

Desarrollo e implementación de control de fuente de corriente por software  
Alumno: Sala De Ferraris, Facundo Ignacio  
Carrera: Ingeniería Electrónica

**Universidad Nacional de General San Martín**  
**Escuela de Ciencia y Tecnología**

*Campus Miguelete - Martín de Irigoyen 3100*  
*San Martín – Pcia. De Buenos Aires - Argentina*  
[www.unsam.edu.ar](http://www.unsam.edu.ar)  
[EscuelaCyT@unsam.edu.ar](mailto:EscuelaCyT@unsam.edu.ar)



**UNSAM**  
UNIVERSIDAD  
NACIONAL DE  
SAN MARTÍN

ESCUELA DE CIENCIA Y TECNOLOGÍA

## **Carrera: Ingeniería Electrónica**

### **Proyecto Final Integrador**

**Alumno: Sala De Ferraris, Facundo Ignacio**

**Legajo: CYT-3257**

**Tema: DESARROLLO E IMPLEMENTACIÓN DE CONTROL DE FUENTE DE CORRIENTE POR SOFTWARE**

**Fecha de presentación:**

---

Sala De Ferraris Facundo Ignacio

---

## Contenido

Introducción.....	4
Requerimientos.....	4
Alcances y Objetivos.....	4
Gestión de Proyecto.....	5
WBS (Work Breakdown Structure).....	5
Diagrama de Gantt.....	6
Evaluación de riesgos.....	9
Presentación de la fuente de corriente.....	11
Componentes del control original.....	13
Relevamiento de funcionamiento de los pines originales.....	14
Modos de control de la fuente y señales de accionamiento.....	16
Comienzo de diseño de funciones.....	17
Funciones a programar para el control.....	17
Presentación de Hardware a utilizar.....	18
Selección de hardware de control.....	18
Asignación de uso de pines de entrada y salida.....	21
Software (entorno de programación para la placa EDU-CIAA).....	24
Lenguaje C.....	24
Testing Software.....	24
Proceso del desarrollo Testing.....	25
Diagrama de flujo de software de control en la EDU-CIAA.....	26
Hardware (Placa interfaz entre EDU-CIAA y fuente).....	29
Detalle del diseño de la placa de control.....	29
Cálculos para las polarizaciones de componentes.....	29
Región de corte y saturación.....	31
Aislación eléctrica.....	32
Diseño de emulador de pulsos de encoder.....	33
Diseño y fabricación.....	33
Criterios de diseño eléctrico y físico.....	33
Software de diseño.....	35



Diseño de Placa (Circuito impreso) .....	35
Placa y modulo de control terminado .....	40
Software de control HMI.....	42
Entorno de desarrollo usado.....	42
Interface de control de fuente .....	42
Pruebas de funcionamiento .....	46
Conclusiones .....	48
Bibliografía .....	49
Referencias externas.....	49

## Introducción

El proyecto consistió en desarrollar e implementar un controlador para dos fuentes de corriente (Microbeams) OM-52e a través de una interfaz gráfica Hombre-Máquina (HMI) en computadora (PC). Las fuentes son utilizadas actualmente por la Comisión Nacional de Energía Atómica (CNEA) para la excitación de distintos elementos, requerida para varios tipos de experimentos. Las mismas se utilizaban manualmente mediante el uso de pulsadores y potenciómetros (ENCODER) que modifican el valor de la corriente entregada. El objetivo del proyecto fue crear una placa interfaz Analógica/Digital que permita la modificación de la corriente entregada mediante el uso de una PC con un software que permite ajustarlo de forma manual o automática dependiendo del elemento a excitar y de las necesidades. Esta interfaz de usuario para PC se desarrolló en Guide, de MATLAB (Entorno (JAVA)). El proyecto fue desarrollado sobre una (EDU-CIAA), la cual contiene un microprocesador ARM Cortex (M4). La conexión de las fuentes hacia la placa se mantuvo de forma original mediante dos conectores Db25 y hacia la PC con una interfaz Universal Serial Bus (USB).

## Requerimientos

Este proyecto por desarrollar deberá contemplar los siguientes requerimientos:

- El control de la fuente de corriente OM-52e de la CNEA.
- Corriente controlable a través de una PC.
- El control de la fuente tiene que mantener los mandos originales, para un manejo intuitivo.
- El protocolo de comunicación estará definido para una posible ampliación del software de control en la PC.

## Alcances y Objetivos

La prestación del proyecto incluye el hardware a utilizar, el diseño del hardware del sistema de control antes mencionado, el software de la placa de control y el software de interfaz HMI.

## Gestión de Proyecto

### WBS (Work Breakdown Structure)

La gestión de proyectos es la disciplina que estudia el planeamiento, la organización, la motivación, y el control de los recursos con el propósito de alcanzar uno o varios objetivos. a producir un único producto, servicio o resultado. A continuación se detalla las etapas del mismo.

