Taller de Proyecto I

Informe de Avance N°01

Arcade Stick implementado en EDU-CIAA

Carrera: Ingeniería en Computación

Facultad de Ingeniería, Universidad Nacional de La Plata

CALDERÓN Sergio Leandro, BLANCO Valentín Nicolás,

BONIFACIO Lucas Gabriel

17 de octubre de 2022

Índice

1 - Introducción

Los videojuegos son un desarrollo tecnológico acorde a nuestros tiempos; se basan en principios de competencia y están directamente relacionados con el uso de estrategia, agilidad y concentración. Los videojuegos pueden ser utilizados como herramientas educativas, en la medida en que su uso no difiere mucho de lo que se hace o podría hacerse con las nuevas tecnologías de la información.

En la mayoría de videojuegos de PC, la conexión de un controlador (joystick) por lo general es opcional, a diferencia de las videoconsolas. Sin embargo, su utilización al jugar hace más divertida la experiencia. Nuestra infancia, así como la de la mayoría de los jóvenes adultos de hoy en día, abarcó un periodo de jugar arcades, por lo que resulta interesante estudiar cómo era la realización de uno en menor escala.

De acuerdo a lo mencionado anteriormente, se plantea un Arcade Stick (joystick tipo arcade). Si bien existen muchos modelos en el mercado, éstos suelen ser fabricados solo por grandes compañías como Sega (con el Astro City), Nintendo (junto a Hori lanzando el Fighting Stick), entre otras. La motivación de este proyecto es la realización de un sistema que resulte económico, accesible y funcional, en comparación a los ya existentes de costes generalmente elevados y baja disponibilidad.

Fig. 1.1: modelo Arcade Stick vendido por Sega.

El proyecto a presentar requiere una comunicación con el sistema donde se empleará, en este caso una PC, en principio bidireccional vía USB. A partir del pulsado de botones y movimiento de una palanca, se podrá jugar videojuegos que se controlen mediante flechas direccionales (o WASD) y hasta 6 teclas adicionales.

2 – Objetivos

2.1 – Objetivos primarios

El objetivo primario de este proyecto es el diseño e implementación de un *arcade stick* funcional para computadoras. El stick tendrá una carcasa plástica y estará compuesto por 6 pulsadores NA, organizados en 2 filas, y una palanca o joystick que permita movimientos en 8 direcciones (resolución 45°). El sistema tendrá como finalidad ser un prototipo de baja latencia y adaptable a mejoras de propósito general.

El proyecto se encuentra dividido en los siguientes 4 módulos:

- Diseño y ubicación de los componentes del arcade stick.
- Calibración de la conversión de las señales analógicas.
- Implementación del código de lectura de pulsadores y joystick.
- Implementación de la comunicación con la PC.

A continuación, se presenta el diagrama en bloques del sistema:

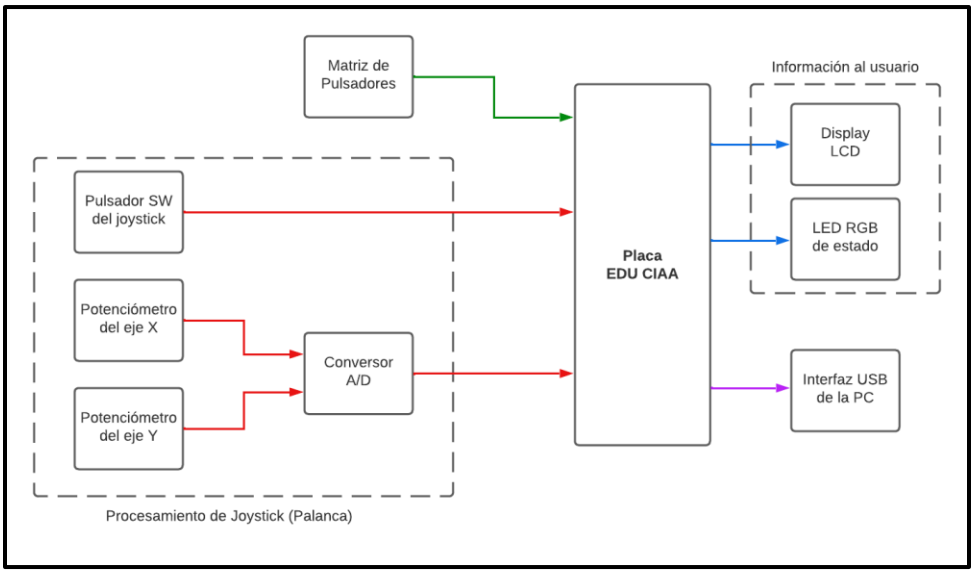


Fig. 2.1. Diagrama en bloques del sistema propuesto.

2.1.1 – Diseño y ubicación de componentes

Se conocen las dimensiones de la placa EDU-CIAA-NXP basada en LPC4337, de 137 x 86 mm [2], así como también las dimensiones aproximadas del *arcade stick* versión Astro City que se toma como base: 410 x 300 mm, grosor 170 mm [3].

Se ubicarán los 6 pulsadores, el joystick, el LED de funcionamiento y la pantalla LCD en las posiciones adecuadas siguiendo el diseño original, de modo que se aproveche el espacio disponible de manera óptima y resulte cómodo para el usuario.

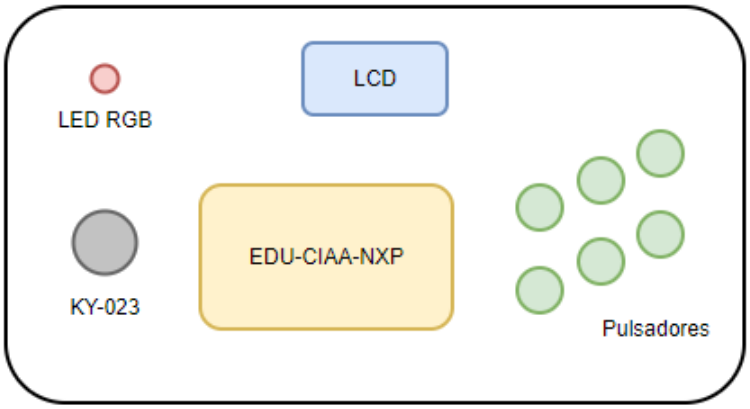


Fig. 2.2. Bosquejo de una posible ubicación de componentes.

2.1.2 – Calibración de conversión

La calibración buscará evitar la detección de un valor erróneo de dirección cuando el joystick se encuentre en reposo. Para ello se dará un margen o rango de valores a determinar, estableciendo un desplazamiento mínimo necesario del joystick para que se tome el movimiento como válido.

Además, se evaluarán las mediciones cuando el desplazamiento es máximo, para aplicar una corrección a los resultados de conversión en caso de ser necesario.

En la figura a continuación, se muestra el rango de valores discretos que toman las señales de posición de los ejes X e Y para una codificación 10 bits (de 0 a 1023).

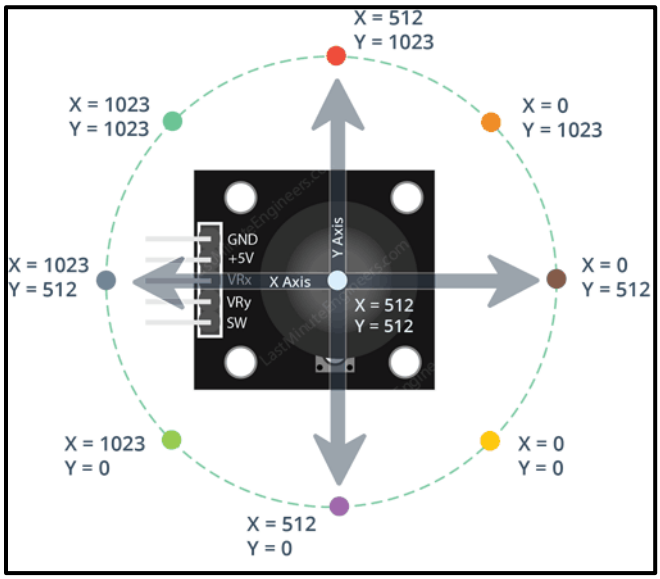


Fig. 2.3. Sistema de coordenadas de un joystick modelo KY-023. [4]

2.1.3 – Lectura de controles

En esta versión, se dispondrán de 6 pulsadores ubicados de forma matricial. Se permite que el usuario presione más de un botón a la vez, por ende, se implementará un algoritmo de lectura capaz de identificar y retornar todos los botones que se encuentren presionados, con un tiempo de procesamiento acorde al resto de tareas.

