

## **“Poncho educativo para la EDU-CIAA desarrollado por la UNSJ”**

**Autor:** Sarmiento, Leandro Gustavo

**Catedra:** Microcontroladores

**Carrera:** Ingeniería Electrónica

**Profesores:** Dr. Ing. Eugenio Orosco, Ing. Carlos Gil, Ing. Gustavo Ensinck.

**Laboratorio de Electrónica Digital**

**Dpto. de Electrónica y Automática**

**Facultad de Ingeniería – Universidad Nacional de San Juan (FI-UNSJ)**

Resumen: En el marco de un proyecto de cátedra, se realizaron el diseño y la construcción de una placa tipo "poncho" complementando la reciente adquisición de EDU-CIAA para la enseñanza en el ámbito universitario. El poncho consta de elementos digitales y analógicos tales como: potenciómetro, buzzer, UART/RS-232, LCD 16x2, teclado matricial 4x3, y pinout para comunicaciones varias y pines de propósito general. Además, se desarrolló software mediante ejemplo para su uso y enseñanza en las cátedras relacionadas a los sistemas embebidos, en las carreras Ing. Electrónica y Bioingeniería. El diseño del hardware parte de la premisa de maximizar compatibilidad con los formatos pre-establecidos por el proyecto CIAA. El formato elegido es el poncho grande, el cual es de hardware libre como también lo es el presente proyecto. Se cumplió el objetivo principal planteado de aportar al proyecto EDU-CIAA funcionalidad adicional en el ámbito universitario.

### **Introducción**

En este proyecto estudiantil se realizó un Poncho educativo para la EDU-CIAA desarrollado como un proyecto de cátedra. El concepto de poncho se refiere a módulos electrónicos montados sobre el módulo procesador principal, en este caso la EDU-CIAA.

La amplitud de campos de aplicación de la EDU-CIAA es muy extensa, sin embargo, ésta plataforma comprende lo que puede entenderse como un “módulo procesador” y no puede utilizarse por sí sola debido a que carece de interfaces. Es por eso que necesitamos otros módulos electrónicos que permitan ser la interfaz entre la EDU-CIAA y el mundo real. Estas interfaces pueden ser, de usuario, de comunicación, adquisición de datos a través de sensores, etc.

En base a la experiencia en otras plataformas (Arduino, Raspberry Pi, etc.), encontramos que utilizar estas interfaces es la forma en que se obtienen mejores y más rápidos resultados a la hora de desarrollar una aplicación nueva. De esta forma no se pierde tiempo y esfuerzo en desarrollar módulos de hardware ad-hoc y se implementan desarrollos modulares mediante placas montadas sobre una placa base.

Interfaces prediseñadas, probadas y con cierto background de desarrollo (ejemplos, drivers, manuales, etc.) permiten disminuir la barrera de ingreso a la utilización de la EDU-CIAA / CIAA y concentrarse en el diseño de software. De esta manera facilitan la enseñanza de programación y acerca la aplicación de la electrónica a gente de otras ramas.

El diseño de los esquemáticos y del pcb se realizó con el software libre KiCad el cual también es utilizado en el proyecto EDU-CIAA. Al no tener experiencia en este software, se aprendió todo lo relacionado con este desde lo más básico a lo más avanzado para cumplimentar todas las tareas propuestas. El desarrollo de los drivers se hizo en lenguaje C con la ayuda de LPCOpen, esta es una extensa colección de bibliotecas de software libre y programas de ejemplo que permiten a los desarrolladores crear productos multifuncionales basados en microcontroladores LPC con editando de los códigos con el IDE LPCXpresso.