1. Hardware
  - 1.1. Investigación sobre el dispositivo
    - 1.1.1. Niveles de tensión
    - 1.1.2. Lógica de funcionamiento
    - 1.1.3. Relevamiento de información de uso
    - 1.1.4. Investigación de la solución
      - 1.1.4.1. Análisis de datos
    - 1.1.5. Investigación de tecnología
      - 1.1.5.1. Aislamiento
      - 1.1.5.2. Comunicación
      - 1.1.5.3. Fuente de alimentación
      - 1.1.5.4. Software CIAA/PC
  - 1.2. Diseño de prototipo
    - 1.2.1. Elección de Hardware
    - 1.2.2. Diseño de esquemático
    - 1.2.3. Diseño de impreso
    - 1.2.4. Fabricación
    - 1.2.5. Compra de componentes
    - 1.2.6. Armado
    - 1.2.7. Test de placa
2. Software
  - 2.1. Programación en CIAA
  - 2.2. Programación del software HMI de PC

## Diagrama de Gantt

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1	Investigación del dispositivo	1 día	jue 26/10/17	jue 26/10/17		
6	Investigación de solución	51 días	vie 27/10/17	vie 05/01/18	1	
7	Análisis de datos relevados	7 días	vie 27/10/17	lun 06/11/17		
8	Investigación de tecnología	44 días	mar 07/11/17	vie 05/01/18		
9	Aislamiento	4 días	mar 07/11/17	vie 10/11/17		
10	Comunicación digital	3 días	lun 13/11/17	mié 15/11/17		
11	Fuente de alimentación	2 días	jue 16/11/17	vie 17/11/17		
12	Software	35 días	lun 20/11/17	vie 05/01/18		
13	Software de PC	20 días	lun 20/11/17	vie 15/12/17		
14	Firmware de CIAA	15 días	lun 18/12/17	vie 05/01/18		
15	Implementación de solución	90 días	jue 01/02/18	mié 06/06/18	6	
16	Elección de hardware a utilizar	1 día	jue 01/02/18	jue 01/02/18		
17	Elección de componentes	1 día	jue 01/02/18	jue 01/02/18		
18	Prototipo de hardware interfaz	22.5 días	vie 02/02/18	mar 06/03/18		
19	Diseño de esquemático	5 días	vie 02/02/18	jue 08/02/18		
20	Diseño de Impreso	4 días	vie 09/02/18	mié 14/02/18		
21	Fabricación de Impreso	3 días	jue 15/02/18	lun 19/02/18		

Página 1

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
22	Compra de componentes	1 día	mar 20/02/18	mar 20/02/18		
23	Armado de placa	3 días	mié 21/02/18	vie 23/02/18		
24	Test de placa	6.5 días	lun 26/02/18	mar 06/03/18		
25	Desarrollo firmware CIAA	39 días	lun 12/03/18	jue 03/05/18		
26	Diagrama de flujo	2 días	lun 12/03/18	mar 13/03/18		
27	Definición de protocolo de comunicación	3 días	mié 14/03/18	vie 16/03/18		
28	Investigación de herramientas de programación	5 días	lun 19/03/18	vie 23/03/18		
29	Software	29 días	lun 26/03/18	jue 03/05/18		
30	Desarrollo software PC	34 días	vie 20/04/18	mié 06/06/18		
31	Diagrama de flujo	2 días	vie 20/04/18	lun 23/04/18		
32	Definición de protocolo de comunicación	1 día	jue 03/05/18	jue 03/05/18		
33	Investigación de herramientas de programación	5 días	vie 04/05/18	jue 10/05/18		
34	Software	20 días	jue 10/05/18	mié 06/06/18		
35	Verificación y validación	16 días	vie 08/06/18	vie 29/06/18	15	
36	Prueba de funcionamiento	10 días	vie 08/06/18	jue 21/06/18		
37	Instalación de prototipo funcional	14 días	mar 12/06/18	vie 29/06/18		
38	Demora personal por temas académicos	122 días	mar 19/06/18	mié 05/12/18		

Página 2

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
39	Cierre de documentación	82.7 días	jue 15/11/18	lun 11/03/19		
40	Manuales de Uso	39 días	lun 03/12/18	jue 24/01/19		
41	Manuales de Service	38.5 días	lun 03/12/18	jue 24/01/19		
42	Informe final	68 días	jue 15/11/18	lun 11/03/19		
43	Presentación	12 días	jue 31/01/19	vie 15/02/19		
44	Vacaciones	10 días	mar 01/01/19	lun 14/01/19		

Página 3

Desarrollo e implementación de control de fuente de corriente por software  
 Alumno: Sala De Ferraris, Facundo Ignacio  
 Carrera: Ingeniería Electrónica