Por otra parte, para la lectura del joystick se iniciarán las conversiones y se leerán los resultados de manera periódica, debiendo traducir el par de coordenadas a alguna dirección admitida por el sistema, con previa calibración según 2.1.2. El joystick también posee un pulsador SW [5], que se considerará en la lectura de pulsadores descrita.

2.1.4 – Comunicación con la PC

Al conectar el arcade stick a una computadora, la misma proveerá de energía al sistema con una tensión de alimentación vía USB. Luego, el stick se comunicará siguiendo un protocolo a determinar para enviar uno o más códigos que representen la combinación de pulsadores presionados y dirección actual del joystick.

Se buscará en primera instancia que el microcontrolador actúe como un *game controller* nativo, similar al comportamiento de placas basadas en ATmega32u4 [6]. En su defecto, se enviarán caracteres que representen a alguna combinación vía UART, para que luego en un programa en la PC simule la presión de la tecla en el teclado.

El correcto funcionamiento de esta comunicación será informado a través de mensajes en la pantalla LCD y un color determinado en el LED RGB de estado.

2.2 – Objetivos secundarios

Se prevee agregar como futuras mejoras la posibilidad de añadir conectividad Bluetooth al joystick, permitiendo la jugabilidad mediante conexión cableada o inalámbrica. En este último caso, la alimentación sería sustituida por baterías.

También la existencia de un menú de configuración accesible mediante un botón específico del joystick para que el usuario pueda alterar el funcionamiento por defecto, con ayuda visual del LCD que muestre mensajes sobre el cambio en desarrollo.

Por otra parte, la visualización de la dirección actual de la palanca del joystick por la pantalla LCD de la placa, de modo que se pueda detectar el origen de una falla.

3 – Requerimientos

3.1 – Funcionales de información al usuario

I. Encender LED rojo cuando el sistema es iniciado.
II. Parpadear LED azul cuando la conexión con PC está en proceso.
III. Mantener encendido LED verde mientras la conexión con PC esté activa.
IV. Mostrar “Conectando…” en LCD mientras la conexión a PC esté en proceso.
V. Mostrar “Listo para jugar” en LCD cuando la conexión a PC esté activa.
VI. Garantizar que un único color del LED RGB esté encendido a la vez.
VII. Garantizar que la impresión de mensajes en LCD sea no-bloqueante.
VIII. Establecer una prioridad menor a las tareas de actualización de LED y LCD.

3.2 – Funcionales de controles

I. Establecer una frecuencia de lectura de los controles de 50 Hz mínimo.
II. Definir un rango de valores de “reposo” para los ejes del joystick.
III. Enviar dirección “UP” a la PC mientras el joystick está hacia adelante.
IV. Enviar dirección “DOWN” a la PC mientras el joystick está hacia atrás.
V. Enviar dirección “LEFT” a la PC mientras el joystick está hacia la izquierda.
VI. Enviar dirección “RIGHT” a la PC mientras el joystick está hacia la derecha.
VII. Enviar las 2 direcciones correctas a PC al desplazar el joystick en diagonal.
VIII. Enviar la dirección del joystick en cada lectura, aunque no haya cambiado.
IX. Detectar todos los pulsadores presionados en un único barrido.
X. Producir acción asociada en la PC al presionar 1 vez un botón del stick.
XI. Producir acciones combinadas en PC al presionar varios botones del stick.
XII. No duplicar una acción asociada a un botón mientras esté presionado.
XIII. Considerar una solución al efecto de rebote de los pulsadores.
XIV. Latencia óptima entre el pulsado de un botón y la recepción en PC.

3.3 – No funcionales

I. Utilización de 1x placa EDU-CIAA-NXP (controlador NXP LPC 4337).
II. Compatibilidad con el sistema operativo Windows, como mínimo.
III. Continuación del funcionamiento del sistema ante un error eventual.
IV. Ubicación del joystick a la izquierda y matriz de pulsadores a la derecha.
V. Finalización del proyecto en tiempo y forma para su muestra en diciembre 2022.

3.4 – Hardware a utilizar

I. Joystick modelo KY-023 (XY Axis).
II. 6x pulsadores de 30 mm normalmente abiertos.
III. Display LCD de 16x2.
IV. LED RGB de 5 mm, de 4 pines.

4 – Diseño del hardware

En base al diagrama en bloques de la figura 2.1, en este apartado se explican los componentes que conforman cada uno y la interconexión entre los mismos.

4.1 – Bloque Joystick

Está compuesto por el sensor KY-023 y el conversor analógico-digital que se dispone en la EDU-CIAA. El sensor está compuesto internamente por 2 potenciómetros de 10 kΩ, utilizados para conocer la posición del joystick en cada eje, y un pulsador SW en el centro [7]. El rango de temperatura permitido para su correcto funcionamiento es de 0°C a 70°C. En la siguiente figura se muestran los terminales que posee:

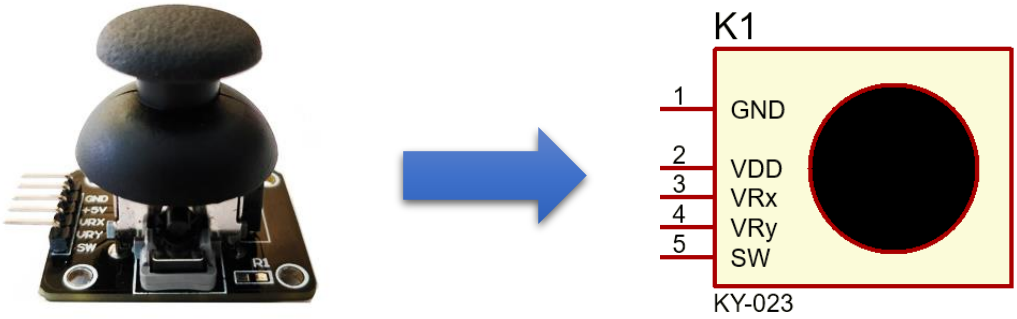


Fig. 4.1. Modelado externo del componente KY-023.

Los terminales GND y VDD corresponden a tierra y alimentación. El rango de alimentación que soporta es de 3.3V a 5V, en este proyecto se utilizará 3.3V.

Los terminales VRx y VRy son salidas analógicas, específicamente de los potenciómetros internos de cada eje, tomando valores desde 0V hasta la tensión de alimentación (en adelante, VDD). En estado de reposo, la tensión para ambos ejes es VDD / 2, mientras que los valores mínimos y máximos se obtienen al desplazar el joystick hacia los extremos. La corriente resulta: $I_{KY023} = 2 * \frac{V_{DD}}{R} = 2 * \frac{3.3V}{10\ k\Omega} = 0.66\ mA$

VRx y VRy estarán conectados a CH3 y CH2, entradas analógicas del ADC. Éste mismo, de 10 bits, devolverá valores entre 0 y 1023 para cada eje. Por otra parte, el terminal SW es una salida digital, conectada al pulsador ubicado en el centro.

4.2 – Bloque Pulsadores

Está compuesto por 6 pulsadores genéricos de diámetro 30mm. Se decidió que cada pulsador estará conectado a un pin de propósito general de la EDU-CIAA, en lugar de utilizar una distribución de “teclado matricial” de 2 filas y 3 columnas (5 pines).

Luego, cada pulsador, en el otro extremo, estará conectado a tierra, como se observa en la hoja n°03 del circuito esquemático (ver anexo 9.1).

4.3 – Bloque Información

El estado general del sistema se informa a través de un display LCD 16x2, modelo LM016L (3 mA máximo), y un LED RGB 5mm de 4 pines tipo ánodo común.