### Especificaciones de diseño planteadas.

La idea de anexarle un poncho a la EDU-CIAA es brindarle más funcionalidad a la placa base para poder implementar diversos programas en el ámbito de la enseñanza a alumnos universitarios. En la Figura 1 se muestra el desarrollo final y las especificaciones de diseño propuestas son las siguientes:

- Potenciómetro
- Buzzer
- Display LCD 16x2.
- Teclado matricial 4 x 3.
- UART – RS232
- Salidas de propósitos generales.
- I2C
- SPI
- CAN

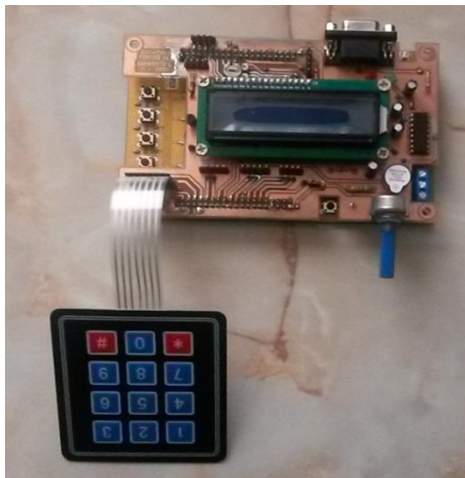


Figura 1. Vista general de placa base EDU-CIAA y poncho con teclado y otros elementos.

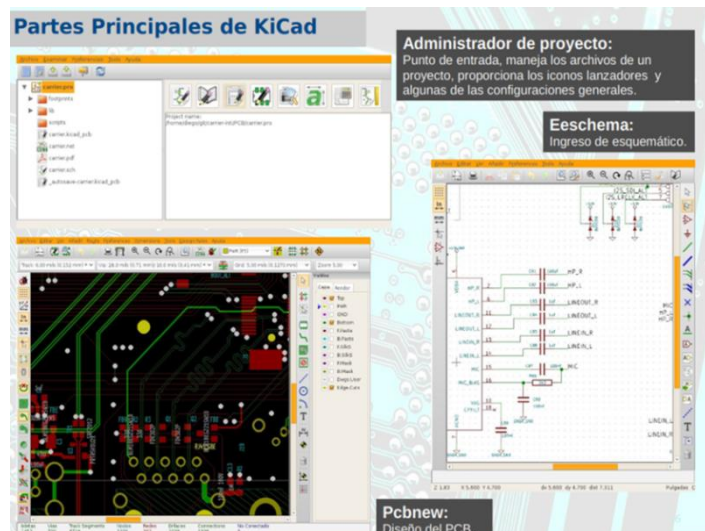
### Desarrollo del hardware.

En el presente desarrollo se utilizó el modelo de Poncho Grande el cual se encuentra en el repositorio de ponchos del Proyecto CIAA cuyo autor es el Ing. Diego Brengi. Una vez que se obtuvo el modelo se empezó a editar el esquemático y posteriormente el pcb. Con la premisa que el proyecto de la CIAA se diseñó mediante el software KiCad, por lo que el punto de inicio en este proyecto fue aprender el uso de esta aplicación de diseño.

El programa KiCad tiene licencia GPL y sus principales ventajas son: disponibilidad de código fuente; gratuidad; formatos abiertos y documentables; formato en modo texto (ASCII); multiplataforma; utilizado en la mayoría de los proyectos actuales libre fuente actuales. Y desventajas como: documentación cambiante; cambios constantes en formatos (mejoras); diferencias entre distintas versiones; necesita seguir mejorando su interfaz; necesita incorporar funcionalidades para optimizar tiempos y para circuitos más complejos.

En virtud que el proyecto de cátedra incluye el objetivo de aprendizaje y difusión de las herramientas relacionadas al proyecto CIAA, tal como el desarrollo del hardware basado en KiCad, se presenta a continuación a modo informativo las partes principales de dicho software.

