funcionamiento detallado de estas subVI se encuentra en el Anexo B.

La distribución de los elementos en este cuadro y la conexión entre ellos se puede ver en el Anexo A, Figura 8.2.

3.2.5. Quinto Cuadro

El quinto cuadro ejecuta un subVI de nombre `prepare_count.vi`. Este subVI realiza una cuenta regresiva de resguardo antes de comenzar la presentación de estímulos en las pantallas. La tarea realizada en este cuadro es solo de carácter útil para el usuario, ya que no tiene ningún efecto funcional sobre la aplicación. Para conocer más detalle de este subVI, ver Anexo B.

3.2.6. Sexto Cuadro

El sexto cuadro, es el cuadro principal del programa desarrollado, en el cual se realizan principalmente tres funciones.

- Presentación de estímulos en ambas pantallas.
- Adquisición de respuesta del roedor examinado y comparación con respuesta esperada.
- Entrega de recompensa o ejecución de medidas correctivas.

Estas tres funciones involucran varios pasos a ejecutar explicados a continuación.

Presentación de estímulos

La presentación de estímulos en pantalla es la base de funcionamiento de este proyecto. Para lograr esta tarea se crea un subVI que envía comandos de forma serial al microcontrolador montado en la pantalla que gestiona la información recibida para buscar determinadas imágenes

en la memoria microSD, para luego desplegarlas en la pantalla. Esta subVI es `two screens.vi`, alojada dentro de un *For Loop*.

Este *For Loop* permite considerar una a una las filas de los arreglos que contienen la información de cual imagen mostrar en cada pantalla y cual es el estímulo correcto, mostrando las imágenes en el instante y durante el tiempo determinado.

Es esta información entra al subVI `two screens.vi` y es procesada para poder mostrar la imagen correspondiente en pantalla. El diagrama de bloques de este subVI se muestra en la Figura 3.14.

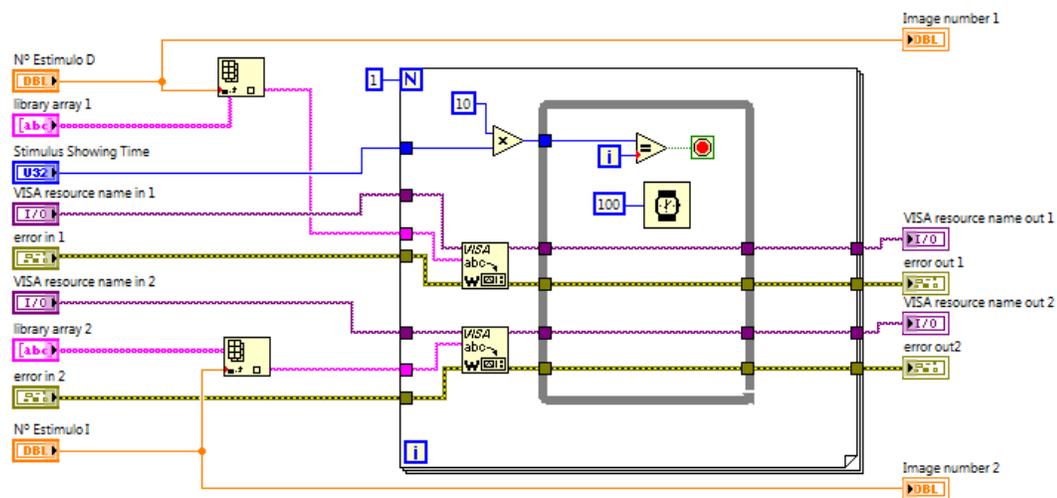


Figura 3.14: Diagrama de bloques - Estímulos en pantallas

Los parámetros entran a un *For Loop* de solo una iteración la cual envía a ambas pantallas la línea de código hexadecimal necesaria para desplegar la imagen correspondiente. Luego, se incluye un *While Loop* que permite presentar las imágenes el tiempo necesario y finalmente se envía el comando hexadecimal para borrar pantalla.

Dentro del mismo loop, se incluye un *For Loop* anidado que censa las palancas en el ambiente de análisis cada 10 milisegundos. El desarrollo de esta función se detalla a continuación.

Adquisición de respuesta y comparación

Como se menciona anteriormente, dentro del *For Loop* principal se incluye otro *For Loop* secundario que permite detectar cuando una palanca es presionada siendo censada cada 10 milisegundos. Este umbral de tiempo es determinado para proveer robustez en la adquisición de respuesta, ya que es prácticamente improbable que el roedor examinado accione una palanca durante un tiempo menor a 0,01 segundos y también se escoge este umbral para no sobreexigir en recursos de procesamiento al computador usando una tasa de muestreo mayor de forma innecesaria. Esto quiere decir que al disminuir el umbral de tiempo, el número de veces que se ejecuta el ciclo *for* aumenta, exigiendo de forma innecesaria al computador, siendo este el principal motivo de la elección de un tiempo de censado.

Para adquirir la respuesta desde las palancas, se utiliza un *While Loop* fuera del *For Loop* principal. Este *While Loop* contiene un bloque de adquisición NI DAQ para cada pantalla. Cada bloque se encuentra configurado como línea de entrada digital y conectado a un indicador LED en el panel frontal. La ejecución de este *While Loop* culmina cuando termina el tiempo de muestra del último par de estímulos. El diagrama de bloques de esta función puede verse en la Figura 3.15. Para ingresar este valor al *For Loop* principal, se invoca a través de variables locales. Este detalle puede verse en el Anexo A, Figura 8.4.

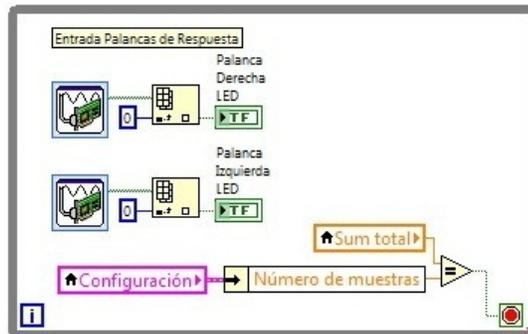


Figura 3.15: Diagrama de bloques - Censo de palancas a través de entradas digitales NI DAQ

El diagrama de la Figura 3.16 muestra los distintos estados en los que el programa puede encontrarse durante el proceso de presentación de estímulos y adquisición de la respuesta proveniente desde la palanca accionada por el roedor en cuestión.

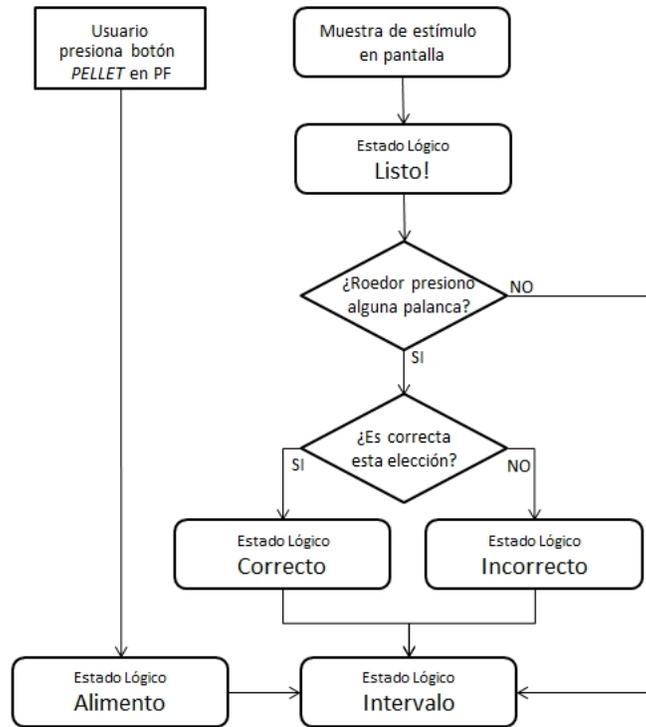


Figura 3.16: Diagrama de estados - Muestra de 1 par de estímulos

El proceso se inicia cuando un par de estímulos es presentado al roedor analizado, luego, es posible que el roedor presione alguna de las dos palancas o simplemente las ignore y no interactúe con el sistema durante la muestra de estímulos. En el primer caso, si presiona una palanca, es posible que sea la respuesta esperada o puede que no lo sea. En el caso que si lo sea, se ingresa al estado CORRECTO, premiando al roedor por su conducta con tantos pellets se haya determinado. En el caso contrario se ingresa al estado INCORRECTO, negando recompensa y enfatizando error con una señal auditiva. Una vez pasado el tiempo de recompensa o castigo, se ingresa al estado final, el estado INTERVALO. Este estado tiene la función evitar una muestra secuencial sin separaciones entre muestra que puedan llevar a confusión al roedor.

En el caso que el roedor no interactúe con el sistema, el estado LISTO! se mantiene el tiempo determinado por el usuario para posteriormente pasar al estado INTERVALO. El caso particular de este sistema de control es el estado ALIMENTO, al cual solo se ingresa cuando el usuario

supervisor presiona el botón PELLET en el PF del software de control. Este estado permite entregar recompensa al roedor cuando el usuario lo determine necesario y puede ser ejecutado a lo largo de la sesión de medición y en cualquier momento.

La determinación del estado de cada acción realizada por el roedor se lleva a cabo por el subVI `logic module.vi`. Como salida de este subVI, se obtiene un dígito el cual se encuentra asignado a cada caso y es la entrada a un *Case Structure*, alojado dentro de un *Flat Sequence Structure* de dos cuadros, estando el *Case Structure* mencionado en el primero de estos cuadros. Cada caso de este *Case Structure* realiza las operaciones necesarias para generar acciones de recompensa, castigo y más, explicados a continuación

En este *Case Structure* se tiene un caso por defecto, en el cual se permanece siempre que no ocurra ninguna situación que determine pasar a otro caso. En este programa, el caso por defecto es el caso LISTO!. Como podemos recordar, en el panel frontal, se define un tiempo LISTO! que es el tiempo que se da al roedor para responder después de terminar el tiempo de muestra del estímulo. Este caso se muestra en la Figura 3.17, denotando su simpleza de implementación.

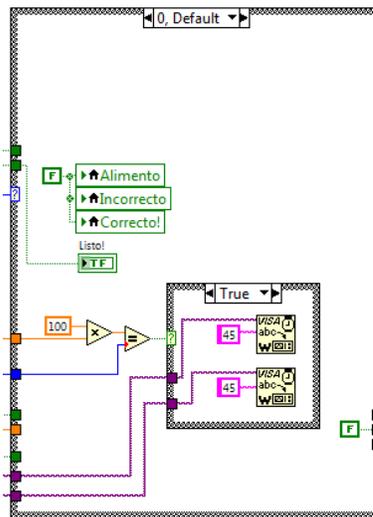


Figura 3.17: Estado lógico - Listo

Si no se presiona ninguna palanca, a la siguiente iteración después de 10 milisegundos, se vuelve a ingresar a este caso y así sucesivamente hasta que ocurra algún evento que lo interrumpa.

Los posibles eventos que interrumpan este caso son las cuatro situaciones mencionadas pre-

viamente.

Partiendo por la primera situación, la palanca correcta es presionada, se ingresa al caso CORRECTO, que se muestra en la Figura 3.18.

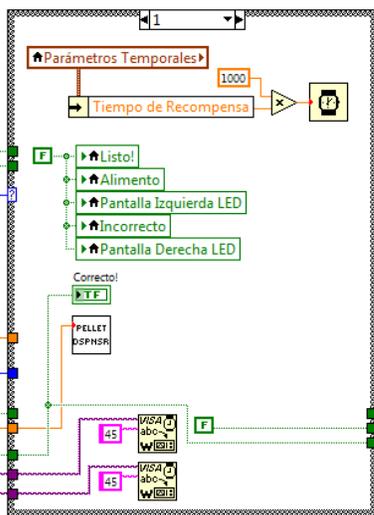


Figura 3.18: Estado lógico - Caso Correcto

El caso CORRECTO permite entregar una recompensa, una cantidad de pellets determinados por el usuario a través del panel frontal, accionando el dispensador de alimento utilizando una salida TTL de la tarjeta NI DAQ. Como se mencionó en las características del dispensador de pellets, para entregar un pellet se debe enviar un impulso de al menos 20 ms al dispensador de alimentos. Este impulso pasa desde la tarjeta NI DAQ hacia un circuito de adaptación y posteriormente, al dispensador de alimento. Esta función se lleva a cabo a través del subVI `pellet dispenser.vi`, el cual puede verse en detalle en el Anexo B, sexto cuadro.

Además de entregar recompensa, se enciende el LED que acusa una respuesta correcta en el PF y se apagan todos los demás LED de estado.

En el caso que la palanca presionada sea la incorrecta, se da paso al caso INCORRECTO, Figura 3.19. Este caso no entrega recompensa, sino que se ejecutan las medidas correctoras activadas previamente en el panel frontal, como puede ser LUZ CASTIGO y/o SONIDO CASTIGO. Los indicadores LED del PF se apagan exceptuando por el indicador de respuesta incorrecta. La emisión de un pulso de luz de castigo se lleva a cabo por el subVI `light pulse.vi`. Este subVI

actúa de forma similar al subVI `pellet dispenser.vi`, ya que el principio de operación es el mismo. Esta función es funcional a nivel de software, pero no se implementará a nivel de hardware hasta que el ambiente de análisis sea fabricado.

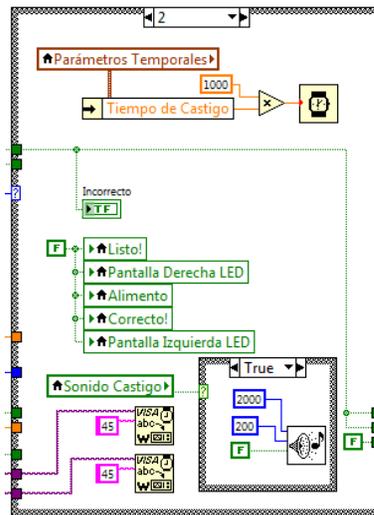


Figura 3.19: Estado lógico - Caso Incorrecto

Cuando ocurre que el roedor no presiona ninguna palanca, se pasa a un estado intermedio de resguardo. Este estado definido como INTERMEDIO, permite separar el estado LISTO! del tiempo de muestra de las imágenes en pantalla y también permite resetear los indicadores LED en el PF. Este caso se muestra en la Figura 3.20.

Paralelamente a la posible respuesta que pueda dar el roedor con alguna palanca, se tiene el botón PELLETT en el panel frontal, que permite entregar la cantidad de pellet requerida en cualquier momento durante la ejecución de la medición. El siguiente cuadro, Figura 3.21, muestra el diagrama de bloques del caso en que se entrega alimento bajo esta condición.