El display consta de los terminales listados a continuación [8]:

- VSS, VDD y VEE: tierra, alimentación +5V y regulación de contraste.
- RS: entrada digital para indicar instrucción (nivel bajo) o dato (nivel alto).
- R/W: entrada digital para indicar lectura (nivel alto) o escritura (nivel bajo).
- E: entrada digital de la señal “Enable”, activada por flanco de bajada.
- D0-D7: líneas digitales de datos. En la configuración para su manejo con 4 líneas de datos y 2 de control, únicamente se utilizan los terminales D4 a D7.

En este proyecto, se trabajará con el LCD únicamente para mostrar mensajes al usuario, por lo que el terminal R/W está conectado a tierra, y se utiliza la configuración de 4 líneas de datos para disminuir la cantidad de conexiones con la placa EDU-CIAA.

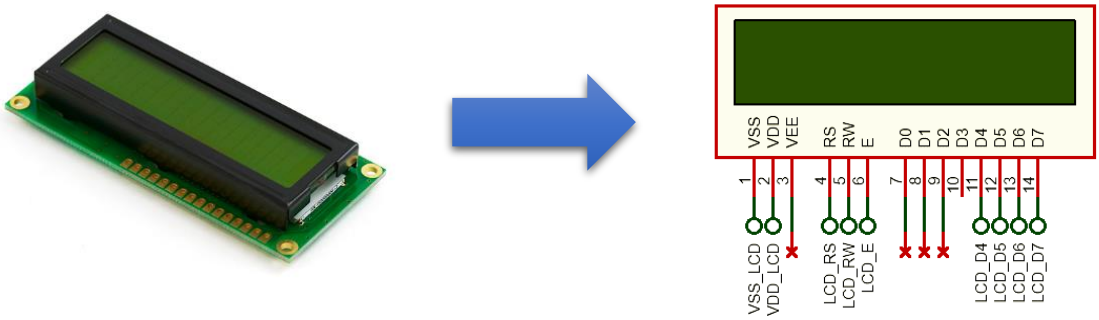


Fig. 4.2. Modelado externo del componente LCD 16x2.

Por otra parte, el LED RGB dispone de un terminal para encender cada color primario que lo compone de manera independiente, y un terminal de ánodo común. Se decidió usar 5V como tensión de alimentación, con resistencias intermedias de 220 Ω en todos los casos para limitar la corriente a un valor cercano a 10 mA en cada diodo.

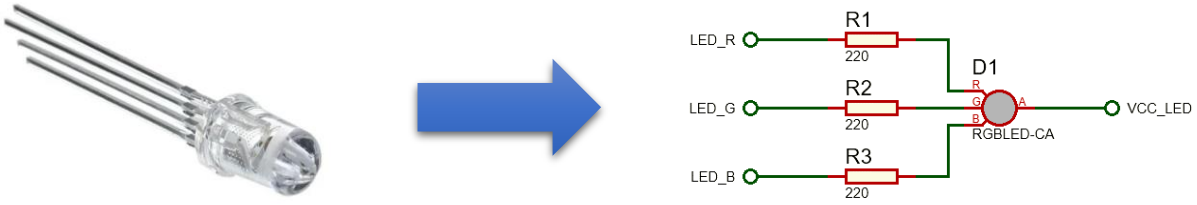


Fig. 4.3. Modelado interno del componente LED RGB.

Se recalcularon las intensidades máximas de corriente para cada diodo, considerando que la tensión V_{OL} de un pin es cero, menor al máximo (0.4V) [9].

$$I_{LED} = \frac{V_{CC} - V_{LED} - V_{OL}}{R} = \begin{cases} I_{rojo} = \dfrac{5V - 2.0V}{220\ \Omega} = 13.63\ mA \\ I_{verde} = \dfrac{5V - 2.81V}{220\ \Omega} = 9.95\ mA \\ I_{azul} = \dfrac{5V - 2.87V}{220\ \Omega} = 9.68\ mA \end{cases}$$

4.4 – Bloque USB

Para la comunicación con la PC, en esta primera versión del proyecto se utiliza exclusivamente USB-OTG, el cual sirve, a su vez, de alimentación a la EDU-CIAA.

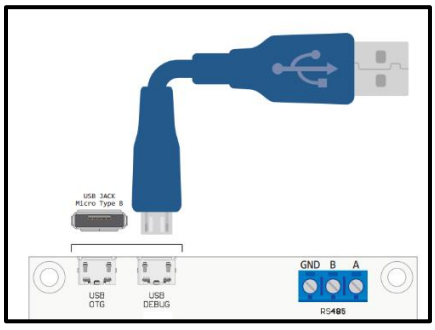


Fig. 4.4. Fichas USB disponibles para su uso en EDU-CIAA.

5 – Diseño de firmware, simulación y depuración

El firmware del Arcade Stick está basado en la biblioteca sAPI, que proporciona una capa de abstracción respecto del hardware, como se observa a continuación:

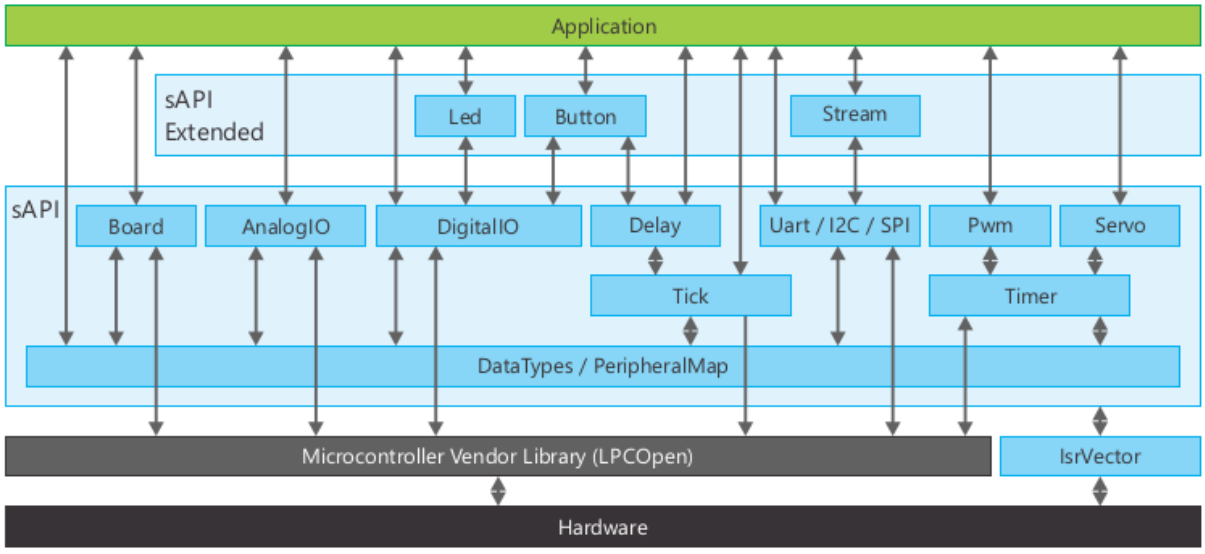


Fig. 5.1. Capas del firmware proporcionado por sAPI.

Para cada periférico descrito en el apartado 4, existe un módulo controlador:

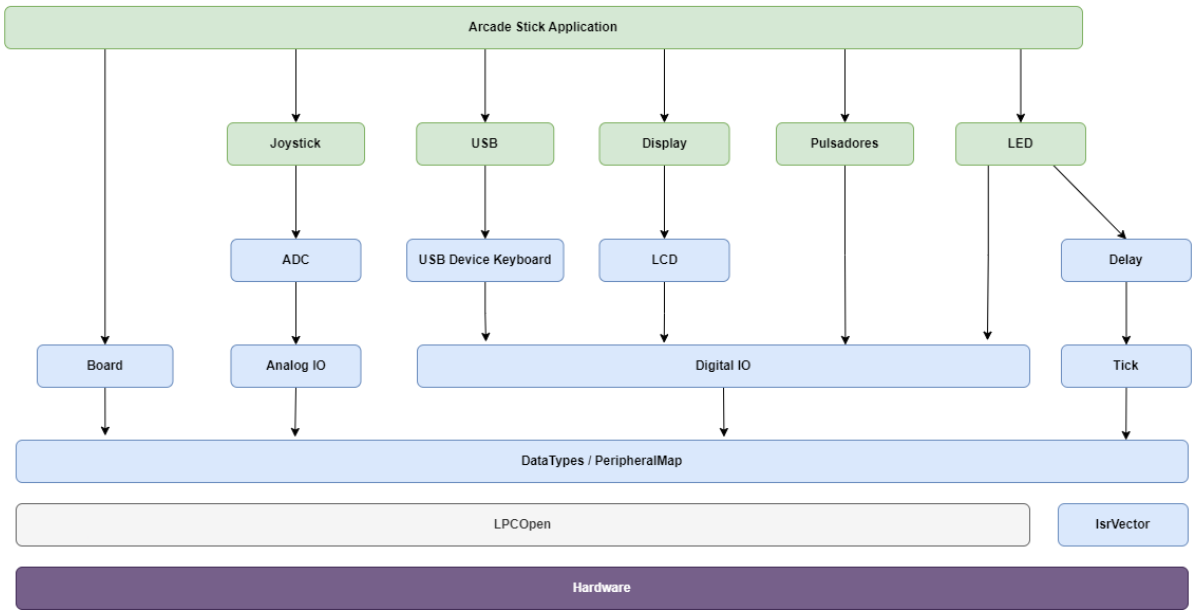


Fig. 5.2. Capas del firmware de Arcade Stick (en verde) basado en sAPI (en azul).