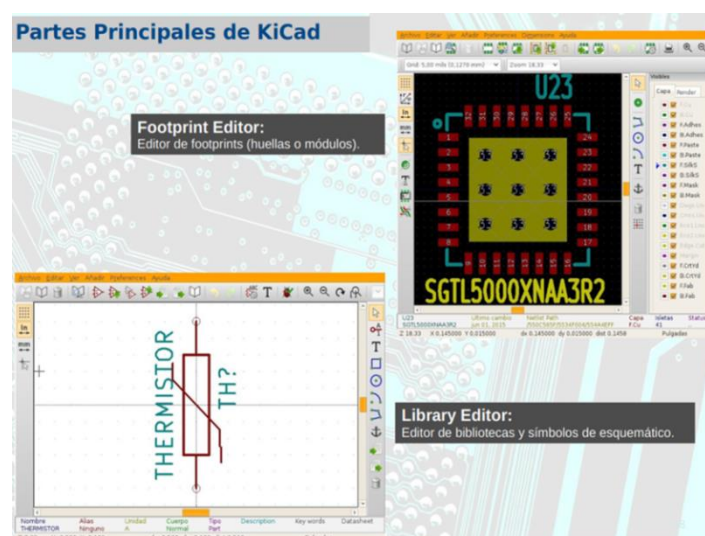
En la Figura 2(a) se muestran el administrado de proyecto, Eschema y Pcbnew, siendo el punto de entrada y manejos de archivos relacionados al proyecto, el diseño de los esquemáticos y del diseño de la placa de circuito impreso (PCB), respectivamente. Luego, el visor 3D es para observar el aspecto físico y el GerbView muestra el diseño en formato estándar Gerber (Figura 2(b)). Finalmente, en la Figura 2(c) se detallan el editor de Footprint y el de librería, para generar componentes nuevos o editar existentes. Toda la documentación oficial del programa se encuentra en <http://kicad-pcb.org/help/documentation/>.



(a)



(b)



(c)

Figura 2. Vistas de partes principales del software KiCad. Las imágenes fueron sacadas del siguiente curso dictado por el INTI.  
<https://github.com/INTI-CMNB/Practicas-Curso-Kicad/blob/master/Materiales/Kicad-Intro-D1.pdf>

## Esquemáticos de poncho educativo.

**ADC/DAC:** Los tres ADC y el DAC se conectan a un conector de cuatro terminales (Figura 3). El ADC3 tiene la particularidad que a través de una Jumper se conecta al punto medio de un potenciómetro circular lineal. Esto se diseñó así para que el alumno pueda interactuar con dicho sensor para simular por ejemplo rotaciones angulares.

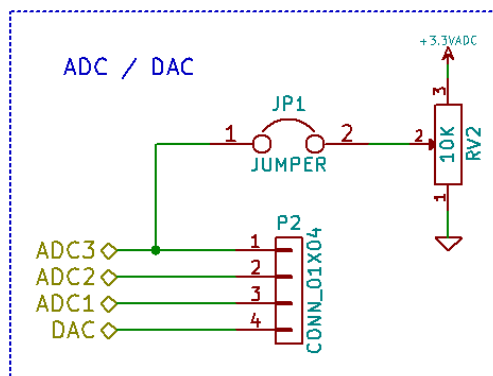


Figura 3. Esquemático ADC/DAC.

**LCD:** Para la interacción visual con la EDU-CIAA se diseñó el circuito típico para manejar un LCD en la configuración de 4 bits. Además, se conectó un preset para poder ajustar el contraste del display (Figura 4).

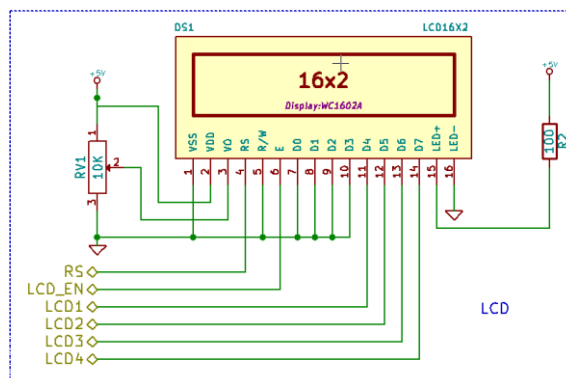


Figura 4. Esquemático para LCD.

**Teclado:** El teclado matricial de 4x3 se diseñó de tal manera que las líneas tanto de las columnas como las filas están conectadas a un conector de siete terminales en el cual se conecta un teclado de membrana quedando externo al poncho (Figura 5).