Entonces, utilizando estos casos que permiten acciones distintas para cada situación, se generan salidas para poder ser incluidas en la ejecución de otras tareas en el cuadro de ejecución siguiente y en este mismo cuadro, como se explica a en la siguiente sección.

El segundo cuadro del *Flat Sequence Structure* mencionado, contiene una función simple pero importante para el correcto desarrollo de la sesión de medición. Esta función prohíbe continuar con el siguiente par de estímulos si alguna de las palancas se encuentra presionada.

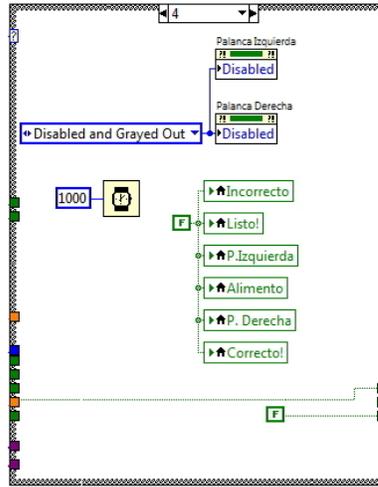


Figura 3.20: Estado lógico - Caso Intervalo

Para lograr tal efecto, se tiene un *While Loop* que censa las palancas continuamente y se detiene solo cuando ambas palancas son un cero lógico, lo que equivale a que ambas se encuentra sin estar presionadas. El diagrama de bloques de esta función de resguardo se puede ver en la Figura 3.22.

Gráfico de respuesta a tiempo real

En el panel frontal se presenta un gráfico que despliega a tiempo real el desempeño del roedor. Este gráfico genera tres curvas indicando los aciertos, yerros y omisiones acumuladas en negro, rojo y verde respectivamente. La utilidad de este gráfico reside en análisis posterior a la medición en el cual se puede determinar patrones de conducta, preferencias a ciertos estímulos, incapacidad de comprensión y más. En la Figura 3.23 se muestra el espacio designado en el panel frontal donde se genera el gráfico de respuestas acumuladas a tiempo real.

Una copia de este gráfico generado durante la sesión de medición es incluido en el reporte final generado. Un ejemplo de este gráfico incluido en el reporte, puede verse en la Figura 3.24, donde puede verse la respuesta acumulada por número de muestra.

Detrás de la presentación de este gráfico, existe un diagrama de bloques generado que utiliza datos obtenidos en bloques anteriores. Este diagrama de bloques programado se muestra en la Figura 3.25.

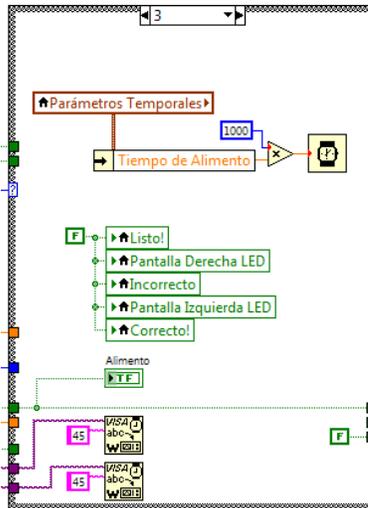


Figura 3.21: Estado lógico - Caso Alimento

Desde el *Case Structure* se obtiene la respuesta del roedor y al salir del *For Loop* se obtiene una matriz ordenada de respuestas correctas, incorrectas y omitidas. De estas matrices, se realiza una suma cada vez que se presenta una imagen, es decir, cada vez que se presenta un nuevo estímulo en pantalla, se realiza un nuevo computo de la cantidad acumulada de respuestas que finalmente son reflejadas en el gráfico en el panel frontal.

Imagen de ruido blanco

Para finalizar la medición y para dar a entender al roedor cuando se termina una sesión de medición, se presenta una imagen de ruido blanco en ambas pantallas. Para llevar a cabo esta función, se crea un subVI de nombre `screen noise.vi` que envía una sucesión de comandos que permite mostrar una imagen previamente almacenada en la memoria microSD en la primera posición del repertorio de imágenes.

Luego de ejecutar este subVI, se utiliza un bloque *VISA Close* para cerrar sesión en ambas pantallas.

Para ver la disposición de los elementos de este cuadro, ver Anexo A, Figura 8.3 y 8.4. Para conocer mayores detalles de los subVI insertos en este cuadro, ver Anexo B.

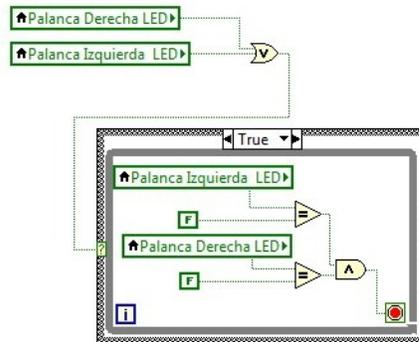


Figura 3.22: Diagrama de bloques - Resguardo de palanca presionada

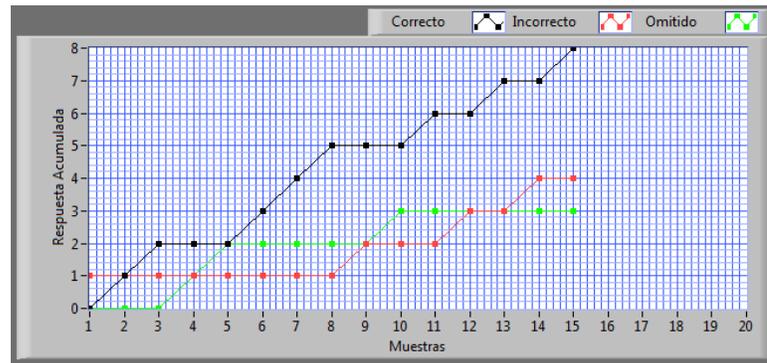


Figura 3.23: Gráfico de respuestas acumuladas en panel frontal

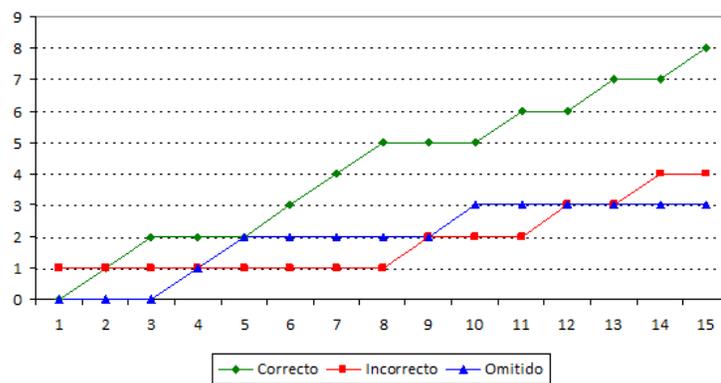


Figura 3.24: Ejemplo de gráfico de respuestas acumuladas para reporte final

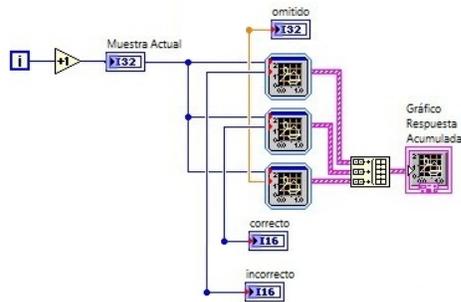


Figura 3.25: Diagrama de bloques - Gráfico respuesta tiempo real

3.2.7. Séptimo Cuadro

Una vez mostrados todos los estímulos y habiendo terminado la ejecución del sexto cuadro, se abre una ventana que avisa al usuario que la sesión ha terminado y da la oportunidad de complementar el informe ingresando comentarios en el cuadro respectivo en el panel frontal y finalmente, presionar el botón GENERAR REPORTE para pasar al siguiente cuadro de ejecución. La ventana emergente con este mensaje se ve en la Figura 3.26.



Figura 3.26: Ventana de diálogo - Fin de sesión y generación de reporte

Al presionar OK en la ventana desplegada, también se detiene la grabación de video. Este cuadro puede verse en el Anexo A, Figura 8.4.

3.2.8. Octavo Cuadro

Utilizando las herramientas disponibles en LabVIEW para generar reportes en MS WORD, utilizamos una plantilla en la que se insertan los datos de cada medición realizada. Los datos incluidos en este reporte son:

- Parámetros temporales, identificación del roedor, n° de muestras, n° de sesión por roedor.

- Tabla de respuestas con columnas de n° de par de estímulos, n° de imagen por pantalla, latencia de respuesta, aciertos, errores y omisiones.
- Gráfico de respuestas acumuladas desde panel frontal.
- Comentarios agregados a través del cuadro de texto en panel frontal.

Toda esta información sirve como respaldo de la medición ejecutada y para posteriores análisis.

Paralelamente, se genera una tabla en MS EXCEL que contiene las respuestas de la sesión de medición finalizada. Además, se cuenta con un archivo EXCEL que se actualiza cada sesión, agregando una nueva fila con los siguientes datos:

- Parámetros temporales, identificación del roedor, n° de muestras, n° de sesión por roedor, latencia de respuesta, aciertos, errores y omisiones.

Estas tablas serán utilizadas como base de datos para poder ser utilizadas posteriormente y poder contar con información histórica almacenada. Este octavo cuadro puede verse en Anexo A, Figura 8.5.

3.2.9. Noveno Cuadro

Finalmente, el último cuadro de la secuencia, se encarga de resetear a su estado inicial los controles e indicadores necesarios. Entre estos, se encuentran los PARAMETROS TEMPORALES, CONFIGURACION DE SISTEMA, LEDS indicadores entre otros. Para mayor detalle, ver cuadro ejecutado en Anexo A, Figura 8.5.

3.2.10. Fuera de cuadro

Paralelamente a la ejecución secuencial de cuadros, existen funciones que se realiza fuera de esta secuencia. Es el caso de la captura de video y la emisión de ruido blanco.

Cámara

El diagrama de bloques 3.27 muestra las funciones de este módulo, partiendo por seleccionar la carpeta destino del archivo de video que se generará, luego se comienza la grabación y en

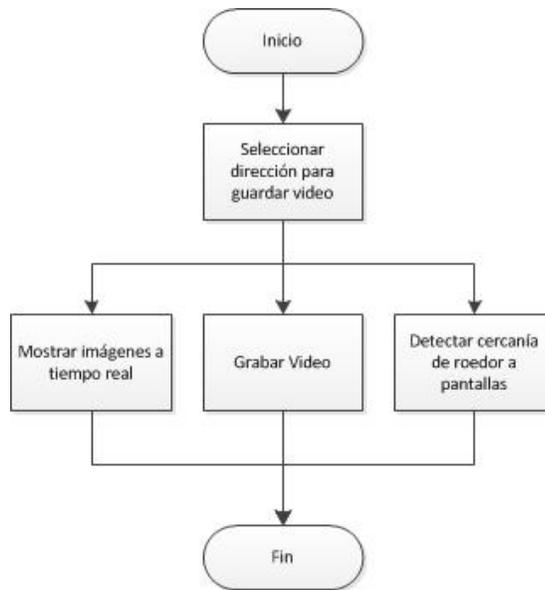


Figura 3.27: Diagrama de bloques - Cámara

paralelo se muestra a tiempo real las imágenes recibidas en el panel frontal. En el diagrama también se incluye la funcionalidad de la cámara como detector de proximidad. Esta función es opcional y se detalla en el capítulo: «Mejoras Potenciales al Proyecto».

Ruido Blanco

Para mantener el interés del roedor examinado en las pantallas y en su desempeño por obtener recompensa, es necesario aislar de toda distracción tanto visual como auditiva. En función del primer tipo de distracción, se maneja el entorno para que solo existan los elementos necesarios en el ambiente de análisis, mientras que el segundo tipo de distracción, es necesario intervenir el espacio con ruido blanco que oculte cualquier sonido que puede interferir en el desempeño del roedor examinado. Los posibles sonidos que se pueden encontrar, son por ejemplo, el dispensador de alimento en funcionamiento, el relé del switch de activación, y principalmente, cualquier sonido externo que pueda interferir con el correcto desarrollo de la medición.

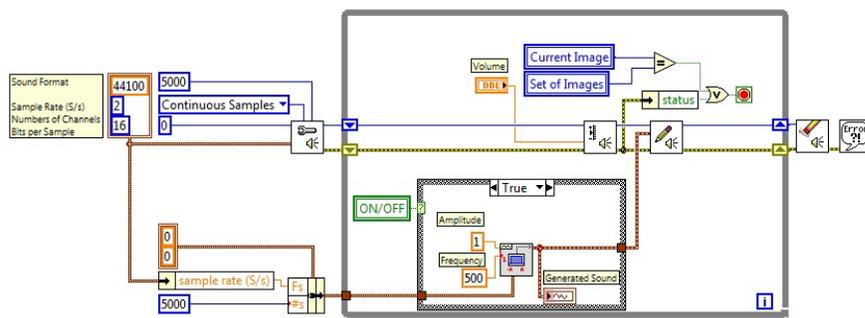


Figura 3.28: Diagrama de bloques - Sonido de refuerzo y rechazo

Capítulo 4

Montaje de dispositivos en ambiente de análisis

El espacio dedicado para montar los elementos que se utilizan en este ambiente de análisis consistirá en una caja de aluminio de seis caras. La tapa superior desmontable soportará la cámara de grabación y la cara frontal interna llevará montada dos pantallas, dos palancas y un receptáculo de pellets. En la Figura 4.1 se puede ver bosquejo de la distribución de elementos en el ambiente de análisis.

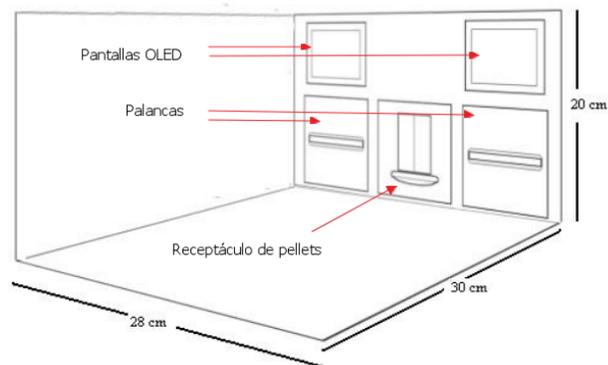


Figura 4.1: Diagrama diseño ambiente de análisis

Las medidas de esta caja son aproximadas, ya que las dimensiones definitivas deben ser espe-

cificadas por los expertos a cargo de la investigación, ya que es necesario lograr una proyección en la retina de un tamaño específico para los estímulos presentados en pantalla.