Se planteó una arquitectura RTOS, que consiste en la planificación de tareas y asignación de prioridades, mediante la biblioteca *FreeRTOS*. En este caso, se prioriza la lectura y envío de los controles respecto a la información de usuario.

El período principal es 20 ms, dado el requerimiento de una frecuencia de lectura de controles de 50 Hz mínimo. A continuación, se detalla cada módulo.

5.1 – Comunicación USB

En este primer avance, se logró la comunicación con la PC a través de una biblioteca sAPI de periféricos, precisamente *"usbd_keyboard.h"*, basada a su vez en *"usbd_rom_api.h"*, de LPCOpen. También se utiliza *"sapi_usb_device.h"*, que contiene una función de inicialización general para mouse, teclado y otros dispositivos USB.

De los 2 archivos sAPI indicados, se utilizaron las siguientes funciones:

- usbDeviceInit: establece la configuración inicial del periférico especificado.
- usbDeviceKeyboardCheckKeysCallbackSet: recibe un puntero a la función que indica cuáles teclas deben "presionarse" y enviar a PC según las condiciones. Es necesario entonces que las variables a consultar sean públicas (ejemplo: Flags).
- usbDeviceKeyboardPress: se marca la tecla indicada por parámetro como presionada. Existe un enumerativo con las teclas mapeadas en ASCII.
- usbDeviceKeyboardTasks: si el transmisor se encuentra disponible, invoca a la función de *callback* para enviar las teclas indicadas a la PC. Por tal motivo, dicha función debe invocar *usbDeviceKeyboardPress* para marcar las teclas.

Las teclas enviadas son "W", "A", "S", "D" y "Enter" a partir del estado del joystick, como se explica en la siguiente sección. Se propone investigar acerca de un método alternativo que permita a la PC reconocer al Arcade Stick como un joystick propiamente dicho, en lugar de un teclado, para la próxima etapa de desarrollo; esto permitiría al usuario elegir la traducción de teclas para cada dirección y botón disponible.

5.2 – Lectura del Joystick

Dado a que se dispone de un único conversor analógico-digital en la EDU-CIAA, primero se realiza la lectura del eje X y luego la del eje Y. Mediante la biblioteca sAPI, se utiliza la función *adcInit* para habilitar el ADC. En esta API se lo configura para trabajar con frecuencia de muestreo $f_{ADC} = 200\ kHz$ y resolución $n = 10$ bits.

Se utiliza la función *adcRead*, que habilita el canal indicado, inicia la conversión y se bloquea hasta que finalice para retornar el valor. El tiempo de cada conversión se puede calcular como $n * T_{ADC} = \frac{n}{f_{ADC}} = \frac{10}{200\ kHz} = 50\ \mu s$, resultando $100\ \mu s$ en total.

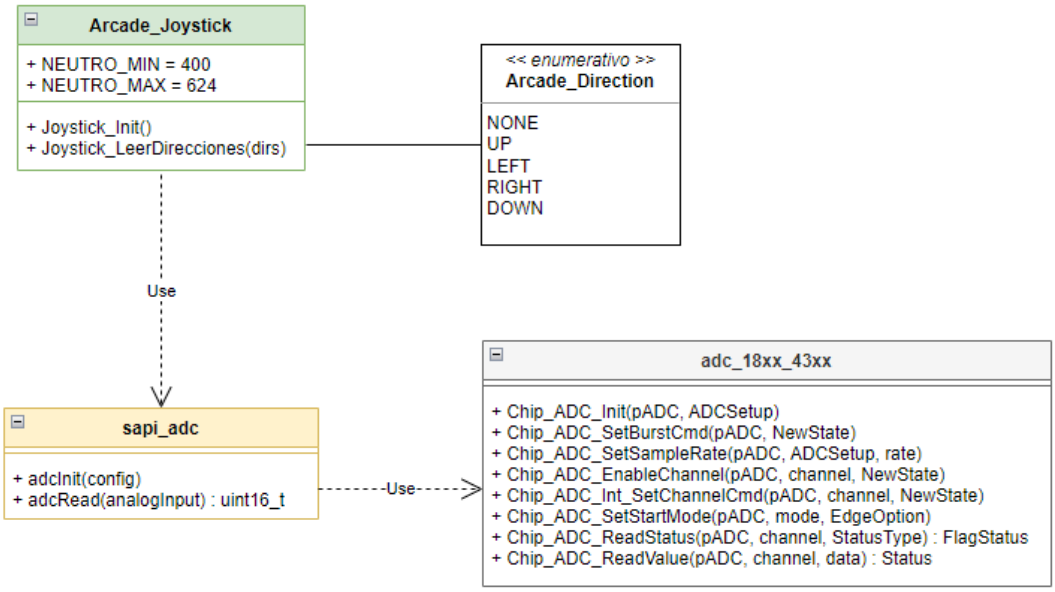


Fig. 5.3. Dependencias sAPI y LPCOpen para módulo Joystick.

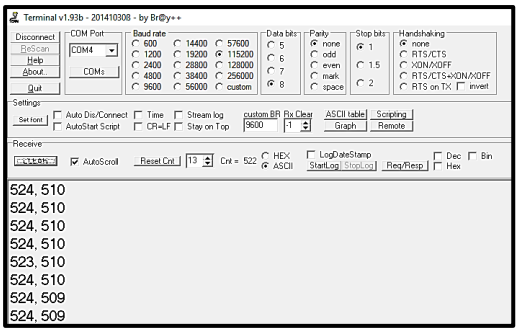
A partir de los valores leídos del conversor (entre 0 y 1023), se asigna por software una dirección principal de cada eje (horizontal y vertical) según cada valor:

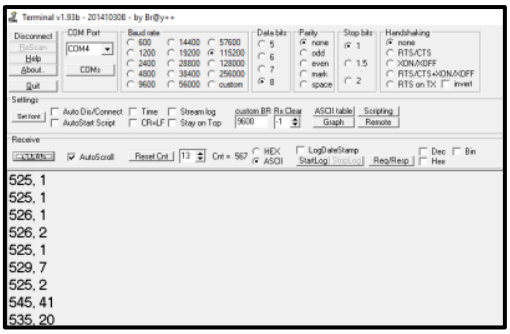
- Para el rango del eje X entre 0 y 400: dirección LEFT, mapeado por tecla “A”.
- Para el rango del eje X entre 624 y 1023: dirección RIGHT, mapeado por “D”.
- Para el rango del eje Y entre 0 y 400: dirección UP, mapeado por tecla “W”.
- Para el rango del eje Y entre 624 y 1023: dirección DOWN, mapeado por “S”.
- Para el rango de un eje entre 400 y 624: dirección NONE, sin mapear.

Como el procesamiento de cada eje es independiente, luego se envían las teclas correspondientes a cada dirección, pudiendo ser hasta 2 en el mismo envío. Las combinaciones posibles dan lugar a 8 direcciones (incluyendo las diagonales).