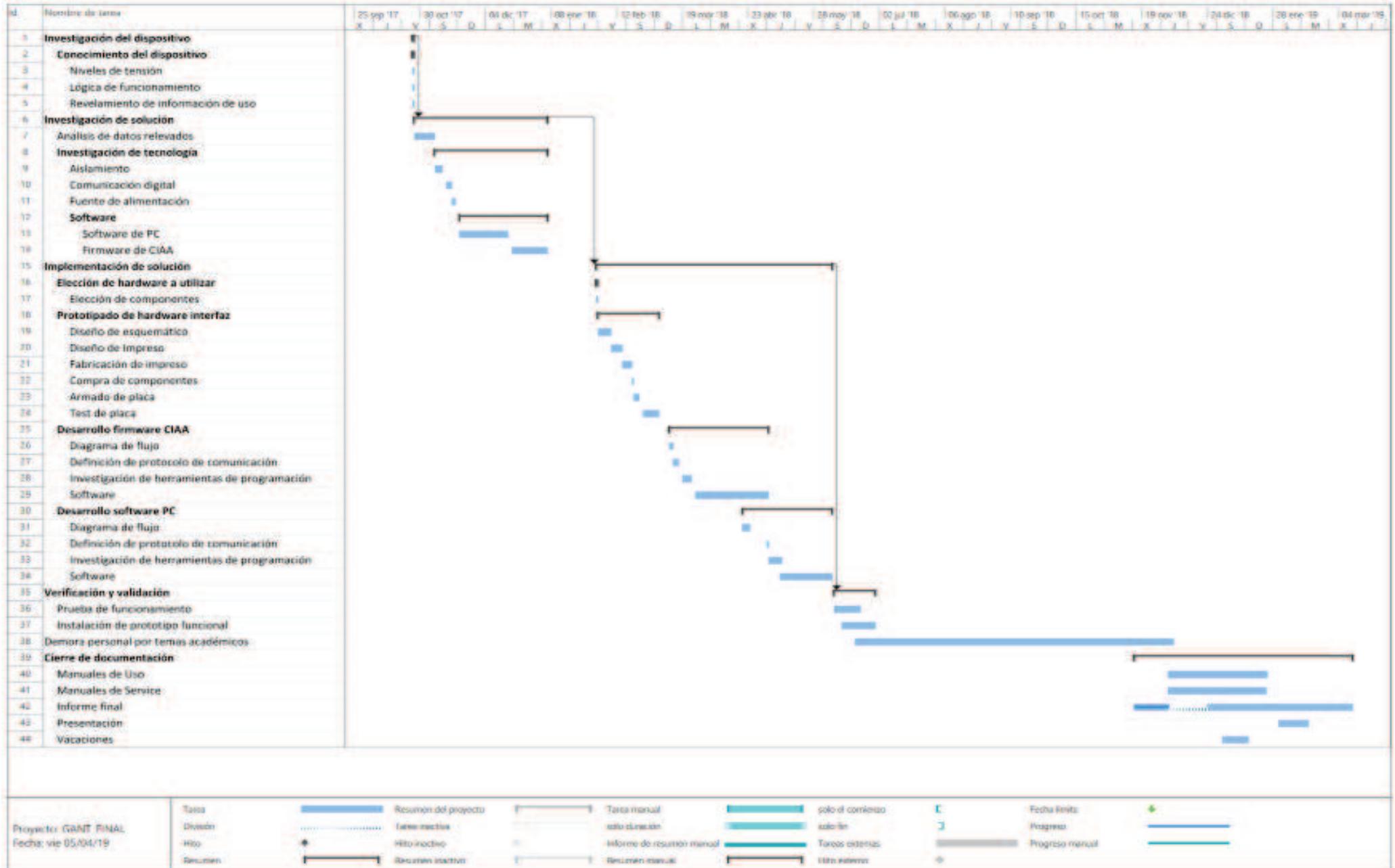


Tabla 1, GANTT

## Evaluación de riesgos

EVALUACIÓN DEL RIESGO		VALORACIÓN DEL RIESGO										MEDIDAS A TOMAR					
n°	Riesgo identificado	Probabilidad			Consecuencias			Estimación del riesgo				Medida de control	Información extra	¿Riesgo controlado?		Acción requerida	
		PB	PM	PA	LD	D	ED	T	TO	M	I			IN	SI		NO
HW1	Perdida del interés en el proyecto	X				X			X				Mantener un régimen de trabajo		X		Darle el tiempo suficiente y continuo
HW2	Cambios en los requisitos del proyecto	X			X			X					Comunicación continua con el Director y/o Codirector del proyecto para solucionar inconvenientes de descoordinación.		X		Información constante del Director del proyecto
HW3	Mala planificación del proyecto		X			X					X		Revisión de los entregables.	Contemplando que baja experiencia del desarrollador del proyecto se implementa una capacitación basada en la revisión de proyectos de índole similar	X		Capacitación basado en proyectos anteriores y con ayuda de gente experimentada en proyectos
HW4	Subestimación de los tiempos necesarios para la finalización del proyecto		X		X				X				Revisión del diagrama de tiempos necesario para la finalización del proyecto.		X		Re planificación y revisión del diagrama de Gantt
HW5	Falta de documentación	X			X				X				Mantener una constante documentación de las tareas a realizar.		X		Mantener al día la documentación
HW6	Falta de equipamiento y/o partes necesarias para el desarrollo del proyecto	X				X			X				Revisión del equipamiento necesario.		X		Control de stock y búsqueda de componentes alternativos
HW7	Fallas del equipamiento anexo (FUENTE)	X				X			X				Se dispone de dos fuentes para minimizar riesgo.		X		Trabajar de forma segura.
HW8	Daño externo	X			X			X					Imposibilidad de controlar el riesgo.	Todo riesgo de probabilidad baja de daño tanto de las personas como del equipamiento.		X	Monitoreo continuo del sistema de control de riesgos

Tabla 2, Evaluación de riesgos del proyecto

		Consecuencias		
		Ligeramente Dañino	Dañino	Extremadamente Dañino
		LD	D	ED
Probabilidad	Baja B	Riesgo trivial T	Riesgo tolerable TO	Riesgo moderado MO
	Media M	Riesgo tolerable TO	Riesgo moderado MO	Riesgo Importante I
	Alta	Riesgo moderado M	Riesgo importante I	Riesgo Intolerable IN

Tabla 3, Tabla de Riesgos, probabilidades y consecuencias

La Tabla 3 nos da las consecuencias en función de la probabilidad o viceversa.