Fuera de la caja contenedora, se encuentra el dispensador de alimento y los parlantes que se utilizan para emitir la señal auditiva y ruido blanco. La siguiente figura, Figura 4.2, presenta una vista lateral de como será este ambiente de análisis.

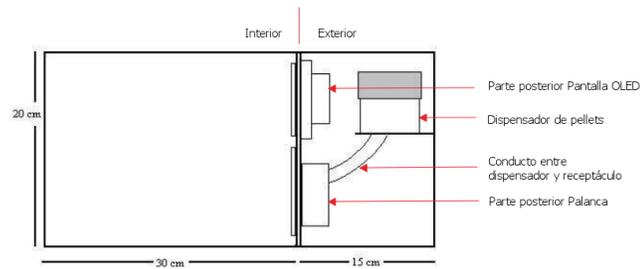


Figura 4.2: Diagrama diseño lateral ambiente de análisis

En esta imagen se puede ver el dispensador de alimento, pantallas, palancas y el conducto por el cual se lleva alimento desde el dispensador automático al receptáculo visto en la Figura 4.1. Los parlantes utilizados, se encuentran a los costados de la caja y no integrados a la estructura.

Dado el plazo estipulado para la realización de este proyecto, esta etapa se encuentra en desarrollo y las pruebas realizadas, se realizaron en base a una maqueta de similares características.

Capítulo 5

Pruebas

La maqueta construida se encuentra en el laboratorio de conducta de la facultad de ciencias de la Universidad de Valparaíso. Dada la cantidad de mediciones a la que se someterá este sistema, es necesario realizar pruebas de funcionamiento con el objetivo de descartar errores, tanto de hardware como de software. Por lo que en primera instancia, se define cuales son los procedimientos, paso a paso, que se deben realizar para iniciar una medición, desarrollando un documento de intrucciones el cual se crea con la finalidad de mostrarse cuando el usuario presione el botón AYUDA en la ventana emergente al iniciar el programa. Este documento puede encontrarse en el Anexo E.

De forma precisa y ya conociendo como opera el software de control, el procedimiento para realizar una medición de prueba es el siguiente:

- Abrir enlace directo en el escritorio del computador «Degus Trainer 1.9». Avanzar la primera ventana emergente para poder configurar parámetros de la sesión.
- Presionar los botones de configuración de parámetros y de sistema e ingresar todos los valores requeridos.
- Ajustar la cantidad de pellets que se entregarán como recompensa y activar sonido de castigo y ruido blanco según se estime conveniente. Dar inicio a la sesión presionando el botón INICIO en el panel frontal.

- Si la prueba lo considera, emular comportamiento de roedor presionando palancas según se haya determinado en la definición de la prueba.
- Una vez terminada la sesión de prueba, revisar el informe generado y contrastar con lo experimentado durante la sesión y verificar su validez. Revisar grabación de video registro de la sesión cursada si corresponde.

Las pruebas realizadas se basan en evaluaciones cualitativas y cuantitativas de resultados, los cuales reflejan niveles de funcionalidad y robustez del sistema elaborado.

5.1. Ejecución de pruebas y evaluación de resultados

Las pruebas realizadas consisten en realizar series de mediciones con la maqueta creada en ambiente controlado y sin roedores. La finalidad de estas pruebas es de certificar la operatividad del conjunto hardware y software, confirmando robustez y efectividad del sistema.

En total, se realizan 4 tipos de prueba, los cuales se detallan a continuación:

5.1.1. Prueba 1

Parámetro	Valor
Tiempo de Muestra	2 - 0.2 segundo
Tiempo Correcto	1 segundo
Tiempo Incorrecto	1 segundo
Tiempo Listo	1 segundo
Cantidad de muestras	10 pares
Cantidad de pellets	0 pellet

Cuadro 5.1: Parámetros de prueba - Prueba 1/2/3

Al iniciar la aplicación, se configuran los valores de parámetros con los que aparecen en el cuadro 5.1, realizando 10 pruebas variando tiempo de muestra desde 2 segundos hasta 0,2 segundos, límite que es menor al límite de tiempo solicitado de 0,3 segundos.. El propósito de esta prueba es la de comprobar y visualizar el despliegue de imágenes en ambas pantallas desde

inicio a fin, sin realizar ninguna intervención. La cámara se encuentra encendida y la generación de reportes se encuentra activa.

Resultados

El resultado de esta prueba es exitosa. Las imágenes se despliegan correctamente partiendo desde tiempos de muestra altos, llegando hasta tiempos de muestra bastante cortos. Además, se verifica que la cámara funciona correctamente visualizando imágenes a tiempo real en el PF y grabando video en todos los tiempos de muestra determinados. El reporte generado concuerda con lo medido.

5.1.2. Prueba 2

La segunda prueba consiste en mostrar imágenes y presionar palancas. Al igual que en la primera prueba, se utilizan los mismos parámetros de configuración del cuadro 5.1, realizando 10 variaciones sobre el tiempo de muestra. Por los 10 pares de muestra, se presiona la palanca izquierda en 5 ocasiones y luego la palanca derecha en las restantes 5.

Resultados

A medida que se presionan las palancas, se confirma que el tiempo de recompensa o castigo corresponde con lo que se configura y en el caso de error, se emite el sonido de aviso de error. La cámara funciona correctamente y la generación de reportes funciona como es debido.

5.1.3. Prueba 3

La tercera prueba, implementada con las configuraciones del cuadro 5.1, es una prueba de estrés al sistema. Durante la muestra de imágenes en pantalla, se presionan ambas palancas a la vez, llegando incluso a presionarlas más de una vez por par de estímulos y también dejando presionada una o ambas palancas por más del tiempo que puede esperarse de respuesta. Esta prueba se realiza en base a que es probable que el comportamiento en algún roedor no sea el esperado e incluso durante su fase de entrenamiento, no realice los ejercicios como se está contemplado forzando al sistema a respuestas a las cuales no está diseñado, es decir, esta prueba es una prueba de robustez de sistema.

Resultados

Los resultados obtenidos son favorables. Al presionar ambas palancas a la vez, el sistema detecta la diferencia mínima de tiempo entre la presión de una u otra palanca. Esto se debe a que el sistema monitorea ambas palancas cada 10 milisegundos. Cuando una o ambas palancas son presionadas y no se liberan, el programa permanece en estado de espera hasta que ambas palancas sean liberadas. Esto permite que el roedor no tenga ningún incentivo para permanecer sobre una palanca. La cámara funciona correctamente, al igual que la generación de reportes.

5.1.4. Prueba 4

Parámetro	Valor
Tiempo de Muestra	1 - 0.3 segundo
Tiempo Correcto	3-5 segundo
Tiempo Incorrecto	2 segundo
Tiempo Listo	1-2 segundo
Cantidad de muestras	20 pares
Cantidad de pellets	1-3 pellet

Cuadro 5.2: Parámetros de prueba - Prueba 4

La última y cuarta prueba consiste en simular una situación lo mas real posible. La configuración de parámetros es como se muestra en el cuadro 5.2. Se utilizan 20 pares de estímulos con un tiempo de recompensa que varía entre 3 y 5 segundos dependiendo de la cantidad de pellets que se entreguen, que va desde 1 pellet hasta 3 pellets. La cámara se utiliza normalmente al igual que la generación de reportes. Además, se prueba el botón ALIMENTO en el PF.

Resultados

Las imagenes se despliegan correctamente mientras se escoge una palanca por par de estímulos. Cuando se contesta correctamente, la recompensa se entrega correctamente en el receptáculo de pellets y cuando la respuesta es incorrecta, el sonido que advierte error se emite correctamente. El botón ALIMENTO entrega la cantidad configurada en el PF correctamente. La cámara

y generacion de reporte funcionan correctamente.

5.2. Conclusión de pruebas

Las pruebas muestran robustez del sistema, actuando correctamente ante situaciones que podrían considerarse normales y otras anormales. Se considera posible el uso del sistema con roedores, esperando conseguir resultados consistentes y fidedignos.

Capítulo 6

Mejoras potenciales al proyecto

Como se mencionó en un comienzo, los alcances de este proyecto de memoria están acotados por los requerimientos generales y temporales, tanto de la investigación en la cual se trabaja, como por el tiempo previsto para el desarrollo de este proyecto de memoria.

Existen variables que pueden ser consideradas para ser incluidas en este proyecto de memoria, pero que por incompatibilidades temporales han sido solo consideradas las variables primordiales para el desarrollo de este proyecto, dejando estas otras de orden secundario como desarrollo complementario.

Específicamente, el módulo de cámara tiene bastante potencial para incluir otras aplicaciones a tiempo real que podrían ayudar mucho en el transcurso de esta investigación. En algunas de estas potenciales mejoras se realizaron avances, mientras que en otras solo se plantea la problemática y posible solución. Estas aplicaciones se explican a continuación.

6.1. Detección de proximidad del roedor a las pantallas

Esta aplicación se desarrolla en base a la maqueta construida, lo que ofrece una ventaja, ya que puede probarse antes de ser incorporada a la caja de análisis definitiva. Esta función permite condicionar el actuar de los roedores, incitándolos a acercarse a las pantallas o palancas entregándoles una recompensa cada vez que lo realicen. Este aspecto puede ayudar mucho en la etapa inicial de entrenamiento de los roedores, acortando el tiempo de adaptación, permitiendo

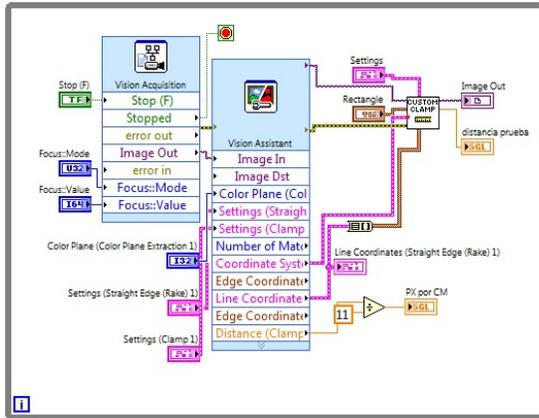


Figura 6.1: Detección de proximidad

llegar a los objetivos centrales de la investigación en menos tiempo. El diagrama de bloques programado, como se puede ver en la Figura 6.1, se compone principalmente de dos bloques, los cuales son *Vision Aquisition* y *Vision Assistant* para adquirir y manipular imágenes respectivamente. Se generan controles e indicadores en el panel frontal para modificar y visualizar características en la adquisición y procesamiento de imágenes.

El bloque principal dentro de este módulo es *Vision Assistant* que se configura para realizar ciertas funciones en secuencia según como se ve en la Figura 6.2.

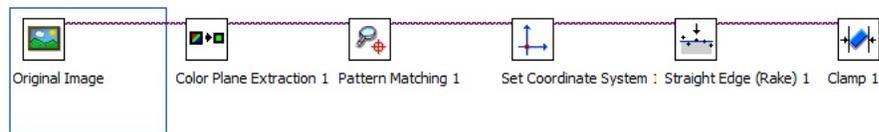


Figura 6.2: Vision Assistant

- Color Plane Extraction**

Se toma la imagen recibida del bloque *Vision Aquisition* y se extrae el plano que se requiera, ya sea en formato RGB o HSV. Esto se debe a que las funciones siguientes utilizan imagenes de 8bit por pixel en vez de 32bit, que es como se adquiere la imagen de entrada. Al extraer el plano, se pasa de una imagen de 32bit a 8bit.

- Pattern Matching**

Este bloque se utiliza para detectar un patrón previamente definido en una región de interés. Este patrón debe tener características bien definidas para no permitir errores de detección, ya que es una función primordial para los siguientes pasos.

- **Set Coordinate System**

Para otorgar un sistema de referencia dentro de la imagen adquirida, se define un sistema de coordenadas utilizando el centro del patrón encontrado en el paso anterior como coordenada (0,0).

- **Straight Edge**

Ya conociendo la ubicación espacial del plano mostrado en pantalla, se ubica un eje de referencia en un área acotada. El eje de referencia escogido es el borde entre las palancas de decisión y el piso de la caja contenedora. Este borde o eje encontrado es referencia para medir distancia entre las pantallas y el roedor.

- **Clamp**

Finalmente, se tienen coordenadas de referencia gracias a un patrón reconocido y un eje de referencia para medir distancia. Este último bloque mide la distancia, en píxeles, de un borde de pantalla a la otra. Esto permite transformar, fuera de *Vision Assistant*, de píxeles a metros, ya que la distancia entre pantallas es un dato conocido.

Una vez realizadas estas funciones, se crea una subVI utilizando el bloque *Clamp* como guía y se renombra como *Custom Clamp*. Este bloque mide la distancia entre el eje de referencia establecido anteriormente y cualquier elemento que se presente en un área de interés. Finalmente, es esta distancia la que puede usarse como referencia a un interruptor ON/OFF para el sistema presentado.

6.2. Determinación de nivel de estrés del roedor

Es normal esperar que el roedor examinado se sienta estresado en las primeras sesiones de medición dentro de un ambiente nuevo, lo que se traduce directamente en la liberación de orina y/o fecas las cuales generalmente son contabilizadas y registradas para medir el nivel de estrés del roedor. Actualmente este es un proceso manual que se lleva a cabo directamente por el

examinador quien debe estar observando atentamente el desarrollo de la sesión. Esta función no fue desarrollada como la anteriormente descrita por haber surgido como posible mejora al final del desarrollo de este proyecto y por ser irrelevante para el correcto desarrollo de una sesión de medición.

El factor limitante en la incorporación de estas aplicaciones del módulo de cámara al sistema general, es la disposición de la caja contenedora del ambiente de análisis, que debe tener ciertas dimensiones en base a los requerimientos técnicos de la investigación.

6.3. Proyecciones futuras para este proyecto

El desarrollo de este proyecto está orientado a cubrir una necesidad particular de una investigación mayor. Sin embargo, es posible ampliar el rango de aplicaciones de este sistema. A continuación, se describen distintas proyecciones para este trabajo:

- Opciones de configuración de hardware

Permitir incluir mas pantallas y palancas de forma modular como se estime conveniente, puede ser un gran atractivo para quien considere trabajar con este sistema. Esto permitiría realizar otros tipos de pruebas conductuales.

- Mejoras en interfaz gráfica

Si se pretende distribuir este sistema, es necesario contar con una interfaz gráfica de mejor calidad, ofreciendo mas opciones de configuración para abarcar mayor número de variantes que puedan ser de interés para los usuarios.

- Control online

Para agregar valor al proyecto realizado, se podría considerar incluir control online. Ingresar a través de una página web que ofrezca una visualización del panel frontal de la aplicación, permitiría coordinar pruebas sin la necesidad de estar en el lugar. Además de agregar un enlace a la aplicación por web, sería necesario contar con una estructura de cajas contenedoras para los roedores y poder optar a seleccionar a cada roedor en particular para realizar alguna medición. Esta funcionalidad es de complejo desarrollo e involucra un desafío considerable, pero que otorgaría gran valor al sistema elaborado.