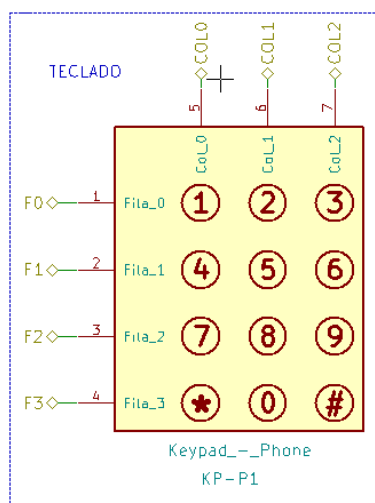


Figura 5. Esquemático de conexión a teclado matricial 4x3.

**GPIO:** Las Entrada/Salida de Propósito General son pines genéricos, cuyo comportamiento (incluyendo si es un pin de entrada o salida) se puede programar por el usuario en tiempo de ejecución (Figura 6). Se determinó poner dos conectores de cuatro terminales cada uno y el GPIO8 se usó para la configuración en la base de un transistor el cual enciende o apaga un Buzzer.

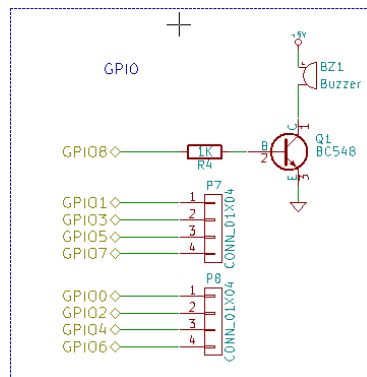


Figura 6. Esquemático de pines GPIO y Buzzer.

**RS232:** Para la comunicación serie se decidió incorporar el circuito típico para la transmisión con la ayuda de un CI MAX232 añadiendo dos Leds para indicar lumínicamente cuando transmite y cuando recibe datos. El conector DB9 también se ubica en la placa el cual es hembra (Figura 7).

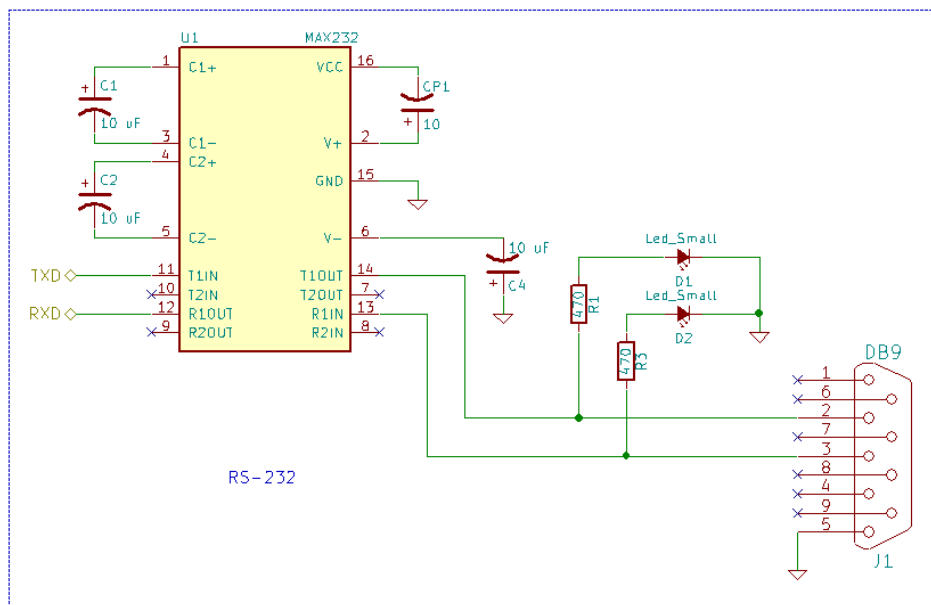


Figura 7. Esquemático para conexiones de RS-232.

**CAN Bus:** El bus CAN es un protocolo de comunicaciones basado en una topología bus para la transmisión de mensajes en entornos distribuidos las señales además de las tensiones y de una señal de masa se conectan a un conector de cinco terminales (Figura 8(a)).