Riesgo	Acción y temporización
Trivial (T)	No se requiere acción específica
Tolerable (TO)	No se necesita mejorar la acción preventiva. Sin embargo se deben considerar soluciones más rentables o mejoras que no supongan una carga económica importante. Se requieren comprobaciones periódicas para asegurar que se mantiene la eficacia de las medidas de control.
Moderado (M)	Se deben hacer esfuerzos para reducir el riesgo, determinando las inversiones precisas. Las medidas para reducir el riesgo deben implantarse en un período determinado. Cuando el riesgo moderado esta asociado con consecuencias extremadamente dañinas, se precisará una acción posterior para establecer, con más precisión, la probabilidad de daño como base para determinar la necesidad de mejora de las medidas de control.
Importante (I)	No debe comenzarse el trabajo hasta que se haya reducido el riesgo. Puede que se precisen recursos considerables para controlar el riesgo. Cuando el riesgo corresponda a un trabajo que se está realizando, debe remediarse el problema en un tiempo inferior al de los riesgos moderados.
Intolerable (IN)	No debe comenzar ni continuar el trabajo hasta que se reduzca el riesgo. Si no es posible reducir el riesgo, incluso con recursos ilimitados, debe prohibirse el trabajo.

Tabla 4, Significado de Riesgos

La Tabla 4 muestra cada riesgo con su significado.

## Presentación de la fuente de corriente

En Ilustración 1 e Ilustración 2 se muestra la fuente de corriente OM-52e donde se ve que cuenta con una pantalla, un pulsador de Reset y una tecla de encendido. Esta fuente es utilizada en CNEA para la excitación de un cañón acelerador de partículas. Dichas partículas son aceleradas y con un colimador de haces se disponen a ser colisionadas contra distintos elementos, según lo requerido por el estudio.

Estas fuentes tienen un rango de operación muy amplio y a la vez muy precisos.

En la fuente se puede ver los siguientes controles:

**Pantalla:** es una display de 4 dígitos de 7 segmentos que muestra el valor de la corriente seleccionada.

**Pulsador Reset:** es presionado para comenzar la configuración de la corriente de la fuente, comenzando en cero amperios (0A).

**Tecla de encendido:** esta tecla enciende y apaga la fuente (ON-OFF)

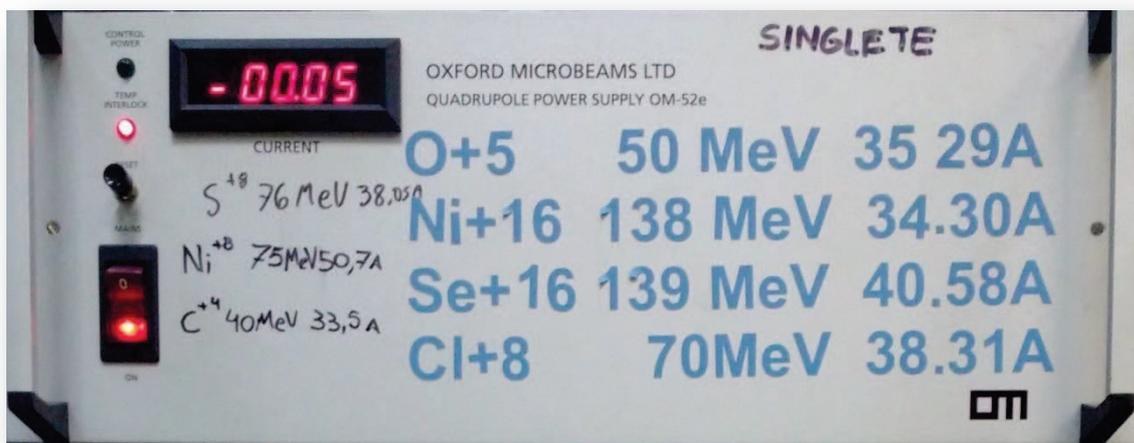


Ilustración 1, Fuente de corriente OM-52e (Vista frontal)

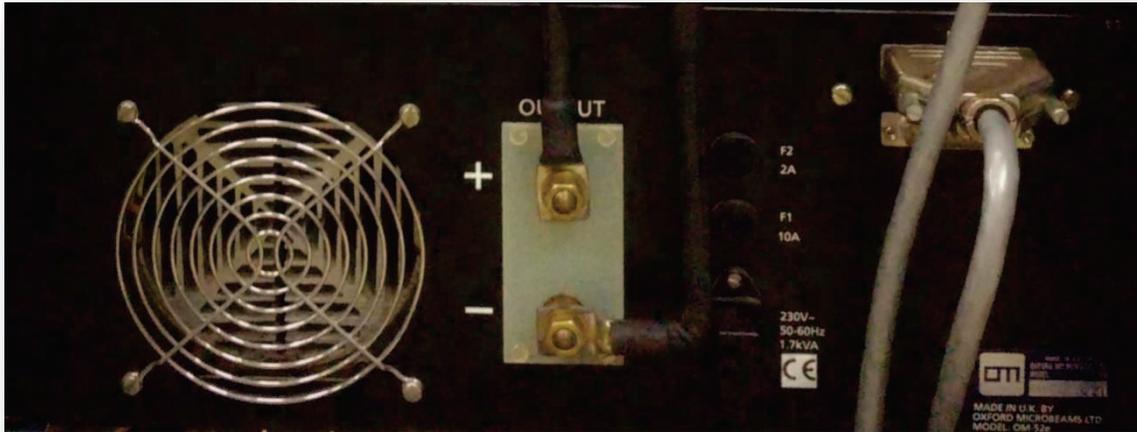


Ilustración 2, Fuente de corriente OM-52e (Vista trasera)

En la Ilustración 3 se muestra el comando original de la fuente. Se ve que el comando para cada fuente es independiente y tiene los mismos controles. Cada una de las salidas tipo conector A (Db25) son también independientes a cada fuente.