Las proyecciones para este proyecto pueden ser bastantes si se consideran necesidades particulares, lo que lleva a incontables formas de mejoras o proyecciones para este proyecto. Todas cuales permitirían automatizar procesos que actualmente se desarrollan de forma manual o con bajo grado de automatización, disminuyendo la cantidad de horas/hombre dedicadas, aumentando productividad.

Capítulo 7

Conclusiones

Los avances tecnológicos han permitido simplificar procesos, minimizando tiempo y recursos. Esta aplicación creada específicamente para el estudio comportamental de roedores, permite minimizar tiempo y trabajo al recopilar automáticamente resultados. La muestra de estímulos y el recuento automático de respuestas integra distintas ramas de la electrónica. Partiendo por el hardware utilizado, se implementa un sistema de transmisión efectivo entre pantallas y computador, utilizando una tarjeta NI 6221 como interfaz. Esta misma interfaz es quien recopila los resultados adquiridos a través de la interacción del roedor con el sistema y finalmente, se genera una respuesta conducida a través de esta misma interfaz, la tarjeta adquisidora. En conjunto con este canal de comunicaciones bidireccional, tarjeta adquisidora, se tiene un software de control. Este software de control es el cerebro del sistema que envía, recibe y genera datos. Los datos son enviados a los dispositivos periféricos, ya sea pantallas, parlantes y otro, dejando la puerta abierta a incluir otros dispositivos según sea necesario, pudiendo ampliar las facultades de este sistema. Los datos recibidos son datos análogos, respuestas físicas transformadas a lenguaje entendible para el software de control, los cuales son manipulados para generar respuestas asociadas a valores correctos o incorrectos. Este software elaborado a medida, permite una visualización a tiempo real de cada variable relevante para llevar el control en cada sesión.

Una vez valorizados estos datos, se envían datos que generan acciones en el sistema. Los datos recopilados y generados, son guardados para posteriores análisis. Entonces podemos identificar un cerebro, el software de control; un canal de comunicaciones, la tarjeta adquisidora y satélites

o dispositivos periféricos como pantalla, dispensador de alimento, entre otros.

La particularidad de esta estructura definida radica en la potencial expansión de capacidades ingresando otros dispositivos según sea necesario. O también la inclusión de nuevas funciones a los satélites ya utilizados, como se menciona por ejemplo en las potenciales funciones de la cámara utilizada.

La herramienta para elaborar el software de control, LabVIEW, demostró ser una potente herramienta de desarrollo, versátil e intuitiva. Si bien no es un programa de libre acceso, si se tiene suficiente soporte en internet para resolver dudas y aprender nuevas funcionalidades. Cabe mencionar la utilidad del foro de soporte de National Instruments [7] en el cual se encontró valiosa información para solucionar problemas y optimizar soluciones. Por lo tanto el conjunto hardware y software escogido para desarrollar este sistema cumple con las expectativas y se puede dar por concluido el desarrollo de este proyecto de memoria.

Capítulo 8

Anexos

8.1. Anexo A

En la siguiente sección se presentan los cuadros del Diagrama de bloques del programa de control desarrollado. Estos diagramas se encuentran unidos en un orden secuencial, como se indica en cada figura.

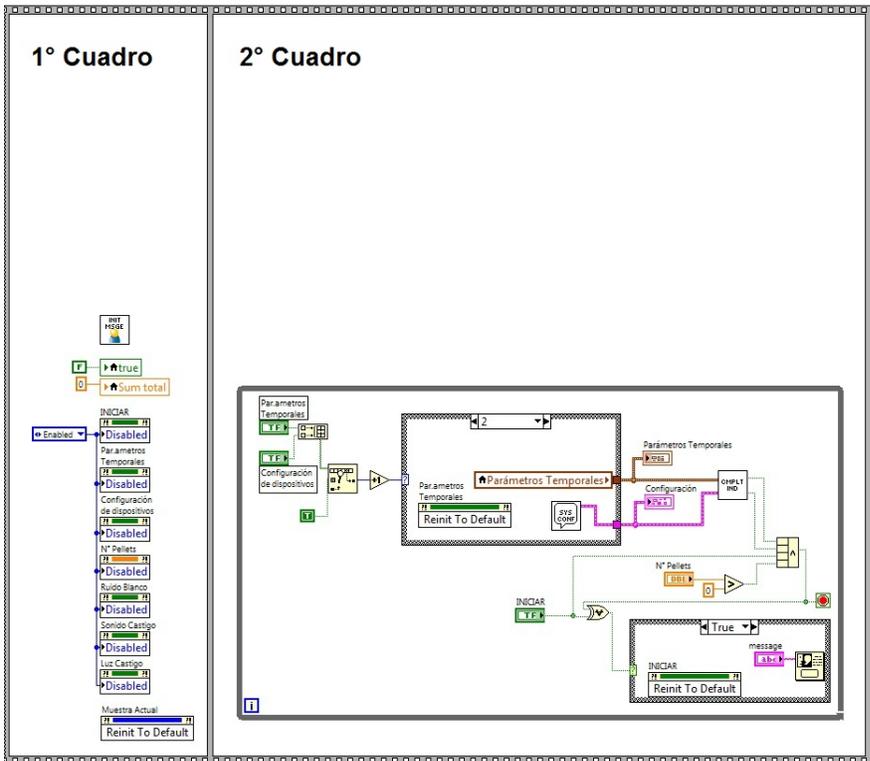
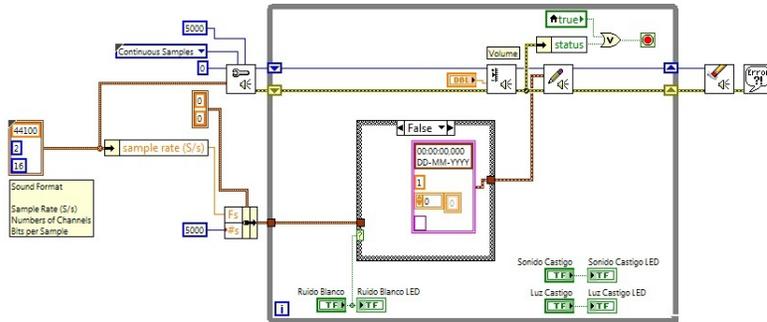


Figura 8.1: Diagrama de bloques - Primer y Segundo Cuadro y Diagrama de Sonido

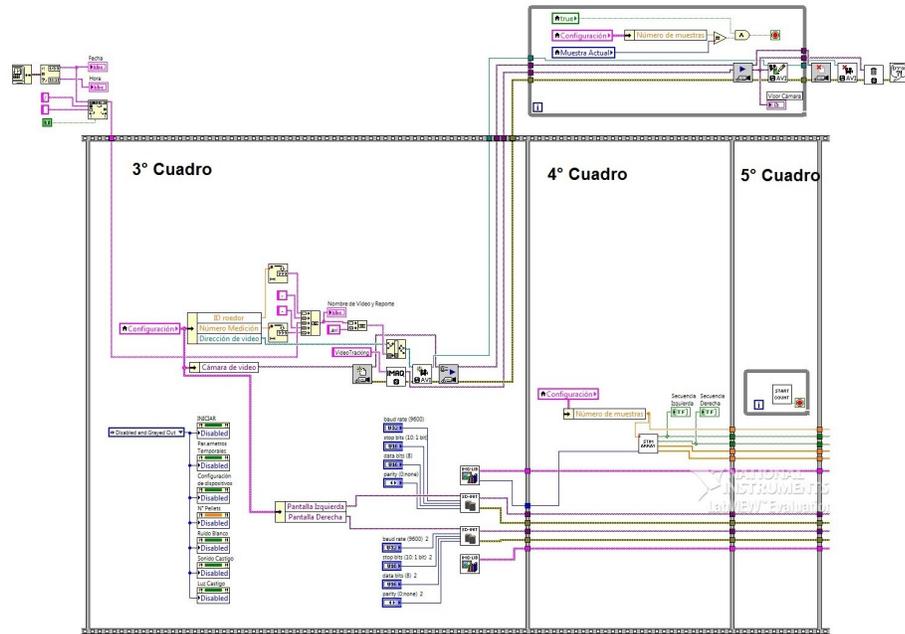


Figura 8.2: Diagrama de bloques - Tercer, Cuarto y Quinto cuadro y Diagrama de Video

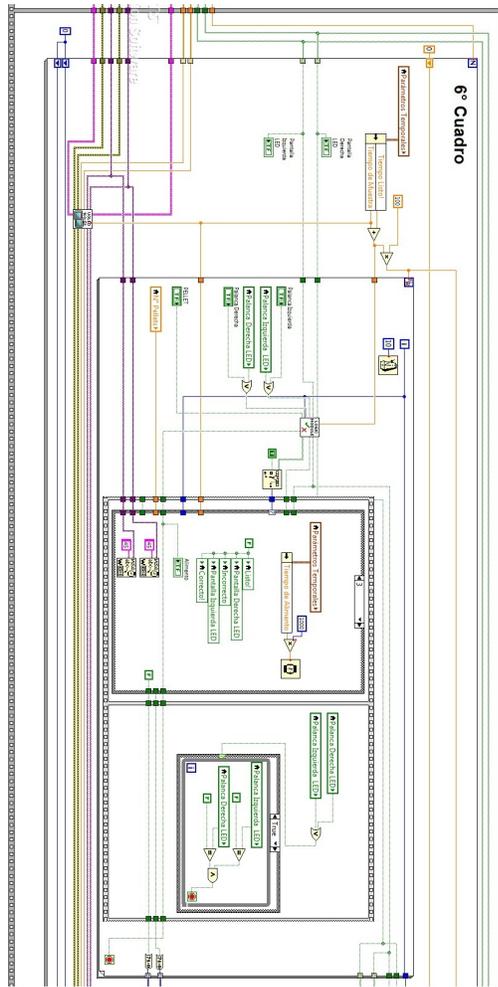


Figura 8.3: Diagrama de bloques - Sexto Cuadro (Incompleto)

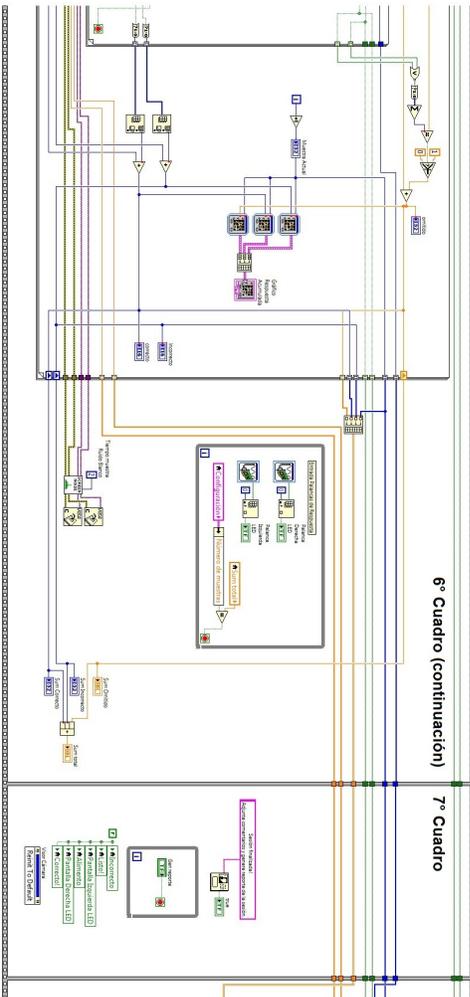


Figura 8.4: Diagrama de bloques - Continuación Sexto Cuadro y Séptimo Cuadro

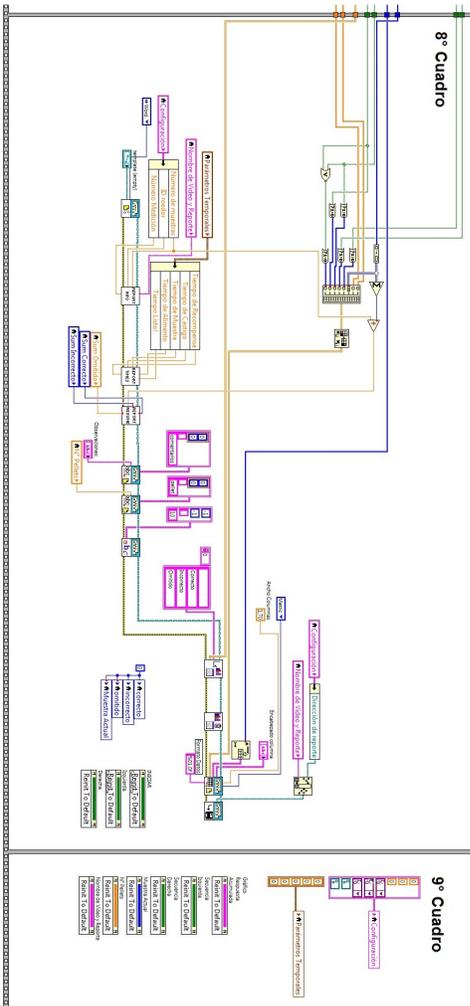


Figura 8.5: Diagrama de bloques - Octavo y Noveno Cuadro

8.2. Anexo B

En el Diagrama de bloques del programa de control, se utilizan bloques, o subVI's, que realizan diversas funciones. La mayoría de estos bloques son propios de LabVIEW, mientras que una menor parte fue creada para cumplir con funciones específicas, proporcionando mayor orden estructural del código programado.

A continuación se detalla cada uno de los bloques, subVI, generados para cada cuadro del diagrama de bloques.

8.2.1. Primer Cuadro

init message.vi

Este subVI permite mostrar un mensaje inicial, Figura 8.6, dando la bienvenida y ofreciendo la opción de obtener ayuda en caso de no conocer el programa o de simplemente continuar con una sesión de medición.

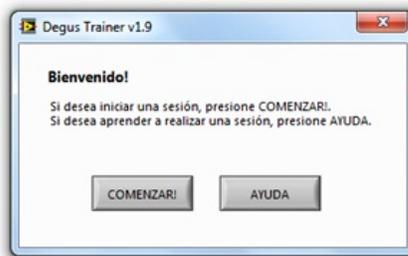


Figura 8.6: Panel frontal - init message.vi

Como se puede ver en la Figura 8.7, este bloque no posee entradas ni salidas, solo se ejecuta y luego de presionar uno de los dos botones, su ejecución finaliza. Esta finalización de ejecución concluye abriendo un documento pdf (Instrucciones de uso, Anexo F) y avanzando al siguiente cuadro de ejecución o solo avanzando al siguiente recuadro de ejecución.