5.3 – Depuración del Joystick

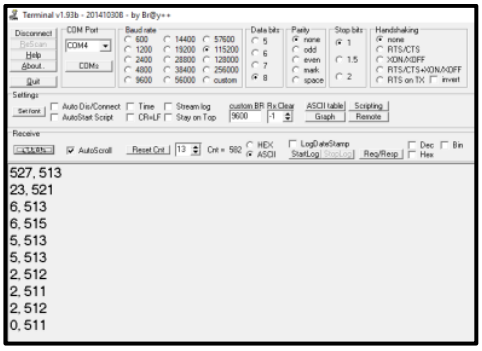
Para verificar que el funcionamiento del sistema de coordenadas mencionado, se utilizó la UART de la EDU-CIAA para enviar el par de valores de los ejes. Los resultados para cada dirección testeada se muestran a continuación. El error no supera el 10%.





Figs. 5.4 y 5.5. Valores (x,y) para el joystick en reposo o *deadzone* // hacia arriba.

Figs. 5.6 y 5.7. Valores (x,y) para el joystick desplazado hacia abajo // hacia la derecha.



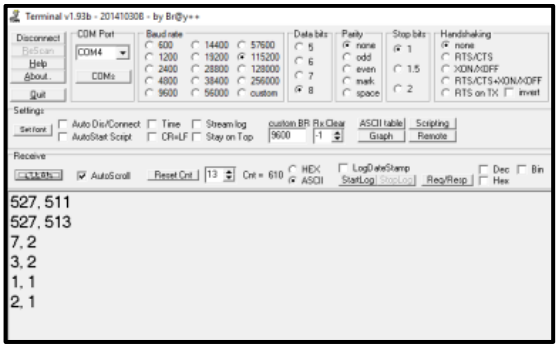


Fig. 5.8 y 5.9. Valores (x,y) para el joystick desplazado hacia la izquierda // hacia arriba e izquierda.

5.4 – Simulación de Joystick

En un primer programa, se accionan LEDs que, según su color, indican en qué posición está siendo accionado el joystick. Cabe destacar que se utilizan los LEDs que ya proporciona la placa EDU-CIAA. En estado de reposo, todos los LEDs se encuentran apagados, como se puede observar en la imagen a continuación:

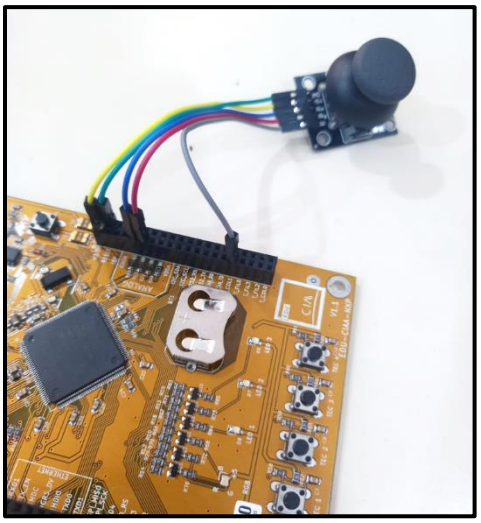
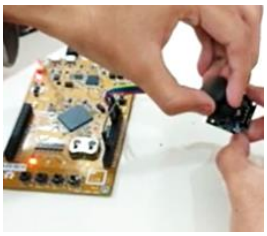


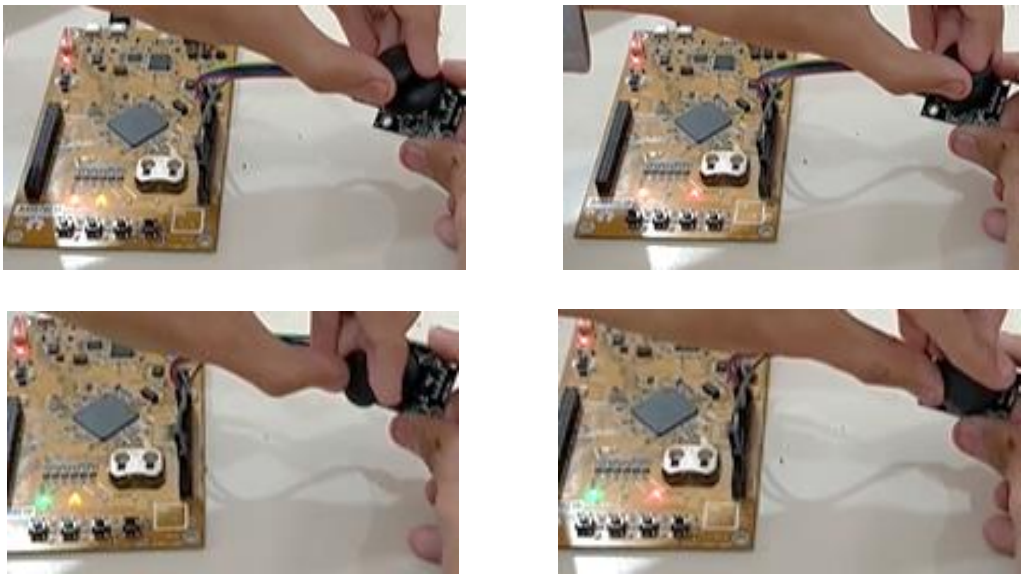
Fig. 5.10. Estado de LEDs mientras el joystick está en su *deadzone*.

Luego, se desplazó el joystick hacia cada una de las direcciones principales, como se puede observar en el siguiente video: https://drive.google.com/file/d/1VUfrMsMoO9ti-2vdj6m1foljctk7Kjif/view?usp=sharing




Figs. 5.11 a 5.14. Estado de LEDs para dirección UP, DOWN, LEFT y RIGHT.

Se prosiguió con el desplazamiento del joystick en direcciones diagonales. Link: https://drive.google.com/file/d/1b2GRXS9GWm1Hsp58RO01RIIvj6qIyMJW/view?usp=sharing



Figs. 5.15 a 5.18. Estado de LEDs para UP-LEFT, UP-RIGHT, DOWN-LEFT y DOWN-RIGHT.

Por último, se probó el funcionamiento en conjunto con la comunicación USB, enviando las direcciones indicadas en la sección 5.2 en cada ciclo de lectura. También se asignó que al accionar el pulsador SW del joystick se envíe la tecla “Enter”. Se adjunta enlace al video: https://drive.google.com/file/d/1Cax3STiZzMDgGUvNg0gO2CHOMenl-58S/view?usp=sharing

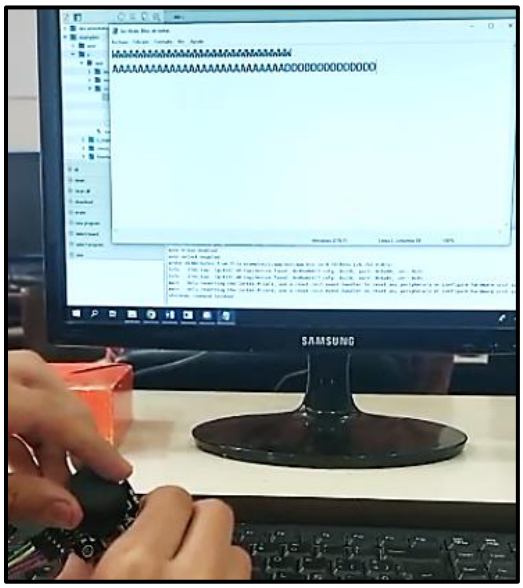


Fig. 5.19. Prueba del envío de teclas “WASD” en el bloc de notas, vía USB OTG.

5.5 – Estados del sistema

El sistema Arcade Stick posee 3 estados que siguen un orden secuencial, como se observa en el siguiente diagrama. La salida “x” representa al color encendido del LED y “z” al estado del display LCD (acción sobre el mismo o un mensaje particular).

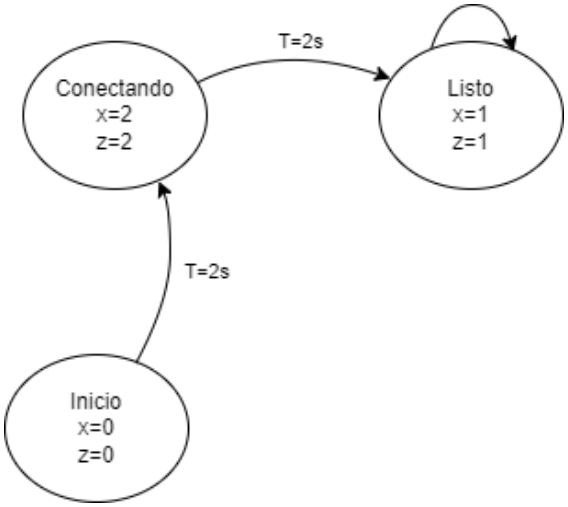


Fig. 5.20. Diagrama de estados del sistema principal Arcade Stick.