Ilustración 3, Control de fuente de corriente OM-52e



Ilustración 4, Control de fuente de corriente OM-52e

En la Ilustración 4 se ven las dos ramas de control de las fuentes de corriente OM-52e.

Para la selección de la escala de cambio de corriente se tienen 3 pulsadores con retención, que pueden ser accionados uno a la vez. Para aumentar y disminuir el valor de corriente se usa un encoder.

1. **Coarse:** Paso grueso que varía de a 5,2506A por vuelta de encoder entera.
2. **Medium:** Paso medio que varía de a 0,58496A por vuelta de encoder entera.
3. **Fine:** Paso fino que hace variar de a 0.0029952A por vuelta de encoder entera.

### Componentes del control original

Acerca del Encoder: Se trata de un encoder rotativo (Rotary encoder, rotary switch...) que es un dispositivo pasivo, similar a un potenciómetro pero de infinitas vueltas, y digital. El utilizado es de 128 pasos por vuelta que arroja pulsos como salida en función del ángulo de giro y de su velocidad. En la Ilustración 5 se muestra la salida de un encoder comercial:

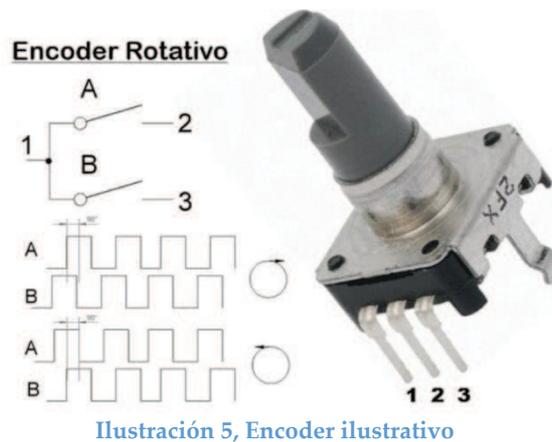


Ilustración 5, Encoder ilustrativo

Observación: El sistema original tiene un encoder a través del cual la fuente 1 y 2 generan los pulsos necesarios para su control. Depende también del sentido de giro que se le de al encoder (Derecha o izquierda). Para emular el sentido de giro se tiene un pin del Conector A en estado alto o bajo, según el caso. Se usa 5V o 0V para esa línea.

La conexión original de la fuente con el control es a través de conectores Db25 como se muestra en la Ilustración 6.



Ilustración 6, Db25 Macho a la izquierda y hembra a la derecha

## Relevamiento de funcionamiento de los pines originales

Los equipos utilizados para el relevamiento de funcionamiento fueron un osciloscopio y un voltímetro digital.

El uso del osciloscopio fue para determinar el comportamiento de cada una de las señales analizando sus períodos de cambios y sus amplitudes. El voltímetro se usó para medir continuidad y tensiones de alimentación.

En la siguiente Tabla 5 se muestran los datos relevados de las mediciones.

<b>Función detectada</b>	<b>Pin del Db25 (Común y Negado)</b>
Sentido	2/ <u>15</u>
Pulso	3/ <u>16</u>
Coarse	4/ <u>17</u>
Medium	5/ <u>18</u>
Fine	6/ <u>19</u>
Nada	7/ <u>20</u>
Reset	8/ <u>21</u>
Temp	9/ <u>22</u>
VDD	1
GND	14

Tabla 5, muestra los pines de ambos controladores

Observación: Todas las señales están duplicadas con su exacto negado de la misma, según su funcionamiento original. También nótese que todo estado posible está descrito dentro del cuadro de posibilidades Tabla 5. Estas son las señales que hay que generar para controlar la fuente.

El procedimiento consistió en medir el estado de todos los pines del Conector A en un determinado momento (sin estar accionado ninguno de los pulsadores y el encoder quieto en una posición conocida).

Después se comenzó a presionar de forma independiente cada uno de los pulsadores. Se tomó nota, entre cada uno de estos pasos realizados, la variación percibida en los 25 pines del Conector A así como también su función en la fuente de corriente. Ver Ilustración 7.

Como segundo paso se tuvo en consideración las variaciones de los encoders (generador de pulsos por paso). Se asociaron los cambios de cada uno de los encoders en los pines del conector A para poder entender el funcionamiento del mismo.

Tomado ya el conociendo del funcionamiento de los pines del conector A, por descarte se encontraron los pulsos de Reset, Temperatura y alimentación del control.