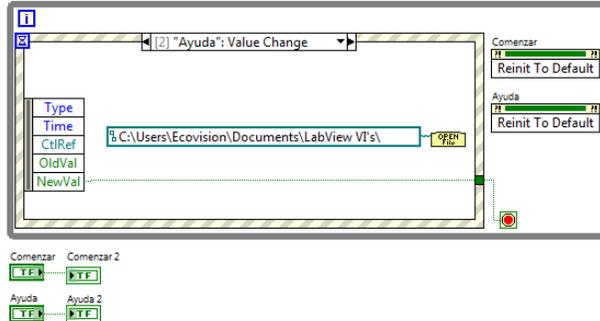


Figura 8.7: Diagrama de bloques - init message.vi

8.2.2. Segundo Cuadro

temporal parameters.vi

Para configurar los parámetros temporales a utilizar en la sesión de medición, es necesario introducirlos manualmente antes de dar inicio. Este subVI permite almacenar tales valores al desplegar una ventana que permite interacción con el usuario. Este bloque no tiene entradas a nivel de diagrama de bloques, solo salida, la cual es un cluster de 5 elementos.

El diagrama de bloques de este subVI se muestra en la Figura 8.8. Este subVI se compone de un *Case Structure* dentro de un *While Loop*. En total son tres casos los que se consideran, los cuales son: `OK Button:ValueChange`, `Panel Close` y `Cancel:ValueChange`. El caso importante es el primero, Figura 8.8, el cual permite el ingreso de los parámetros temporales que se utilizarán en esa sesión de medición.

Los dos casos restantes conectan una constante booleana `TRUE` al botón `STOP` dentro del *While Loop*, lo que ocasiona el término de la ejecución de `temporal parameters.vi`.

system configuration.vi

La configuración de los dispositivos y otros parámetros, se lleva a cabo a través de este subVI que permite introducir manualmente tales valores. Este bloque no tiene entradas a nivel de diagrama de bloques, solo salida, la cual es un cluster de 8 elementos.

Al igual que el subVI anterior, este subVI también presenta un *Case Structure* dentro de un *While Loop*. Se tienen 3 casos, de los cuales solo es relevante el que se muestra en la Figura 8.9,

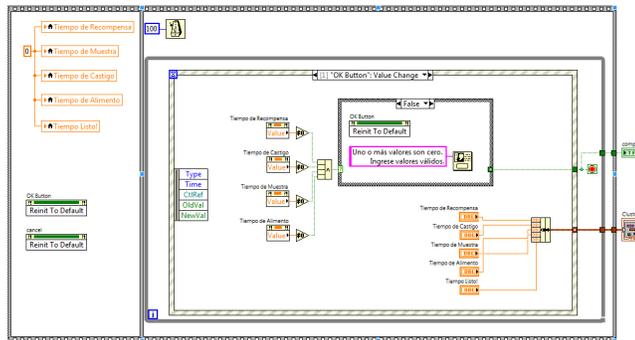


Figura 8.8: Diagrama de bloques - temporal parameters.vi

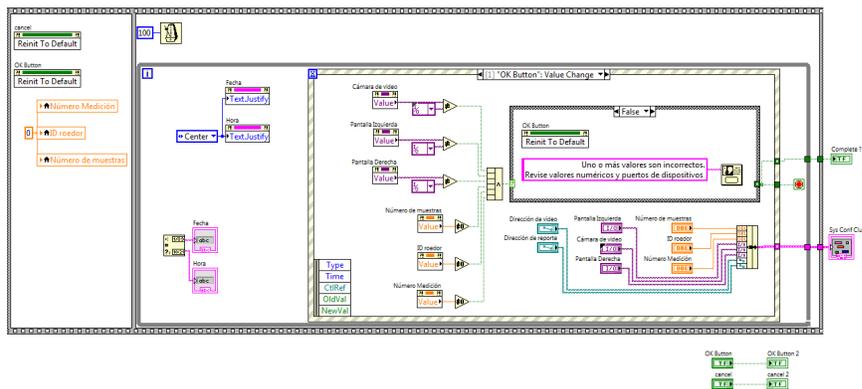


Figura 8.9: Diagrama de bloques - system configuration.vi

que permite el ingreso manual de las variables de medición. Mientras que los dos casos restantes, solo conectan una constante booleana TRUE al botón STOP del *While Loop*.

complete indicator.vi

Para comprobar que todos los valores ingresados son no nulos, esta subVI realiza una comparación lógica para cada uno de los valores. Las entradas de este bloque, Figura 8.10, son el cluster de salida de *temporal parameters.vi* y el cluster de *system configuration.vi*. La salida de este bloque es un valor booleano, indicando si ambos cluster cumplen con poseer valores válidos o no.

En caso de que alguno de los valores sea nulo y se intente seguir con la ejecución de la

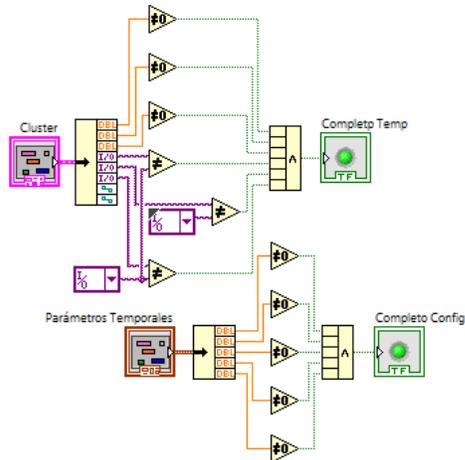


Figura 8.10: Diagrama de bloques - complete indicator.vi

sesión de medición, se despliega una pantalla de aviso advirtiendo esta situación que puede ser corregida para luego proceder correctamente.

8.2.3. Tercer Cuadro

image library.vi

La librería de imágenes es el repertorio de estímulos visuales que pueden mostrarse en pantalla. Estas imágenes se identifican con un *string*, el cual es la ubicación específica dentro de la memoria microSD en el módulo lector de memoria de la pantalla uOled. Este bloque, Figura 8.11, no posee entrada a nivel de diagrama de bloques, solo dos salidas. Un arreglo de *string* con las direcciones de cada una de las imágenes disponibles y el tamaño del arreglo, indicando la cantidad de imágenes disponibles en el repertorio.

Este subVI alimenta al bloque UOled 160G1.vi, presente en el cuadro 6 de ejecución.

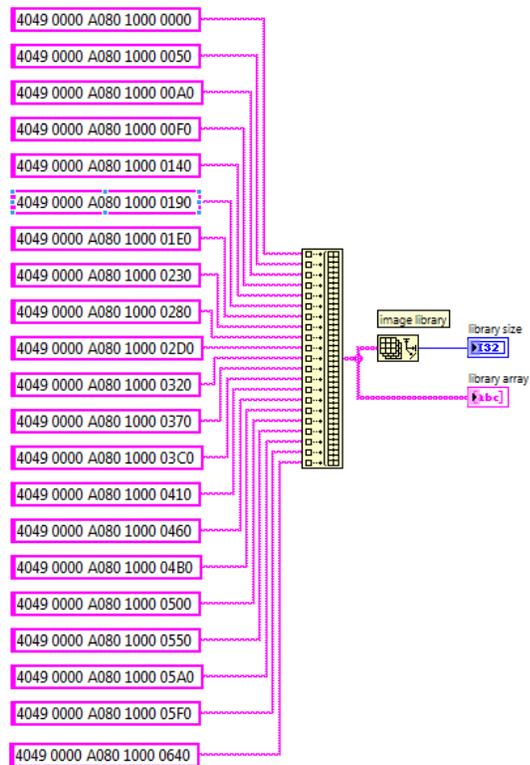


Figura 8.11: Diagrama de bloques - image library.vi

SD init.vi

La memoria microSD debe ser inicializada antes de utilizarse. Esto permite la correcta conexión del módulo lector con el computador. Para inicializar la memoria microSD, se debe enviar secuencialmente tres comandos. Estos comandos hexadecimales son:

1. 55: Comando para configurar automáticamente el **baud rate** entre módulo pantalla y computador.
2. 4060: Comando inicializador de memoria microSD.
3. 45: Para asegurar el inicio de mediciones con una pantalla limpia, sin imágenes previas.

Para enviar secuencialmente estos comandos, primero se configura el canal serial utilizado y se construye un arreglo con los comandos hexadecimales. Estos datos ingresan a un *While Loop* que envían secuencialmente los datos al puerto serial correspondiente. Este diagrama de bloques puede verse en la Figura 8.12.

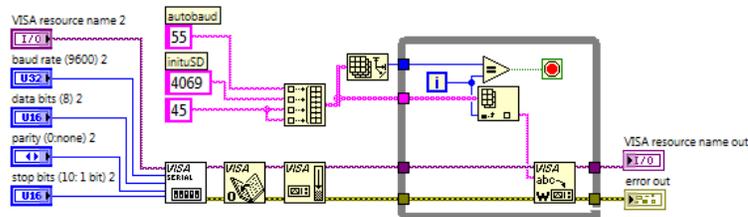


Figura 8.12: Diagrama de bloques - sd init.vi

8.2.4. Cuarto Cuadro

stimulus array.vi

Para mostrar los estímulos en pantalla, se necesita configurar la forma en que estos serán mostrados. Estos pueden ser mostrados en una secuencia aleatoria, como también mostrados en una secuencia determinada por el usuario. Para escoger entre una alternativa o la otra, una ventana de diálogo se muestra en pantalla presentando dos botones, ALEATORIA y MANUAL, ambos vinculados a un *Case Structure*. En total, se tienen tres casos. Estos corresponden a un caso por cada botón mas un caso adicional cuando la pantalla es cerrada.

El diagrama de bloques del subVI `stimulus array.vi` cuenta con un *Case Structure* dentro de un *While Loop*. Los valores de entrada a esta subVI son a través del la ventana de diálogo al usuario y dos valores procedentes de otros subVI. Estos dos valores son NUMERIC y LIBRARY SIZE. Más adelante será explicada la utilidad de cada valor.

1. Caso 0 (Default)

El primer caso, Caso 0 (o caso por defecto) en la Figura 8.13, es invocado cuando la pantalla que permite escoger una configuración de secuencia aleatoria o manual es cerrada. Esta acción ejecuta este caso, ejecutando a la vez el subVI `random.vi`, que configura una

secuencia de estímulos de forma aleatoria.

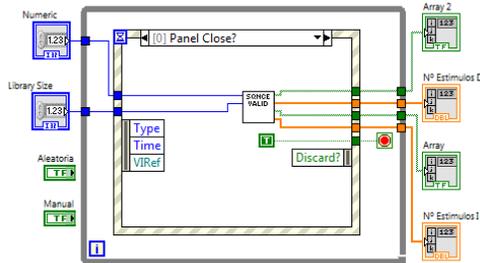


Figura 8.13: Diagrama de bloques - stimulus array.vi (caso 0 (Default))

En el diagrama de bloques de `random.vi`, Figura 8.14, ingresan como parámetro la cantidad de estímulos (NUMERIC) y el tamaño de la librería de imágenes (LIBRARY SIZE). La cantidad de estímulos se utiliza para ejecutar un ciclo FOR tantas veces como pares de estímulos sean necesarios, mientras que el tamaño de la librería de imágenes se utiliza para determinar cual imagen se asigna a cada par de estímulos al escoger un número de forma aleatoria entre (1) y (LIBRARY SIZE/2) para los estímulos correctos y entre (1+(LIBRARY SIZE/2)) y (LIBRARY SIZE) para los estímulos incorrectos. Esta determinación sobre cuáles índices pertenecen a los estímulos correctos o incorrectos, se da a nivel del subVI `image library.vi`, donde se ordenan las imágenes para tal efecto.

En la ventana desplegada, el usuario puede validar la secuencia aleatoria entregada o cargar una nueva secuencia si así lo desea para luego confirmarla. Es importante contar con una carga similar de estímulos correctos/incorrectos por cada pantalla, por lo que en la ventana emergente se incluye un contador de estímulos correctos e incorrectos para mostrar al usuario la distribución de estímulos. Para lograr que la repartición entre pantallas sea similar, se implementa como regla que en cada pantalla no pueda haber mas de tres estímulos del mismo valor de forma consecutiva.

2. Caso 1 Este caso es ejecutado cuando en la ventana emergente para decidir forma de generar la secuencia de estímulos se presiona el boton ALEATORIA. Al igual que en el Caso 0, se ejecuta el subVI `random.vi`. La Figura 8.15 muestra el diagrama de bloques del Caso 1.

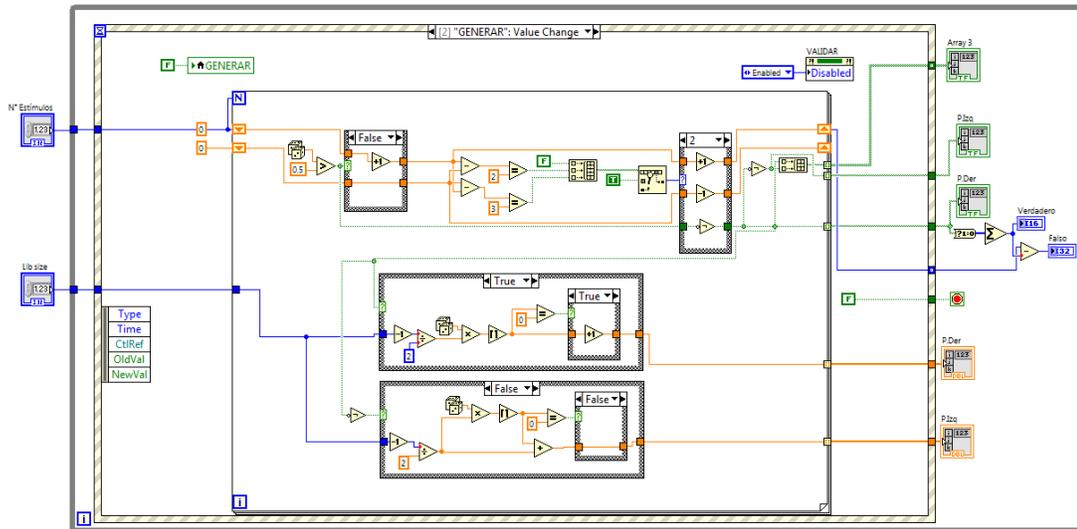


Figura 8.14: Diagrama de bloques - random.vi

3. Caso 2 Este caso, Figura 8.16, se permite la configuración manual de la secuencia de estímulos ejecutando el subVI `manual.vi`.

La ventana de interacción con el usuario permite ingresar el número de imagen a mostrar y seleccionar cual pantalla muestra el estímulo correcto/incorrecto. Es necesario determinar todos los pares de estímulos para poder continuar.