**SPI:** El Bus SPI es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. El bus de interfaz de periféricos serie o bus SPI es un estándar para controlar casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj (comunicación sincrónica) por ende estas líneas se conectan junto a señales de tensión y masa a un conector de cinco terminales (Figura 8(b)).

**I<sup>2</sup>C:** Este Bus de datos serial se utiliza principalmente internamente para la comunicación entre diferentes partes de un circuito. El I<sup>2</sup>C está diseñado como un bus maestro-esclavo. La transferencia de datos es siempre inicializada por un maestro; el esclavo reacciona. Por

ende estas líneas se conectan junto a señales de tensión y masa a un conector de cinco terminales (Figura 8(c)).

**Ethernet:** Estos terminales no se usaron porque no aspiramos a tener comunicación Ethernet pero en un futuro tenemos ganas de diseñar un poncho para poder utilizar este tipo de comunicación ya que se puede usar tranquilamente para diseñar un sistema embebido IOT (Internet of Things) (Figura 8(d)).

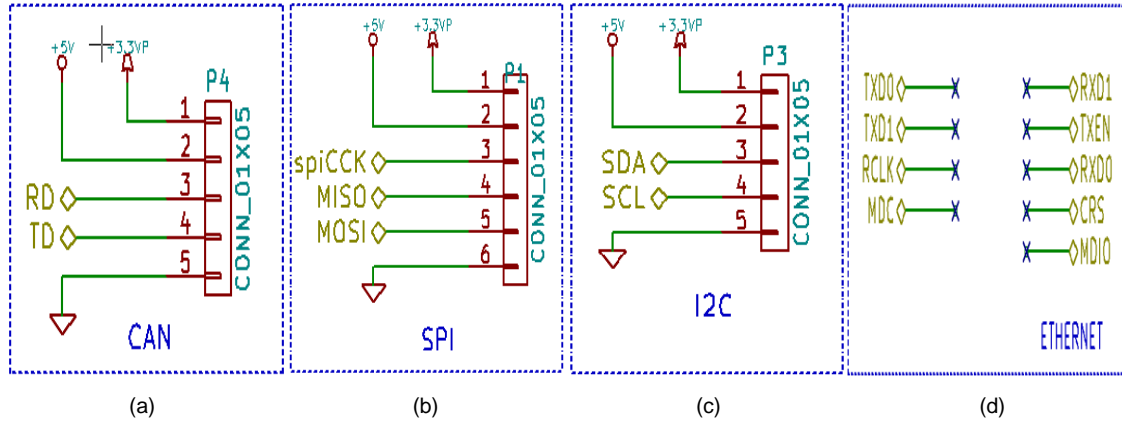


Figura 8. Esquemáticos de comunicaciones digitales.

### Placa de circuito impreso de poncho educativo (PCB).

Para llevar acabo el diseño del PCB se uso la plantilla de “poncho grande” proporcionada por el proyecto en la pagina web en la seccion ponchos. La misma se diseñó de dos capas y la distribución de los componentes se estudió las recomendaciones de diseño del tutorial en <http://alternatezone.com/electronics/files/PCBDesignTutorialRevA.pdf>. En las imágenes de la Figura 9 se ven la parte de inferior (a) y la parte superior (b) del diseño final del poncho educativo.