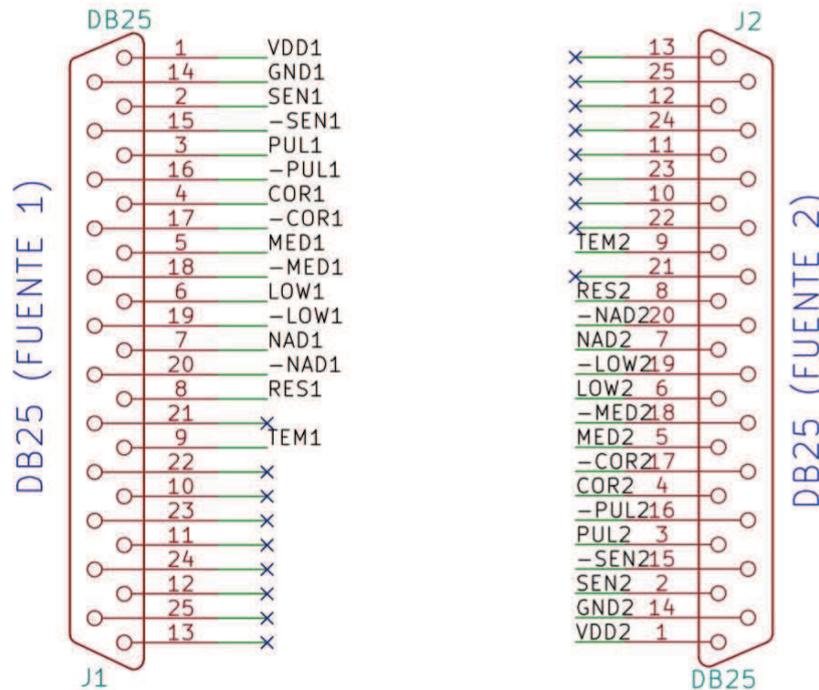


Ilustración 7, Db25 distribución de pines

Como último y muy importante, se mantuvo contacto con el operador de las fuentes para conocer su funcionamiento y todas las condiciones de operación que tienen. Éstas son detalladas en el siguiente resumen de modos que completa la información necesaria para comenzar el trabajo de desarrollo.

### Modos de control de la fuente y señales de accionamiento

- Selección de "Coarse"
- Selección de "Medium"
- Selección de "Fine"
- Quitar selección de todos los comandos
- Identificación de sentido de giro: Incremento/Decremento
- Cantidad de incrementos/decremento (Pulsos de 10  $\mu$ S de ancho)
- Lectura de Reset del sistema (Pulso de 500 mS aproximadamente)
- Lectura de protección de temperatura (sujeto a pruebas de corte por sobre-temperatura, no se garantiza su funcionamiento sin realizar dichas pruebas)
- Los estados solicitados se integran en modo diferencial (dos pines) con un estado Hi de 3,3V y un estado Low de 0,8V. En modo diferencial entre pines se tiene un estado de 2,5V y otro de -2,5V.
- Intervalo de tiempo mínimo entre incrementos/decrementos es de 10 ms.

## Comienzo de diseño de funciones

### Funciones a programar para el control

Para el control se mantuvieron las mismas funciones y sus nombres como original de la fuente de corriente. Estas funciones fueron tomadas y detalladas en la Tabla 6, donde se ve la función a modelar y en que pin se obtiene un "1" lógico. Estos solo muestran las señales normales y no las negadas ya que se entiende que vienen de a pares, la señal más su complemento negado.

Función	Sentido	Sentido	Pulso	Coarce	Médium	Fine	Nada
		2/15	3/16	4/17	5/18	6/19	7/20
Coarse	+	X	P	X			
Coarse	-		P	X			
Médium	+	X	P		X		
Médium	-		P		X		
Fine	-	X	P			X	
Fine	+		P			X	
Nada							X

Tabla 6, secuencia de funciones según pin

La Tabla 6 muestra para cada función el "1" lógico (X = 5V) y la P indica que existe un pulso. En el pin Pulso solo se ve por cada carácter que se manda un pulso. Esto sucede siempre que el carácter esté dentro de la tabla de posibilidades.

Cada fila determina una función y describe en donde tiene que estar los valores en alto y en que líneas del Conector A.

## Presentación de Hardware a utilizar

### Selección de hardware de control

Basándose en las directivas de uso del proyecto original, comenzado con la cátedra de digitales 2, utilice una placa CIAA NXP prestada por el establecimiento y en concordancia con el co-director de proyecto.



Ilustración 8, Versión educativa de la Computadora Industrial (EDU-CIAA)

Esta placa es la EDU-CIAA-NXP, una versión de bajo costo de la CIAA-NXP (Ilustración 8) pensada para la enseñanza universitaria, terciaria y secundaria. Contiene un micro controlador LPC4337 (dual core ARM Cortex-M4F y Cortex-M0).

El Proyecto CIAA nació en el año 2013 como una iniciativa conjunta entre el sector académico y el industrial, representados por la ACSE y CADIEEL, respectivamente.

Los objetivos del Proyecto CIAA son:

1. Impulsar el desarrollo tecnológico nacional.
2. Darle visibilidad positiva a la electrónica Argentina.
3. Generar cambios estructurales en la forma en la que se desarrollan y utilizan los conocimientos.

4. Todo esto en un marco de trabajo libre, colaborativo y articulado entre industria y academia.

En la siguiente figura se observa un diagrama en bloques de la EDU-CIAA basada en

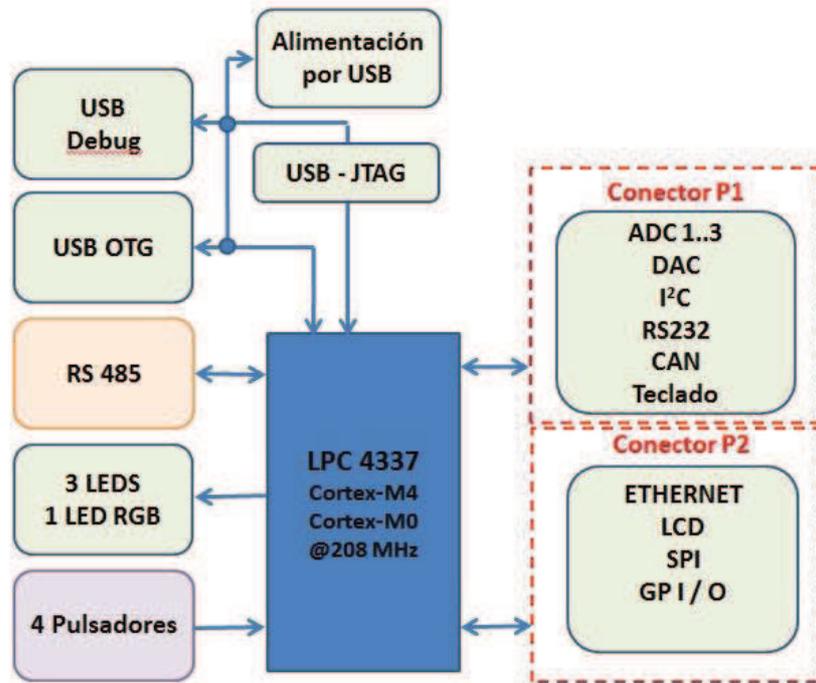


Ilustración 9, diagrama en bloques de EDU-CIAA

LPC4337 (Ilustración 9).