El diagrama de bloques cuenta con tres cuadros de ejecución. El primer cuadro reinicia el cluster que aparecerá en la ventana emergente y también se reinicia el botón ACEPTAR.

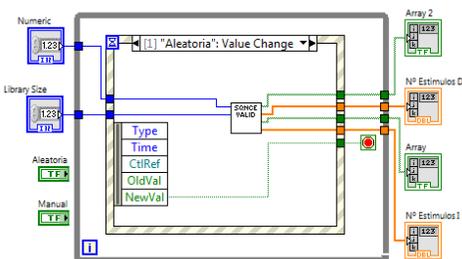


Figura 8.15: Diagrama de bloques - stimulus array.vi (caso 1)

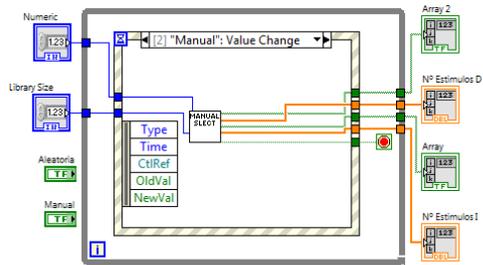


Figura 8.16: Diagrama de bloques - stimulus array.vi (caso 2)

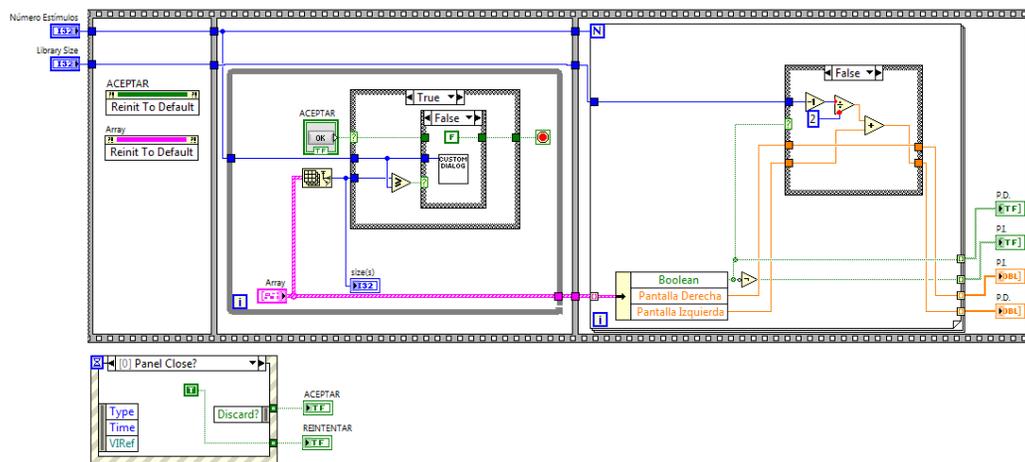


Figura 8.17: Diagrama de bloques - manual.vi

El segundo cuadro abre la ventana emergente y recopila los datos ingresados por el usuario hasta que se presione el botón ACEPTAR. Mientras que el cuadro final, toma este cluster y separa los valores para darles el formato apropiado para entregarlos como salida de este subVI.

8.2.5. Quinto Cuadro

start count.vi

Antes de comenzar una sesión de medición, se muestra en pantalla un contador regresivo de tres segundos, Figura 8.18. El objetivo de este contador es preparar al usuario para comenzar y de separar la configuración de parámetros del inicio de la muestra de estímulos.

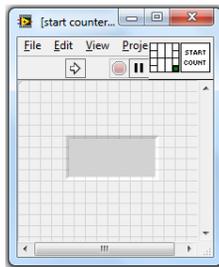


Figura 8.18: Panel frontal - start count.vi

El diagrama de bloques de este subVI es bastante simple como se puede ver en la Figura 8.19. Es un *While Loop* que contiene en su interior un *Case Structure* y que invoca cada caso con el contador interno del *While Loop*. Este *While Loop* está ralentizado a una frecuencia de 1 segundo por iteración, lo que permite que pase de caso a caso cada 1 segundo. El ciclo culmina cuando en pantalla se muestra la señal GO! al cuarto segundo, que corresponde a la cuarta ejecución del *While Loop*.

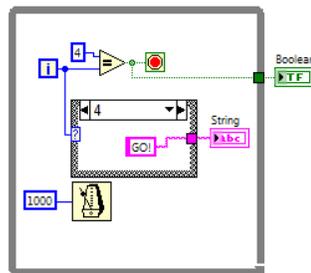


Figura 8.19: Diagrama de bloques - start count.vi

8.2.6. Sexto Cuadro

UOled 160G1.vi

Dentro de los subVI creadas, esta es la de mayor importancia. Este subVI permite la presentación de estímulos en pantalla a través de envío secuencial de datos desde el computador hacia el módulo pantalla en formato hexadecimal a través de un adaptador USB-serial. La Figura 8.20 muestra el diagrama de bloques del subVI mencionado. El diagrama se compone de una estructura principal, un *For Loop* de una iteración en el cual se incluye un *While Loop* que se ejecuta cada 0.1 segundos por (N segundos de muestra) x 10, lo que permite mostrar estímulos en pantalla N segundos. El *For Loop* permite que todas las entradas a este subVI avancen paralelamente y no existan carreras de ejecución.

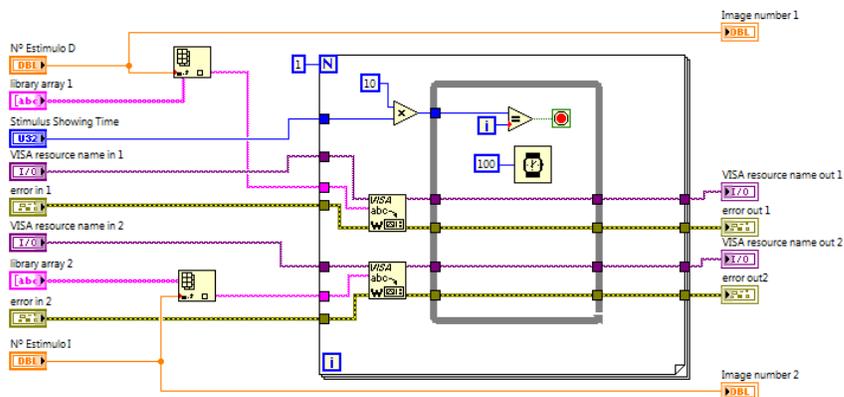


Figura 8.20: Diagrama de bloques - UOled 160G1.vi

La ejecución de este subVI comienza con la obtención de la dirección de memoria desde el arreglo entregado por `image library.vi` a través de el par de identificadores numéricos asignados a cada pantalla que ingresan secuencialmente al *For Loop* del sexto cuadro de ejecución del diagrama de bloques general. Luego, estos comandos que invocan a determinadas imágenes en la memoria microSD del módulo pantalla, son enviados a través del adaptador USB-serial. El *While Loop* mantiene la ejecución del *For Loop* el tiempo requerido de muestra de imágenes para luego terminar la ejecución de este subVI.

logic module.vi

Cada estímulo mostrado en pantalla espera una respuesta del roedor a través de las palancas. Esta respuesta, que ingresa al programa como una variable booleana, debe ser declarada como correcta o incorrecta. Este bloque realiza esta comparación entre la respuesta recibida y el estímulo mostrado en pantalla. Las entradas de este bloque son ambas palancas, censadas a través de un módulo NI-DAQmx configuradas como líneas digitales de entrada, los estados lógicos de cada pantalla, el tiempo de muestra de cada estímulo, el estado del botón PELLET y el contador del *For Loop* en el cual esta inmerso.

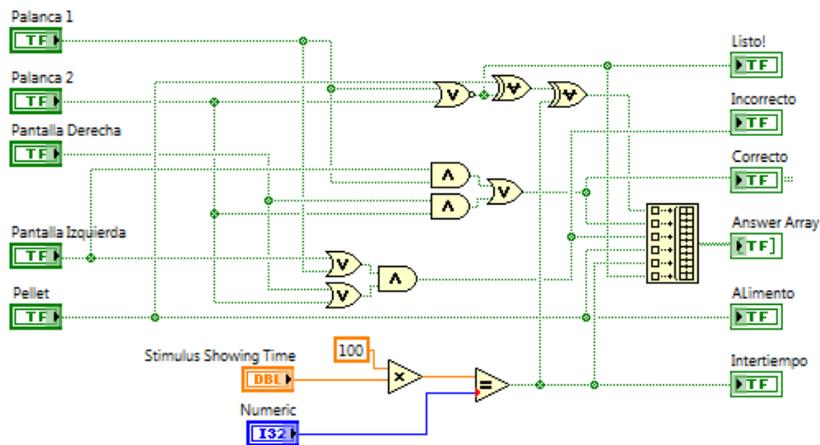


Figura 8.21: Diagrama de bloques - logic module.vi

Este diagrama de bloques, Figura 8.21, presenta una complejidad centrada en la cantidad de conexiones que se realizan entre entradas y los resultados de los operadores lógicos en cadena.

Las siguientes tablas muestran las operaciones lógicas establecidas en esta subVI para los estados *Correcto*, *Incorrecto* y *Listo*.

En las tablas de los estados *Correcto* e *Incorrecto*, ningún caso se dará simultáneamente porque son excluyentes, es decir, nunca se dará que en ambas pantallas, derecha e izquierda, exista un uno o un cero a la misma vez y lo mismo para las palancas, ya que se debe permitir que solo una palanca sea presionada a la vez y que siempre haya una pantalla con estímulo correcto y la otra con el estímulo incorrecto.

	Derecha			Izquierda			
n°	Palanca	Pantalla	and	Palanca	Pantalla	and	or
1	1	1	1	1	1	1	1
2	1	0	0	1	0	0	0
3	0	1	0	0	1	0	0
4	0	0	0	0	0	0	0

Cuadro 8.1: Lógica de estado - Correcto

	Derecha			Izquierda			
n°	Palanca	Pantalla	or	Palanca	Pantalla	or	and
1	1	1	1	1	1	1	1
2	1	0	1	1	0	1	1
3	0	1	1	0	1	1	1
4	0	0	0	0	0	0	0

Cuadro 8.2: Lógica de estado - Incorrecto

n°	Palanca Derecha	Palanca Izquierda	nor
1	1	1	0
2	1	0	0
3	0	1	0
4	0	0	1

Cuadro 8.3: Lógica de estado - Listo

El estado *Listo* solo toma en cuenta las palancas y se activa solo cuando ambas palancas estan en cero. Para el estado *Alimento*, se considera como un caso particular, ya que no tiene relación directa con las palancas ni con las pantallas, sino que solo se activa cuando se hace directamente desde la interfaz del programa de control presionando el botón ALIMENTO. Para crear el estado *Alimento*, se utiliza un OR exclusivo luego del OR o AND final en los demás estados. El estado restante, *Intervalo*, tiene la función de separar los estímulos visuales emulando un "parpadeo." entre cada estímulo visual. Este estado se crea comparando tiempos de ejecución y muestreo de imágenes.

pellet dispenser.vi

Este subVI utiliza una línea digital de salida a través de la tarjeta de adquisición NI DAQ para emitir una señal TTL (5V) para activar el dispensador de pellets y entregar recompensa. La Figura 8.22 muestra que la entrada de este subVI solo es la cantidad de pellets (Pellet N°) el cual entra multiplicado por dos a un *While Loop* y luego entra a un *For Loop* que se ejecuta tantas veces sea el este valor de entrada. Además como entrada al *For Loop* se tiene un arreglo booleano (data) que se encuentra lleno de 1 y 0 alternados, por lo tanto, la ejecución de este subVI no hace mas que enviar un uno y un cero alternadamente tantas veces sea necesario para entregar la cantidad de pellets determinado por el valor Pellet N°.

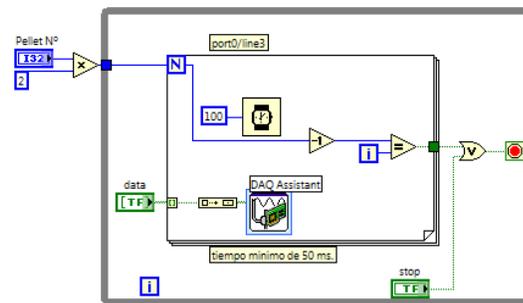


Figura 8.22: Diagrama de bloques - pellet dispenser.vi

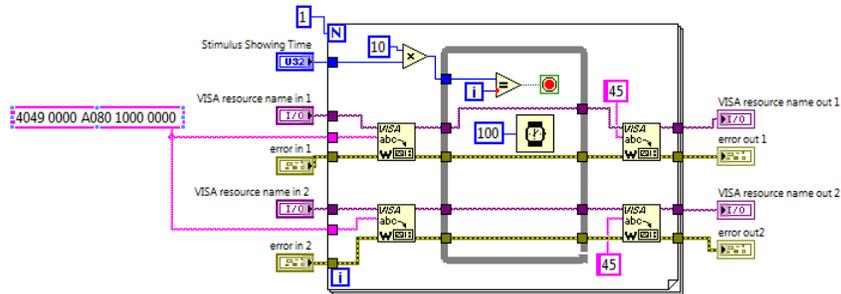


Figura 8.23: Diagrama de bloques - screen noise.vi

light pulse.vi

Al igual que el subVI anterior, `pellet dispenser.vi`, este subVI funciona a través del mismo principio, pero emitiendo un único pulso de medio segundo. Este subVI tiene como entrada el valor booleano dado por el control en el panel frontal LUZ DE CASTIGO el cual cuando es un 1 lógico, entra a un estado TRUE en un *Case Selector* emitiendo una señal de salida. Como se mencionó en el capítulo correspondiente, no se implementa esta funcionalidad en hardware por el momento.

screen noise.vi

Para demarcar la finalización de una sesión de medición, se presenta una imagen de ruido blanco. Esto permite indicar tanto al usuario cuando analice el video reporte y al roedor, el final de la sesión. El diagrama de bloques de este subVI, Figura 8.23, tiene una estructura similar al subVI `U01ed 160G1.vi` y a que se tiene un *For Loop* de una iteración, pero en este caso, solo se muestra una imagen la cual está almacenada en la primera casilla del cluster del subVI `image library.vi`.

8.2.7. Séptimo Cuadro

Este cuadro no incluye ningún subVI generado.