- Inicio: es el estado por defecto al alimentar el sistema. Se realiza una inicialización de todos los módulos, pines de entrada/salida y periféricos. Durante este lapso el LED rojo se encuentra encendido (x = 0) y el display LCD es limpiado (z = 0).
- Conectando: una vez realizada la inicialización, se procede a evaluar el estado de la comunicación con la PC. En esta primera versión, al tratarse del mismo medio que la alimentación, se garantiza que la conexión existe. Este estado está reservado para una posterior implementación con Bluetooth. Durante este lapso se parpadea el LED azul (x = 2) y el display muestra “Conectando…” (z = 2).
- Listo: se realiza la lectura de los controles y el envío de las teclas indicadas. Como el sistema solo puede funcionar mientras está alimentado por USB, la única transición posible es hacia el mismo estado, al tratarse de una tarea periódica. En este estado, permanece encendido el LED verde (x = 1) y el display muestra el mensaje “Listo para jugar” (z = 1).

5.6 – Manejo de LCD

La inicialización del display y el envío de caracteres al mismo se realiza por medio del módulo *"display_lcd_hd44780_gpios.h"*, de sAPI. Su utilización permite tener una abstracción respecto a la escritura directa sobre los pines. Se agregó además un módulo propio llamado "LCD", que internamente almacena los mensajes de cada estado y el "avance" de su escritura en el display. Se llama a las siguientes funciones de sAPI:

- lcdInit: recibe el ancho de línea (caracteres por línea, en este caso 16), la cantidad de líneas (en este caso 2), el ancho de carácter y la altura de carácter.
- lcdCursorSet: establece un modo definido del cursor del LCD (ON, OFF, BLINK).
- lcdGoToXY: mueve el cursor a la celda (x,y) indicada por parámetros.
- lcdSendChar: escribe el carácter en línea de datos y activa señal Enable.
- lcdClear: limpia la pantalla del LCD.

También existe una función *lcdSendStringRaw*, para escribir Strings por pantalla de manera bloqueante. Debido a esto último, se decidió enviar cada carácter de un mensaje por separado mediante *lcdSendChar*, evitando el bloqueo del programa, ya que además esta tarea tiene una prioridad menor.

5.7 – Uso de GPIO

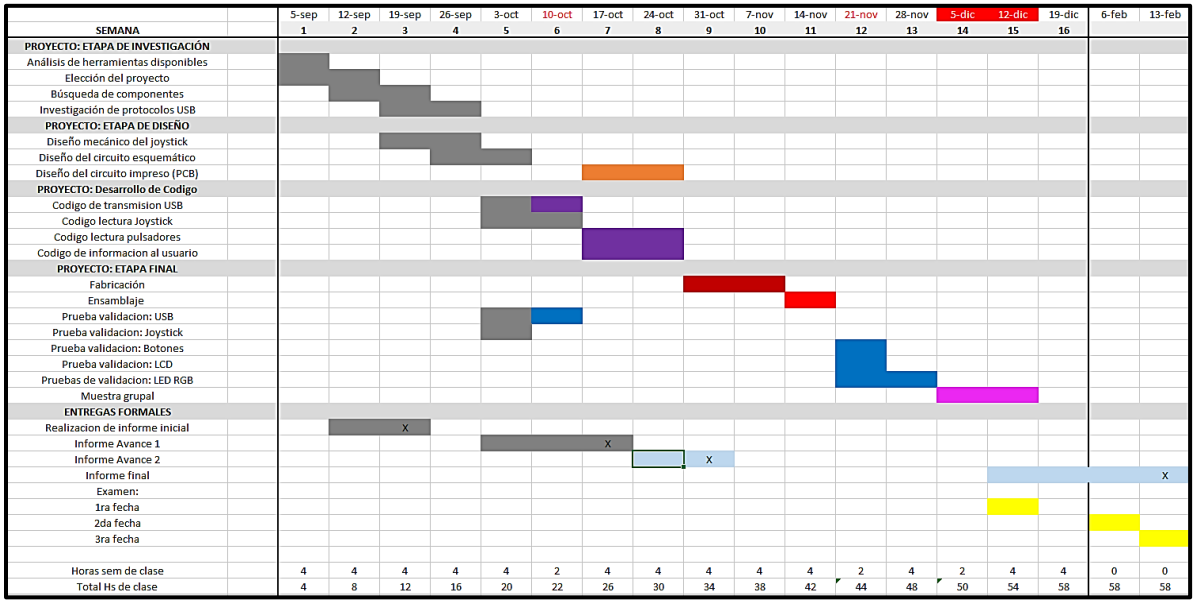
La lectura y escritura directa de los pines I/O de la placa EDU-CIAA se realiza para el barrido de estados de los pulsadores (S1-S6) y el manejo del LED RGB. Sin embargo, para ambos componentes se especificó un módulo que se encarga de estas tareas, de modo de abstraer al programa principal respecto a la distribución elegida de los pines.

La biblioteca sAPI utilizada para este propósito es *"sapi_gpio.h"*.

En el caso del módulo LED, se utiliza además la biblioteca *"sapi_delay.h"* para manejar el parpadeo de un LED de manera no bloqueante, mediante la creación de una variable tipo *delay_t*, y el llamado de las funciones *delayInit* y *delayRead*.

6 – Cronograma

A continuación, se muestra el cronograma estimado del desarrollo del proyecto.



	5-sep	12-sep	19-sep	26-sep	3-oct	10-oct	17-oct	24-oct	31-oct	7-nov	14-nov	21-nov	28-nov	5-dic	12-dic	19-dic	6-feb	13-feb
SEMANA	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
PROYECTO: ETAPA DE INVESTIGACIÓN																		
Análisis de herramientas disponibles																		
Elección del proyecto																		
Búsqueda de componentes																		
Investigación de protocolos USB																		
PROYECTO: ETAPA DE DISEÑO																		
Diseño mecánico del joystick																		
Diseño del circuito esquemático																		
Diseño del circuito impreso (PCB)																		
PROYECTO: Desarrollo de Codigo																		
Codigo de transmision USB																		
Codigo lectura Joystick																		
Codigo lectura pulsadores																		
Codigo de informacion al usuario																		
PROYECTO: ETAPA FINAL																		
Fabricación																		
Ensamblaje																		
Prueba validacion: USB																		
Prueba validacion: Joystick																		
Prueba validacion: Botones																		
Prueba validacion: LCD																		
Pruebas de validacion: LED RGB																		
Muestra grupal																		
ENTREGAS FORMALES																		
Realizacion de informe inicial			X															
Informe Avance 1							X											
Informe Avance 2									X									
Informe final																		X
Examen:																		
1ra fecha																		
2da fecha																		
3ra fecha																		
Horas sem de clase	4	4	4	4	4	2	4	4	4	4	4	2	4	2	4	4	0	0
Total Hs de clase	4	8	12	16	20	22	26	30	34	38	42	44	48	50	54	58	58	58

Fig. 6.1. Cronograma de realización del proyecto.

7 - División de tareas

Se especifica la delegación de las actividades del proyecto a continuación:

Tabla 7.1. División de tareas acordado por el grupo.