La EDU-CIAA cuenta con los siguientes módulos:

- 2 puertos micro-USB (uno para aplicaciones y debugging, otro para alimentación).
- Salidas digitales implementadas con leds RGB.
- Entradas digitales con pulsadores.
- 1 puerto de comunicaciones RS 485 con bornera.
- 2 conectores de expansión:
  - P1
    - 3 entradas analógicas (ADC<sub>0\_1,2y3</sub>),
    - 1 salida analógica (DAC<sub>0</sub>),
    - 1 puerto I<sup>2</sup>C.
    - 1 puerto asincrónico full dúplex (para RS-232).
    - 1 puerto CAN.

- 1 conexión para un teclado de 3x4.
  - P2
- 1 puerto Ethernet.
- 1 puerto SPI.
- 1 puerto para Display LCD con 4 bits de datos, Enable y RS.
- 9 pines genéricos de I/O.

En la siguiente Ilustración 10 se muestra la asignación de Pines de la EDU-CIAA

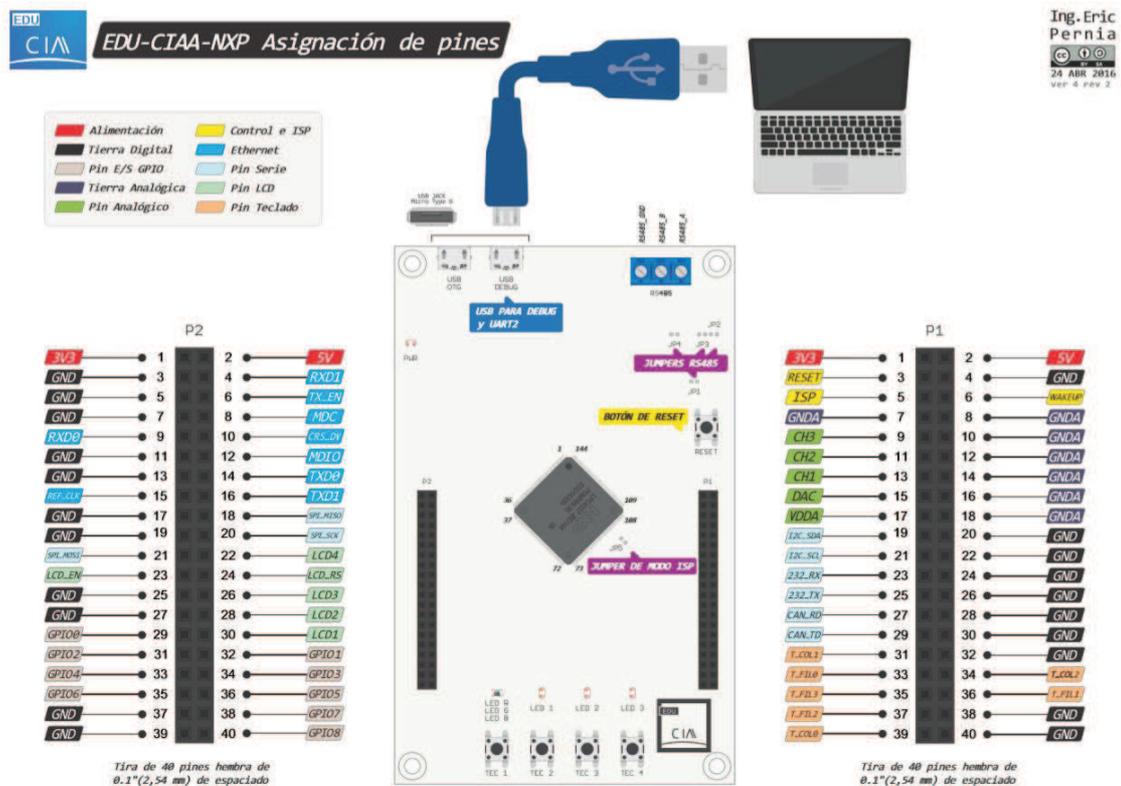


Ilustración 10, Pines de acceso a la placa con sus puertos y configuraciones

Con esta información estudiada, se comenzó a fijar los pines a los cuales darle el uso de comunicación y de accionamiento, junto a los de entrada de estados lógicos.

## Asignación de uso de pines de entrada y salida

Las adaptaciones de las señales fueron hechas en función de las tolerancias de entrada tanto del lado de la fuente como de la placa, ya que una funciona con 3,3v y la otra con 5v. Esto se va a detallar en la parte de diseño de hardware.

Se determinó el uso de entradas y salidas en función de los requerimientos (Entradas y salidas).

También se utilizó un puerto UART Comunicación (luego hay más detalles) para poder conectar y controlar la placa EDU-CIAA+CONTROLADORA con la PC utilizando una comunicación con lenguaje que será presentado posteriormente. Estos vínculos están descriptos en el Tabla 7.