8.2.8. Octavo Cuadro

El octavo cuadro contiene tres subVI creados, los cuales son:

- `report info.vi`

Este subVI, Figura 8.24, escribe en el reporte los siguientes datos:

- Nombre de Video y Reporte
- Número de Muestras
- ID Roedor
- Número Medición

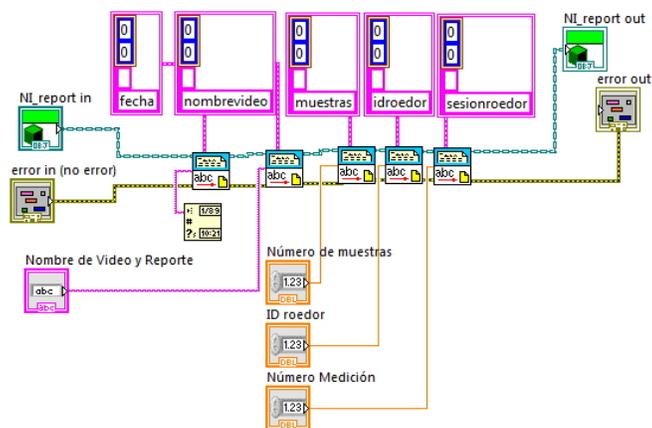


Figura 8.24: Diagrama de bloques - `report info.vi`

- `report times.vi`

Este subVI, Figura 8.25, escribe los parámetros temporales utilizados durante la sesión de medición en el reporte, los cuales son los siguientes datos:

- Tiempo de Recompensa
- Tiempo de Castigo
- Tiempo de muestra

- Tiempo de Alimento
- Tiempo Listo

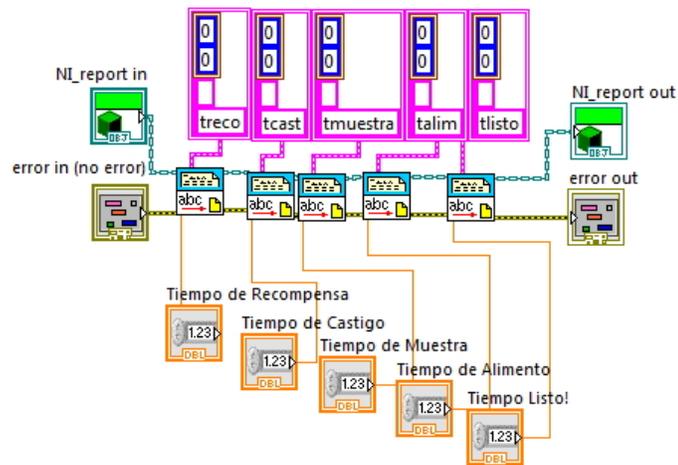


Figura 8.25: Diagrama de bloques - report times.vi

■ report resume.vi

Este subVI, Figura 8.26, inserta resultados finales de la sesión de medición cursada. Estos resultados totales son:

- Correcto
- Incorrecto
- Omitido
- Latencia

Como puede verse en cada respectiva figura, estos tres subVI se crean solo para mantener un orden en el diagrama general de ejecución, disminuyendo la cantidad total de elementos dentro del octavo cuadro.

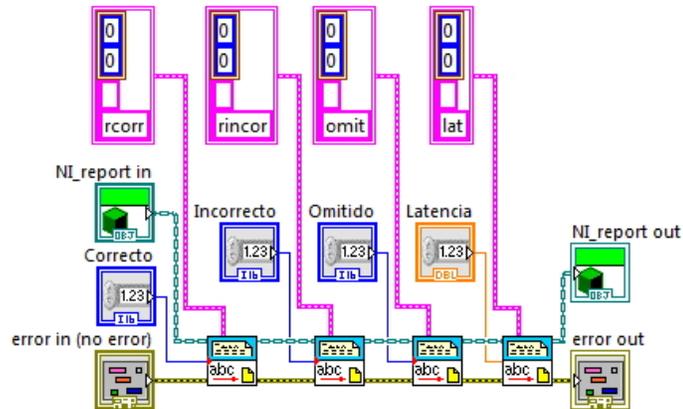


Figura 8.26: Diagrama de bloques - report resume.vi

8.3. Anexo C

Las pantallas utilizadas en el desarrollo de este trabajo, son pantallas uOLED 160-G1, fabricadas por 4D Systems. Las principales especificaciones técnicas se encuentran a continuación (En inglés).

- Low-cost OLED display graphics user interface solution.
- 160 x 128 resolution, 65K true to life colours, PMOLED screen.
- 1.7" diagonal size, 52 x 32 x 6.1mm. Active Area: 33.6mm x 27mm.
- No back lighting with near 180° viewing angle.
- Easy 5 pin interface to any host device: VCC, TX, RX, GND, RESET.
- Serial TTL interface with auto-baud feature (300 to 256K baud).
- Powered by the 4D-Labs GOLDELOX-SGC processor (also available as separate OEM IC for volume users).
- On-board micro-SD memory card adaptor for storing of icons, images, animations, etc. Supports 64MB to 2GB micro-SD memory cards.

- Comprehensive set of built in high level graphics functions and algorithms that can draw lines, circles, text, and much more.
- Display full colour images, animations, icons and video clips.
- Supports all available Windows fonts and characters (imported as external fonts).
- Multiple switch/button feature on a single pin.
- Dedicated sound pin with complex sound generation.
- 4.0V to 5.5V range operation (single supply).
- RoHS Compliant.

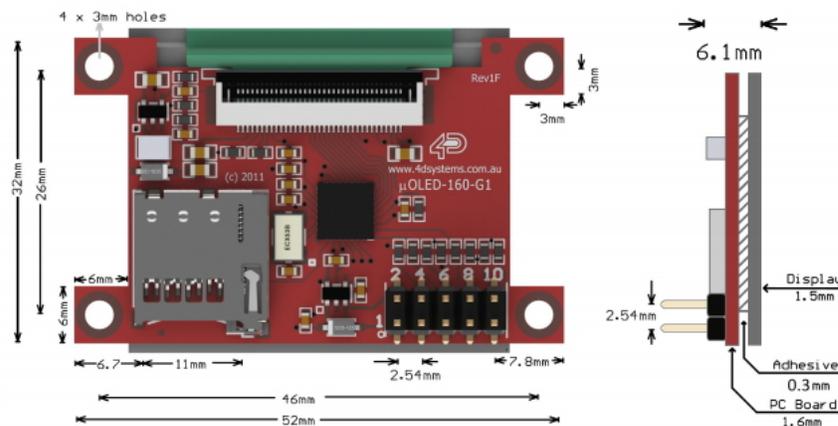


Figura 8.27: Dimensiones físicas módulo pantalla uOLED

Para conocer mayores detalles de esta pantalla, ver:

[http://www.4dsystems.com.au/downloads/Serial-Display-Modules/uOLED-160-G1\(SGC\)/Docs/uOLED-160-G1SGC-DS-rev6.pdf](http://www.4dsystems.com.au/downloads/Serial-Display-Modules/uOLED-160-G1(SGC)/Docs/uOLED-160-G1SGC-DS-rev6.pdf)

Pin	Symbol	I/O	Description
1	VCC	I	Main Voltage Supply +ve input pin. Reverse polarity protected. Range is 4.0V to 5.5V, nominal 5.0V.
2	NC	-	Not Connected.
3	TX	O	Asynchronous Serial Transmit pin. Connect this pin to host micro-controller Serial Receive (Rx) signal. The host receives data from μ OLED-160-G1(SGC) via this pin. This pin is tolerant up to 5.0V levels.
4	SOUND	O	Sound generation output pin. Connect this pin to a simple speaker circuit described in section 2.3. Leave open if unused.
5	RX	I	Asynchronous Serial Receive pin. Connect this pin to host micro-controller Serial Transmit (Tx) signal. The host transmits commands and data to the μ OLED-160-G1(SGC) via this pin. This pin is tolerant up to 5.0V levels.
6	SWITCH	I	Multi Button or Joystick switch input pin. Option is available to connect from 1 up to 5 push buttons. If connected to GND on power-up it will auto-run a script program from the memory card.
7	GND	P	Supply Ground.
8	GND	P	Supply Ground.
9	RESET	I	Master Reset signal. Internally pulled up to 3.3V via a 4.7K resistor. An active Low pulse greater than 2 micro-seconds will reset the module. If the module needs to be reset externally, only use open collector type circuits. This pin is not driven low by any internal conditions. The host should control this pin via one of its port pins using an open collector/drain arrangement.
10	3.3Vout	P	3.3V regulated output. Available current up to 50mA to power external circuitry.

I: Input, O: Output, A: Analogue, P: Power

8.4. Anexo D

Las siguientes páginas, muestran un ejemplo de reporte generado al finalizar una sesión de medición. En la primera página del reporte, Figura 8.28, se insertan todos los parámetros de configuración, una tabla resumen de resultados y el gráfico de respuestas acumuladas. En la segunda página del reporte, Figura 8.29, se incluye una tabla con los resultados totales.

DEGUS TRAINER 1.9

Reporte de medición

Fecha:	23/08/2012
ID roedor:	22
Nº sesión de roedor:	23
Nombre video registro:	22 - 23082012 - 23

Número de Muestras:	10 pares de estímulos
Tiempo de Muestra:	2 [s]
Tiempo Listo!:	5 [s]
Tiempo Recompensa:	5 [s]
Tiempo Castigo:	5 [s]
Tiempo Alimento:	5 [s]
Cantidad pellet de recompensa:	4 unidad(es)

Resumen

Respuestas Correctas:	6
Respuestas Incorrectas:	2
Omisiones:	2
Latencia promedio de respuesta:	379,100

Grafico de respuestas acumuladas

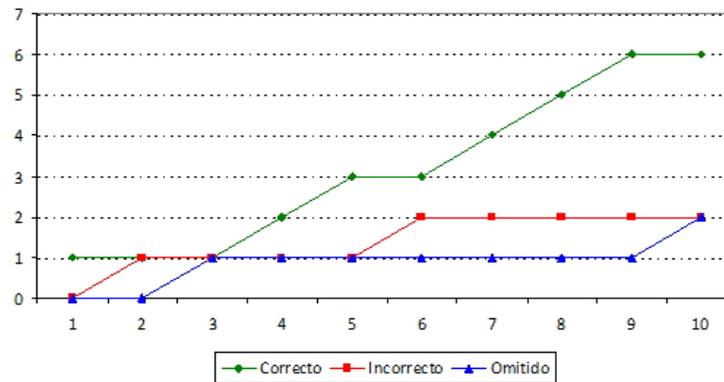


Figura 8.28: Primera página de reporte de sesión

Comentarios

Sesión de prueba terminada sin inconvenientes.

Tabla de Respuestas

	N° Imagen	N° Imagen	Latencia (ms)	Pantalla 1	Pantalla 2	Correcto	Incorrecto	Omitido
1	11	2	627	0	1	1	0	0
2	5	12	197	1	0	0	1	0
3	4	11	699	1	0	0	0	1
4	3	11	87	1	0	1	0	0
5	16	4	198	0	1	1	0	0
6	3	16	438	1	0	0	1	0
7	19	8	328	0	1	1	0	0
8	17	6	225	0	1	1	0	0
9	12	2	293	0	1	1	0	0
10	4	13	699	1	0	0	0	1

Figura 8.29: Segunda página de reporte de sesión

8.5. Anexo E

El repertorio de imágenes disponibles para mediciones puede ser reemplazado si es necesario. Para cargar nuevas imágenes en la tarjeta de memoria, se necesita utilizar el programa provisto por el fabricante de las pantallas, *Graphic Composer*. Este programa permite borrar las imágenes previamente almacenadas y reemplazarlas por un nuevo set de estímulos. El procedimiento para realizar esta acción, es el siguiente:

1. Iniciar aplicación *Graphic Composer* y visualizar pantalla inicial, Figura 8.30.

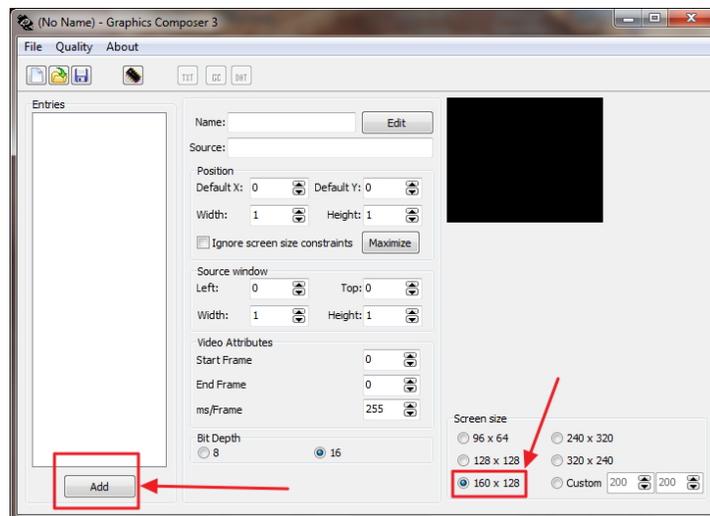


Figura 8.30: Graphic Composer - Pantalla inicial

Para ajustar el tamaño de las imágenes a almacenar, se debe seleccionar la resolución correcta, la cual en este caso es de 160x128 px. Luego, se debe adjuntar las imágenes correspondientes presionando en botón *Add*.

2. Las imágenes pueden ordenarse en la lista del costado izquierdo. También se dispone de una ventana de visualización de imágenes. Esto se muestra en la Figura 8.31.
3. Para guardar el set de imágenes creado, se debe conectar la memoria microSD que se utilizará al computador a través de un adaptador. Luego, en *Graphic Composer* se debe presionar el botón *Build*, destacado en la Figura 8.32.

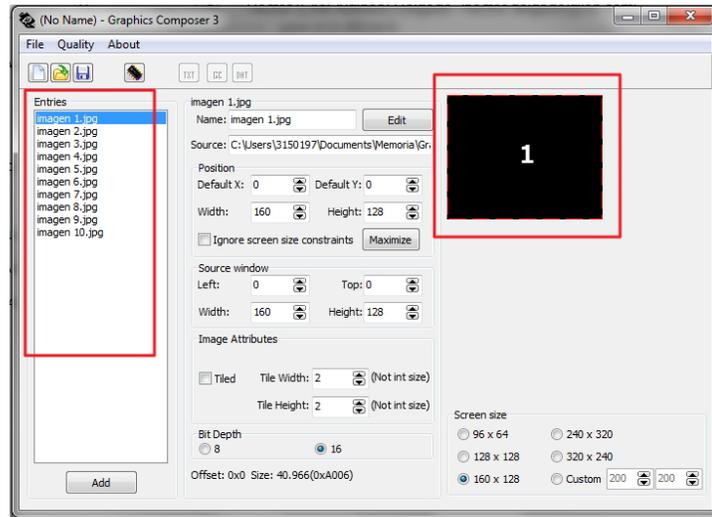


Figura 8.31: Graphic Composer - Carga y previsualización de imágenes

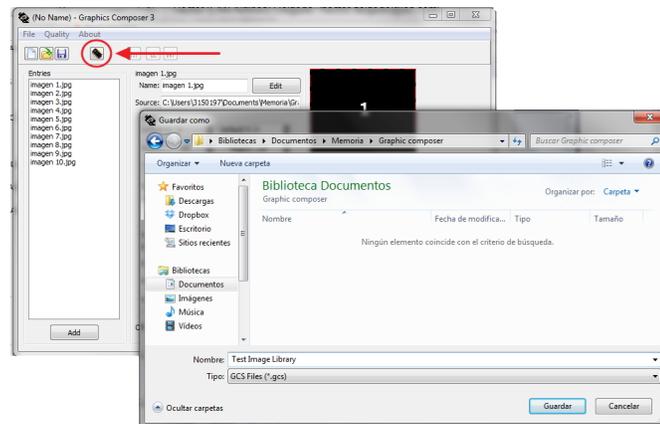


Figura 8.32: Graphic Composer - Guardado de imágenes

Luego, se despliega una pantalla que permite escoger las opciones de guardado, las cuales para este caso de uso se encuentran resaltadas con rojo en la Figura 8.33. Estas opciones contemplan el tipo de guardado, en este caso *uSD RAW*, y disco, el cual es la ubicación de la tarjeta de memoria en el computador.