Tarea	Responsable	Colaborador
Especificación de requisitos	Calderón, Sergio	Blanco, Valentín
Adquisición de componentes	Blanco, Valentín	-
Diseño mecánico del stick	Calderón, Sergio	Bonifacio, Lucas
Algoritmos de lectura de controles	Bonifacio, Lucas	Blanco, Valentín
Desarrollo del código de comunicación con PC	Blanco, Valentín	Calderón, Sergio Bonifacio, Lucas
Diseño del esquemático	Bonifacio, Lucas	Blanco, Valentín
Diseño del PCB	Blanco, Valentín	Calderón, Sergio
Confección del poncho / Ensamble	Calderón, Sergio	Blanco, Valentín Bonifacio, Lucas
Testeo de controles	Bonifacio, Lucas	Blanco, Valentín
Testeo de comunicación con PC	Calderón, Sergio	Blanco, Valentín
Armado de informes de avance	Blanco, Valentín Bonifacio, Lucas	Calderón, Sergio
Armado de informe final	Blanco, Valentín Calderón, Sergio Bonifacio, Lucas	-

8 - Bibliografía

[1] Pernia, Eric (2019). *Especificaciones EDU-CIAA-NXP v1.1*. Disponible en: https://github.com/epernia/firmware_v3/blob/master/documentation/CIAA_Boards/NXP_LPC4337/EDU-CIAA-NXP/EDU-CIAA-NXP%20v1.1%20Board%20-%202019-01-03%20v5r0.pdf

[2] CIAA (2015). *Dimensiones de la placa EDU-CIAA-NXP.* Disponible en: https://github.com/ciaa/Hardware/blob/master/PCB/EDU-NXP/Doc/PCB.pdf

[3] Play Asia (2020). *Astro City Mini Arcade Stick*. Disponible en: https://www.play-asia.com/astro-city-mini-arcade-stick/13/70dptj

[4] Last Minute Engineers (s.f). *In Depth: How 2-Axis Joystick Works & Interface with Arduino + Processing.* Consultado el 14 de septiembre de 2022. Disponible en: https://lastminuteengineers.com/joystick-interfacing-arduino-processing/

[5] Joy-IT (2017). *KY-023 Joystick Module (X-Y Axis) Datasheet*. Disponible en: https://datasheetspdf.com/pdf-file/1402034/Joy-IT/KY-023/1

[6] Arduino Education (2021). *DIY Game Controllers.* Disponible en: https://www.arduino.cc/education/diy-game-controllers

[7] Component101 (2018). *Joystick Module: Pinout, Features, Arduino Circuit & Datasheet.* Disponible en: https://components101.com/modules/joystick-module

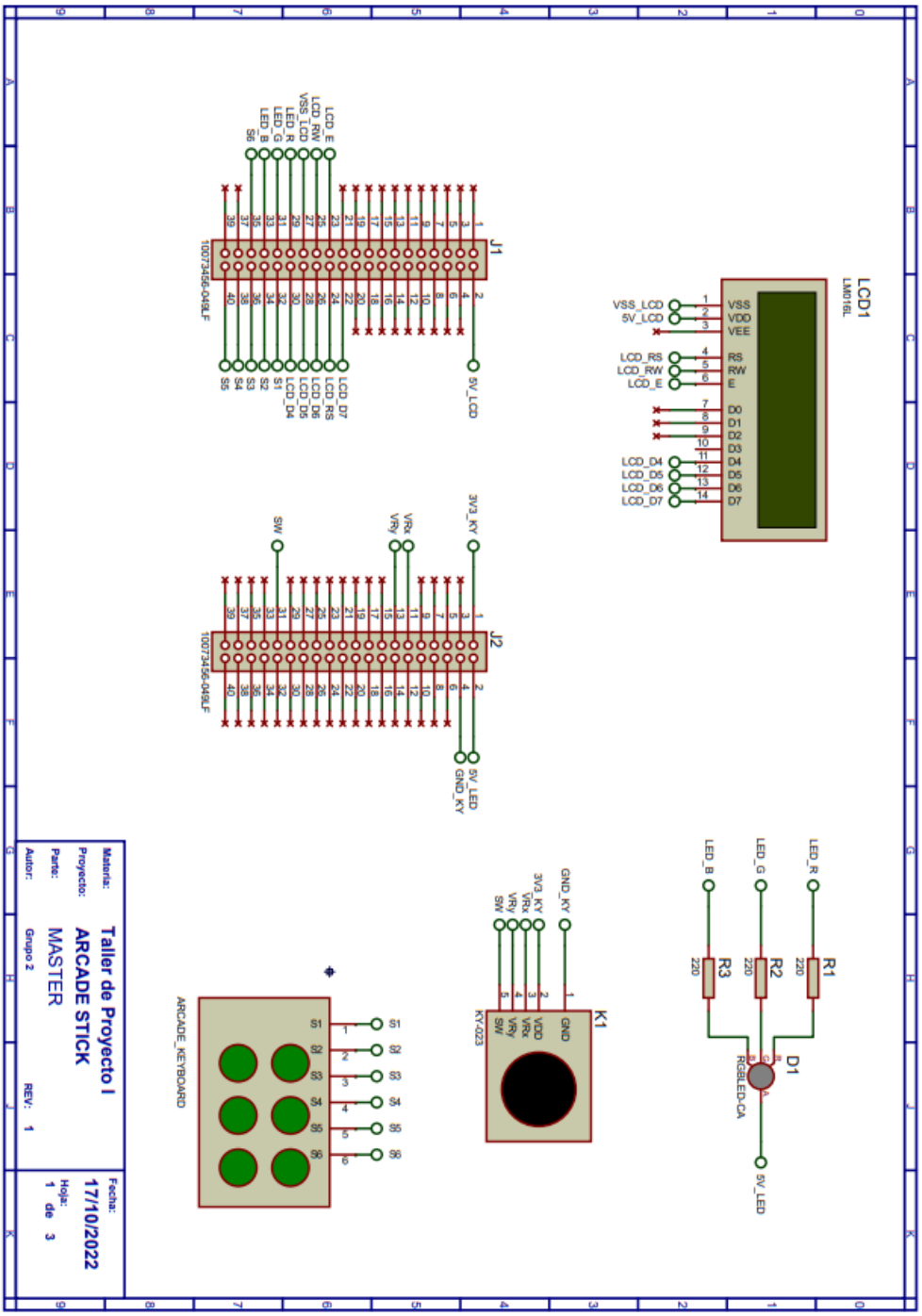
[8] Hitachi (s.f.). *LM016L Datasheet.* Disponible en: https://datasheetspdf.com/pdf-file/1462370/Hitachi/LM016L/1

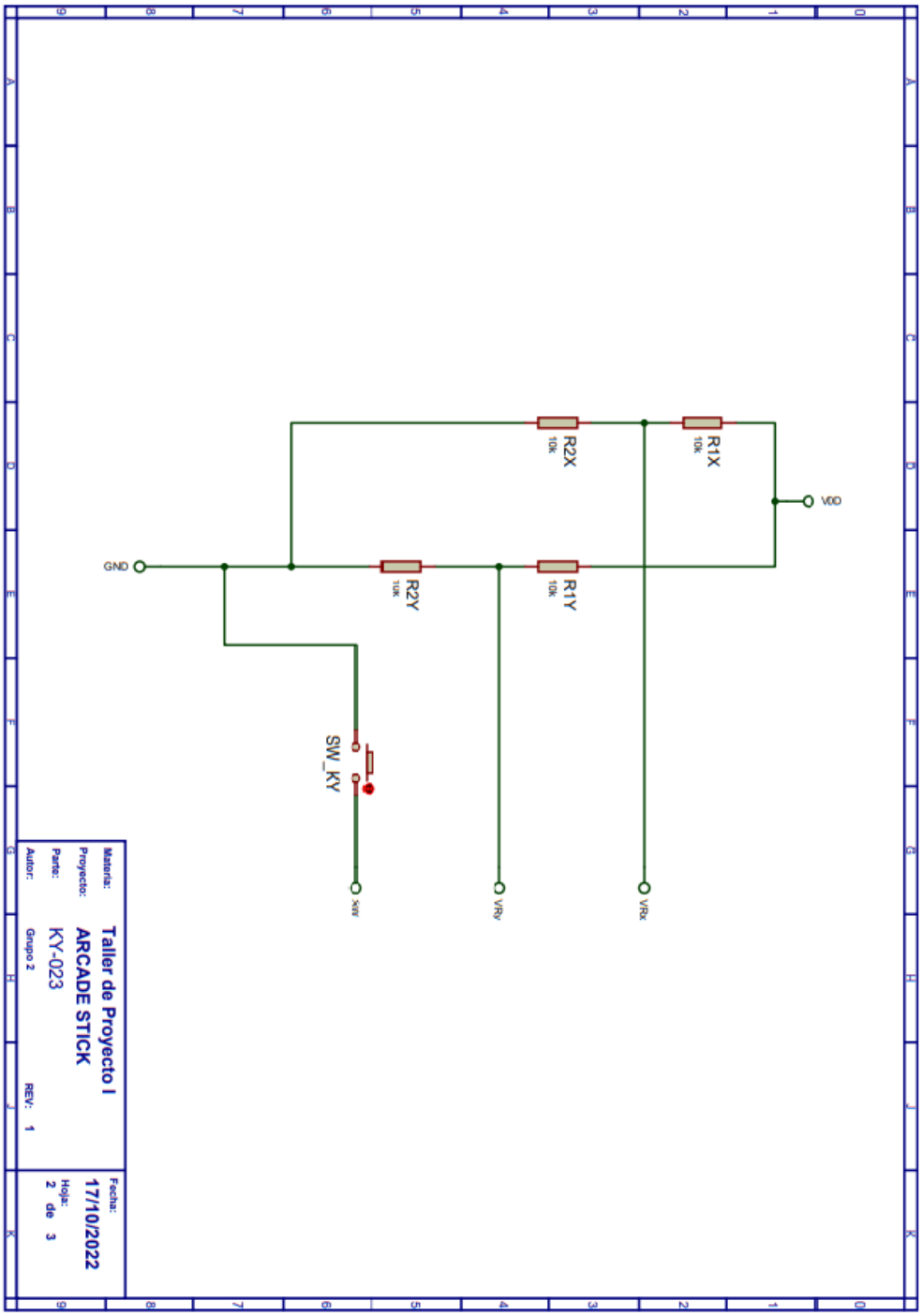
[9] NXP (2020). *LPC435x/3x/2x/1x Datasheet.* Capítulo 10, p. 91.