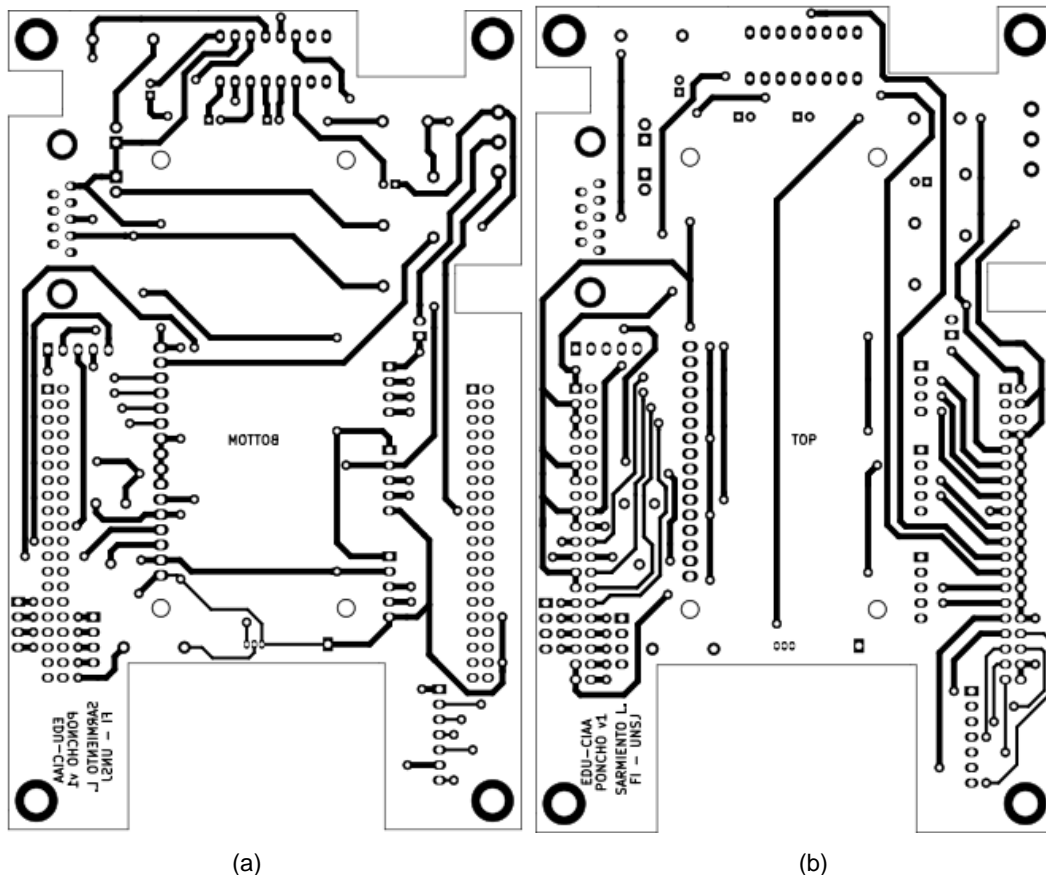


Figura 9, Diseño final de placa circuito impreso parte de inferior (a) y la parte superior (b)

### **Desarrollo de Drivers.**

El desarrollo de los drivers se hizo en lenguaje C con la ayuda de LPCOpen, esta es una extensa colección de bibliotecas de software libre y programas de ejemplo que permiten a los desarrolladores crear productos multifuncionales basados n microcontroladores LPC con la edición de los códigos con el IDE LPCXpresso. Los drivers de los periféricos se desarrollaron en lenguaje C, de autoría del presente grupo de trabajo (autor y asesores), y se destacan los aportes de las personas de la comunidad de embebidos<sup>32</sup>.

<https://groups.google.com/forum/#!forum/embebidos32>

Los proyectos ejemplos se encuentran en el siguiente repositorio donde se respetan los derechos de autor (GPL) de todos los contribuyentes al proyecto en el desarrollo de los drivers.

<https://github.com/leansarmi/Projects>

### **Conclusion y discusiones.**

Se cumplieron las pautas de diseño y funcionalidad en el desarrollo del hardware y del software. Se espera que se pueda utilizar el poncho en materias de enseñanza de microcontroladores, donde en la presente institución se dictan en carreras de Ingeniería Electrónica y Bioingeniería. También se espera emplearlo con profesionales y estudiantes de otras áreas y de todo el país, ya que el diseño del hardware es libre.

Con la funcionalidad ampliada de la EDU-CIAA se podrá ampliar los alcances de la CIAA, bajar costos de desarrollo y aprendizaje, democratizar el uso de la CIAA en empresas e individuos y reducir tiempo de enseñanza.