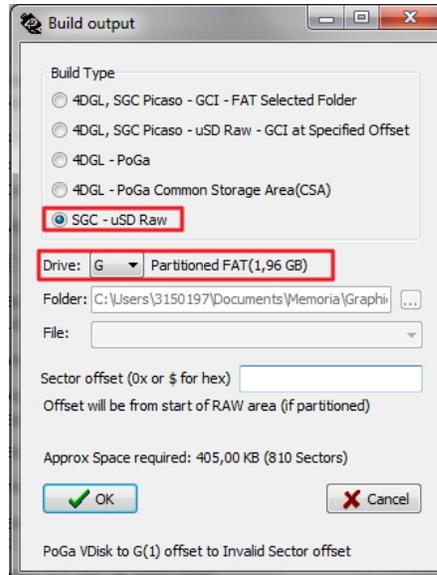


Figura 8.33: Graphic Composer - Opciones de guardado

Para terminar el guardado de imágenes, se debe presionar el botón *OK*.

4. Abrir el documento de texto generado con el botón señalado en la Figura 8.34. Este documento de texto, Figura 8.35, contiene las direcciones de memoria de cada imagen guardada, las cuales deben ser ingresadas manualmente a la subVI `image library.vi`.

En la subVI `image library.vi` se tiene un array que debe tener una entrada por cada imagen que se agregue a la librería de imágenes. Para agregar espacios en el array, se debe colocar el cursor sobre el el array y arrastrar el borde inferior como se muestra en la Figura 8.36 y para agregar cuadros para los comandos hexadecimales, se recomienda copiar y pegar alguno de los cuadros ya existentes y reemplazar texto acorde a las direcciones de las imágenes en el documento de texto, para luego unir este recuadro con un espacio vacío del array. Una vez hechos todos estos cambios, debe guardar cambios en subVI y

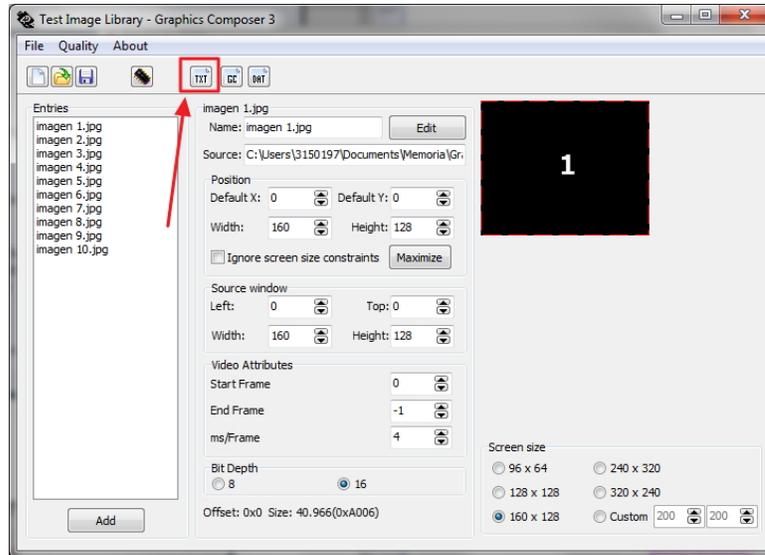


Figura 8.34: Graphic Composer - Abrir documento de texto

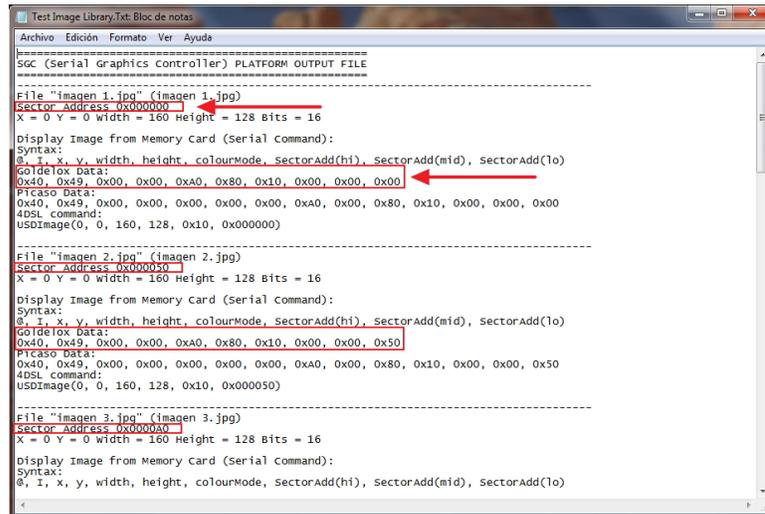


Figura 8.35: Graphic Composer - Documento de texto

cerrarlo. Finalizando estos pasos, la aplicación se encuentra lista para iniciar una sesión con un nuevo set de imágenes.

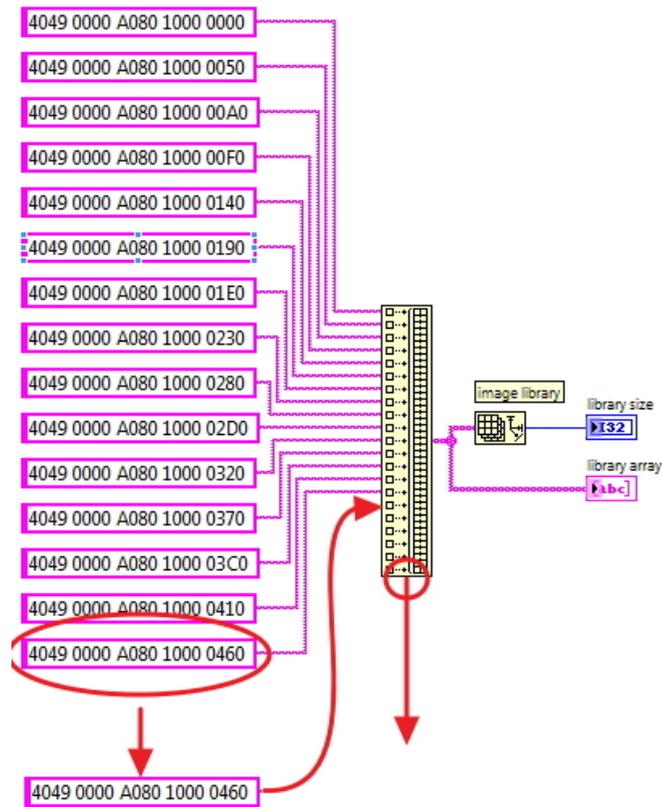


Figura 8.36: Graphic Composer - image library.vi

8.6. Anexo F

Las instrucciones de uso que se ofrecen al usuario cuando se inicia la aplicación son las siguientes.

DEGUS TRAINER 1.9

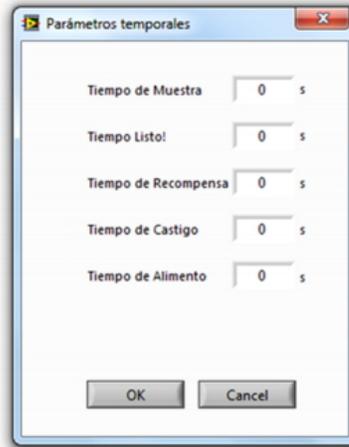
Instrucciones de uso

Este programa está especialmente diseñado para realizar mediciones de discriminación de estímulos visuales en roedores, en conjunto con la caja de estímulos visuales.

En la pantalla de este programa, dispone de todo lo necesario para controlar los parámetros de muestra y para evaluar los resultados finales.

Para iniciar una medición, debe seguir los siguientes pasos:

1. Abrir *Degus trainer 1.9.vi* desde el enlace directo en el escritorio del computador. (En caso de no encontrarlo, se ubica en C: Mis Documentos/Lab View VI's/Degus Trainer 1.6.vi)
2. La ventana de bienvenida permite abrir este documento, como también continuar con el proceso para iniciar una medición.
3. Tendrá una visión del *Panel Frontal* en el cual se presentan indicadores y controles para operar con la caja de estímulos visuales.
4. Presione el botón *Parámetros Temporales*. Se desplegará la siguiente ventana.



Debe ingresar valores numéricos, en segundos, en cada casilla según sea su criterio de medición.

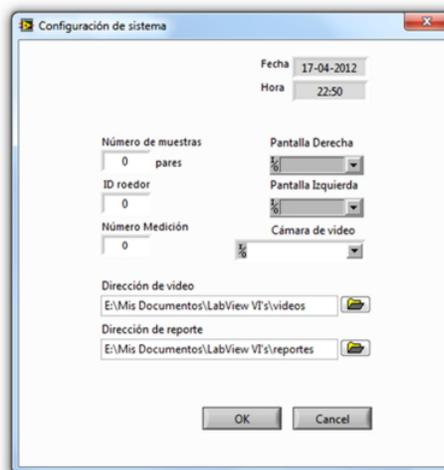
Para ilustrar brevemente a que corresponde cada tiempo, se define cada uno a continuación.

- Tiempo de Muestra: Corresponde al tiempo de presentación de los estímulos en pantalla.

- **Tiempo Listo!:** Es el tiempo que separa el término de presentación de un par de estímulos del comienzo del par siguiente. Durante este tiempo, se espera que el roedor de su respuesta a través de las palancas disponibles.
- **Tiempo de Recompensa:** Como su nombre lo dice, es el tiempo que se le da al roedor para recibir y alimentarse de su recompensa al responder correctamente.
- **Tiempo de Castigo:** Similar al tiempo de recompensa, el tiempo de castigo, es un tiempo de separación entre imágenes que se da cuando el roedor escoge erróneamente su respuesta.
- **Tiempo Alimento:** Este tiempo corresponde al intervalo que se proporciona para que el roedor se alimente luego de que el usuario a cargo presione el botón *PELLET* en el panel frontal.

Una vez ingresados todos los valores, presionar *OK*.

5. Pulse el botón *Configuración de sistema*. Se desplegará la siguiente ventana.



Debe ingresar valores válidos en cada casilla. En las casillas *Dirección de Video* y *Dirección de Reporte*, se tienen valores predeterminados los cuales pueden ser cambiados si estima necesario.

Los parámetros a completar en esta ventana, son los siguientes:

- **Número de muestras:** Es el número de pares que se mostrarán en total a lo largo de la medición.
- **ID roedor:** Cada roedor posee un número identificador de tres dígitos.
- **Número medición:** Como es probable que se realice más de una medición por roedor, este número registra el número de medición.

- Pantalla Derecha/Izquierda: Este control despliega una pantalla en la cual se debe elegir cuál es el puerto COM en el cual está conectada cada pantalla.
- Cámara de video: Dentro de las opciones disponibles de cámara, debe escoger la cámara que utilizará para esta medición.
- Dirección de Video/Reporte: Corresponde a la dirección de guardado del video y reporte generado, respectivamente.

Una vez completados todos estos campos, presionar *OK*.

Además de los campos señalados, se encuentra el indicador de fecha y hora en pantalla. Estos datos, mas *ID roedor* y *Número Medición*, se compone el nombre con el cual se guardará el video y reporte generado.

6. Defina cantidad de pellets a entregar en cada acierto en el control numérico ubicado a la izquierda del botón *PELLET*. Cabe mencionar, que esta cantidad también es la que será entregada cada vez que se presione ese botón durante la medición.
7. Si desea utilizar una señal auditiva y/o lumínica como señales correctivas, debe activarlas presionando los botones correspondientes. El indicador LED al costado derecho de cada botón, indica si estas opciones están activas o inactivas.
8. Además, puede activar o desactivar ruido blanco y controlar volumen según sea requerido.
9. Presione el botón *INICIAR*.
10. Durante la medición, se muestra el *Panel Frontal*, destacando el visor a tiempo real de la caja de estímulos y el gráfico acumulativo de respuesta.
11. Una vez terminado el proceso de medición, aparece una venta que lo señala. Presionar *OK* cierra esta ventana y permite ingresar observaciones en el recuadro que corresponde.
12. Presionar el botón *Generar Reporte*.
13. Si desea realizar una nueva medición, presione *CTRL+R* o haga clic en el botón  en el costado superior izquierdo. En caso contrario, solo cierre la ventana.

Si desea conocer más sobre la aplicación, si tiene dudas o comentarios, escriba a:

Luis Cáceres Lecaros
 luis.caceresl@gmail.com

Bibliografía

- [1] Neurobiology of vision in the retina of the diurnal rodent Octodon Degus: A Psychophysical, Multi-electrode and Computational approach. Proyecto FONDECYT 1110292
- [2] Visual adaptations in a diurnal rodent, Octodon degus G. H. Jacobs, J. B. A. Fenwick. Krogh, G. A. Williams. Received: 19 December 2002 / Revised: 18 February 2003 / Accepted: 5 March 2003 / Published online: 5 April 2003
- [3] Ruling out and ruling in neural codes Adam L. Jacobs, Gene Fridman, Robert M. Douglas, Nazia M. Alam, Peter. E. Latham, Glen T. Prusky, and Sheila Nirenberg. Communicated by David W. McLaughlin, New York University, New York, NY, January 16, 2009 (received for review September 2, 2008)
- [4] Tool-Use Training in a Species of Rodent: The Emergence of an Optimal y and Functional Understanding Kazuo Okanoya, Naoko Tokimoto, Noriko Kumazawa, Sayaka Hihara, Atsushi Iriki
- [5] The Reinagel Lab, Visual Behavior in Rodents:
<http://biology.ucsd.edu/faculty/reinagel.html>
- [6] Manual IMAQ National Instruments
- [7] National Instruments forums : <http://forums.ni.com/>