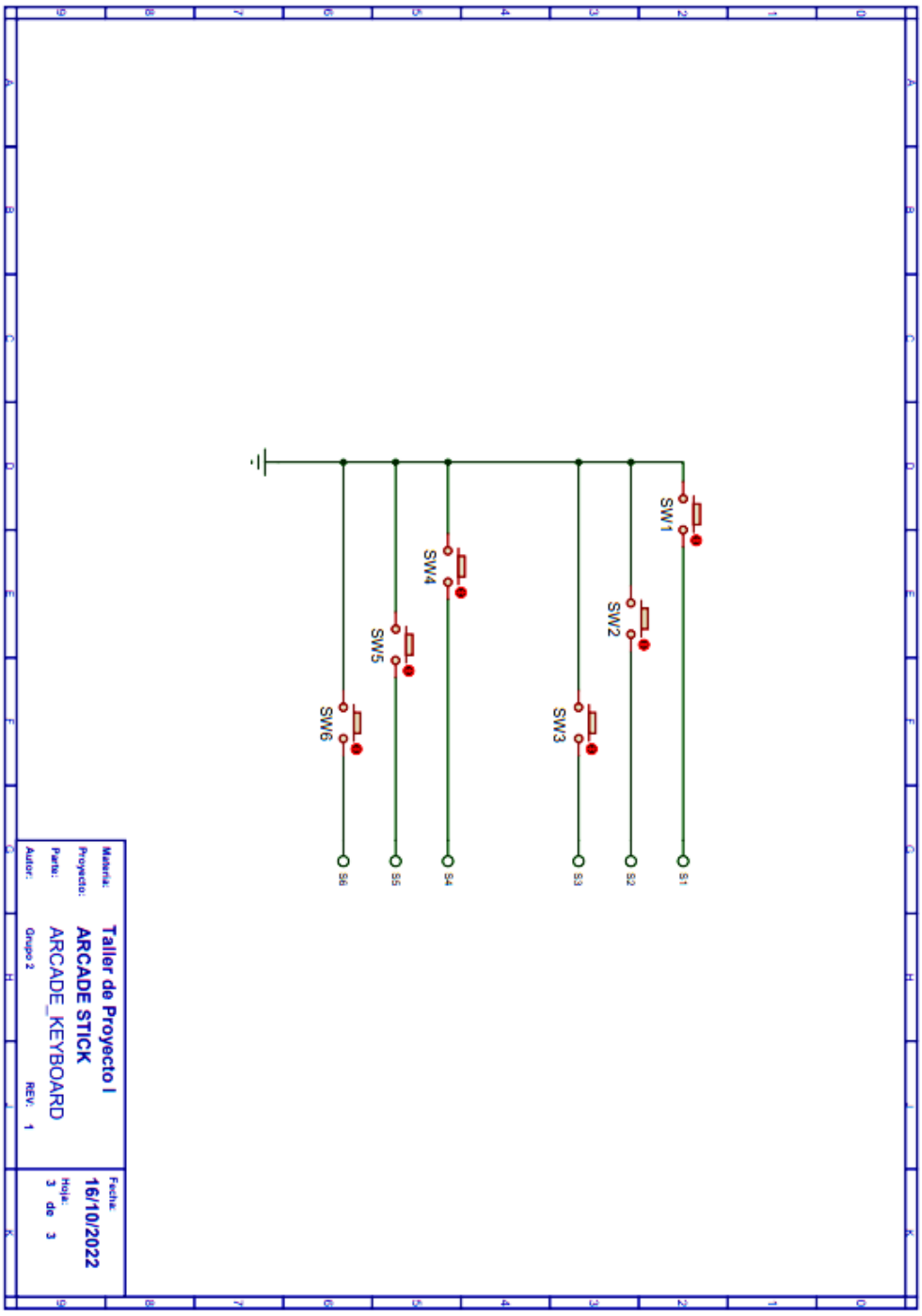
[10] Semana (2013). *Los videojuegos estimulan la creatividad en adolescentes*. Disponible en: https://www.semana.com/vida-moderna/articulo/los-videojuegos-estimulan-la-creatividad-en-adolescentes/363868-3/

9 – Anexos

9.1 – Circuito Esquemático




R2X
10k
R1X
10k
VDD
GND
R2Y
10k
R1Y
10k
SW_KY
SW
VRy
VRx
Materia:
Taller de Proyecto I
Proyecto:
ARCADE STICK
Parte:
KY-023
Autor:
Grupo 2
REV: 1
Fecha:
17/10/2022
Hoja:
2 de 3


SW1
SW2
SW3
SW4
SW5
SW6
S1
S2
S3
S4
S5
S6
Material:
Taller de Proyecto I
Proyecto:
ARCADE STICK
Parte:
ARCADE_KEYBOARD
Autor:
Grupo 2
REV: 1
Fecha:
16/10/2022
Hoja:
3 de 3

9.2 – Lista de materiales

A continuación, se presenta la lista de resistencias, diodos, sensores y otros componentes a adquirir para la realización de este proyecto. Las resistencias de 10kΩ y el pulsador SW_KY vienen incluidos en el sensor KY-023, por lo que su costo es cero.

Taller de Proyecto I

Bill Of Materials for ARCADE STICK

Design Title	ARCADE STICK
Author	Grupo 2
Document Number	1
Revision	1
Design Created	jueves, 29 de septiembre de 2022
Design Last Modified	domingo, 16 de octubre de 2022
Total Parts In Design	19

7 Resistores

Quantity	References	Value	Stock Code	Unit Cost
3	R1-R3	220		$28,34
4	R1X,R1Y,R2X,R2Y	10k		$0,00
Sub-totals:				$85,02

6 Pulsadores

Quantity	References	Value	Stock Code	Unit Cost
6	SW1-SW6	30 mm		$489,00
Sub-totals:				$2.934,00

1 Sensores

Quantity	References	Value	Stock Code	Unit Cost
1	K1	KY-023		$549,00
Sub-totals:				$549,00

1 Diodos

Quantity	References	Value	Stock Code	Unit Cost
1	D1	RGBLED-CA		$50,00
Sub-totals:				$50,00

4 Misceláneos

Quantity	References	Value	Stock Code	Unit Cost
2	J1-J2	10073456-049LF	FCI 10073456-049LF	
1	LCD1	LM016L		$980,00
1	SW_KY			$0,00
Sub-totals:				$980,00

Totals: $4.598,02

domingo, 16 de octubre de 2022 19:55:12