Fuente		PIN PLACA	PUERTO	FUNCION	GPIO	Db25-1	Db25-2	NOMBRE_PIN_CIAA
1	Sentido	2-29	P6_1	0	3-0	<u>2/15</u>	-	GPIO0
	Pulso	2-32	P6_5	0	3-4	<u>3/16</u>	-	GPIO2
	Coarse	2-31	P6_4	0	3-3	<u>4/17</u>	-	GPIO1
	Médium	2-34	P6_8	4	5-16	<u>5/18</u>	-	GPIO4
	Finest	2-33	P6_7	4	5-15	<u>6/19</u>	-	GPIO3
	Nada	2-36	P6_9	0	3-5	<u>7/20</u>	-	GPIO5
	Reset	2-35	P6_10	0	3-6	<u>8/21</u>	-	GPIO6
	Temp	2-38	P6_11	0	3-7	<u>9/22</u>	-	GPIO7
2	Sentido	2-40	P6_12	0	2-8	-	<u>2/15</u>	GPIO8
	Pulso	1-31	P4_0	0	2-0	-	<u>3/16</u>	T_FILE0
	Coarse	1-33	P7_4	0	3-12	-	<u>4/17</u>	T_COL0
	Médium	1-34	P4_3	0	2-3	-	<u>5/18</u>	T_FILE3
	Finest	1-35	P7_5	0	3-13	-	<u>6/19</u>	T_COL2
	Nada	1-36	P4_1	0	2-1	-	<u>7/20</u>	T_FILE1
	Reset	1-37	P4_2	0	2-2	-	<u>8/21</u>	T_FIL2
	Temp	1-39	P1_5	0	1-8	-	<u>9/22</u>	T_COL0
UART	TX2	DEBUG	P7_1	6	-	-	-	USB
	RX2	DEBUG	P7_2	6	-	-	-	USB

Tabla 7, asociación de pines

En la segunda columna tenemos los tipos de funciones y/o variables que tienen las fuentes para ser controladas. Se mantuvo con la misma nomenclatura que la original.

En la Columna 7 y 8 aparecen los pines de uso en el Conector A y su señal negada.

## Comunicación a utilizar

Se determinó una comunicación entre la placa EDU-CIAA y la PC de control a través de un USB que provee la EDU-CIAA. Este USB es una interfaz virtual (VCOM) que es utilizada, con los mismos criterios y especificaciones de uso que una UART. En la Ilustración 11 se muestra el diagrama de comunicación.

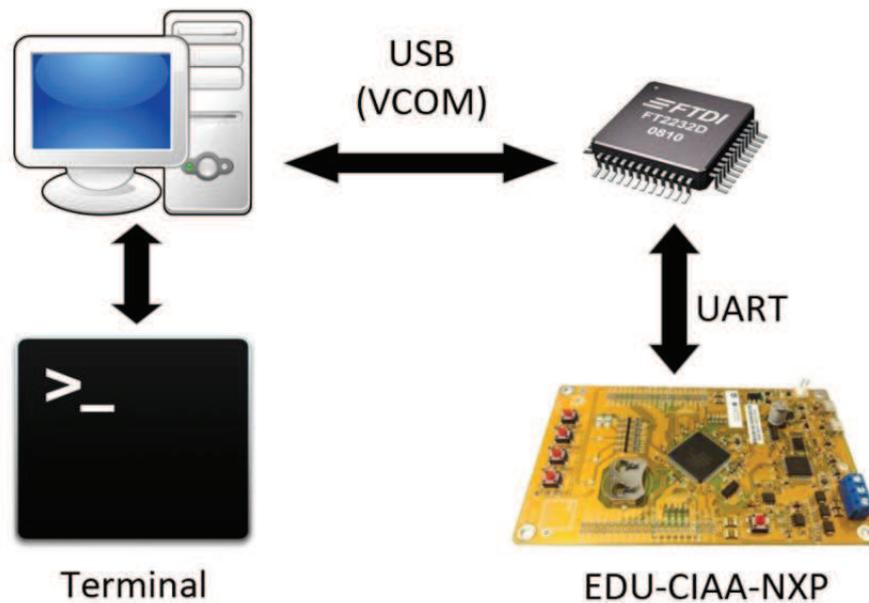


Ilustración 11, Conexión de EDU-CIAA con PC

UART, son las siglas en inglés de Universal Asynchronous Receiver-Transmitter, en español: Transmisor-Receptor Asíncrono Universal, es el dispositivo que controla los puertos y dispositivos serie. Este protocolo de comunicación es muy conocido y popular por su versatilidad y confiabilidad.

Se configuró la UART en 9600 baudios, que son velocidades de transmisión de 9600 Kbps (Kilo bits por segundo). Esto excede completamente en ancho de banda necesario para la transmisión de la información usada, a una tasa de 8bits por carácter cada 200mseg que sería 5 caracteres por segundo.

$$I \frac{Inf}{seg} = 9600 \frac{bits}{seg} \times \frac{1}{8 bits} = 1200 \frac{carateres}{seg}$$
$$1200 \frac{caracteres}{seg} \gg 5 \frac{caracteres}{seg}$$

Esta comunicación se utilizó solo para el enlace entre la placa EDU-CIAA y la PC con el fin de tener una forma de controlar y darle las ordenes de cambio a la placa.

Para lograr la interpretación de los controles, desde la PC hacia la fuente, utilizando la placa EDU-CIAA y a través del USB, se definieron los siguientes comandos en formato ASCII. Se pueden ver en la Tabla 8.

Función	Sentido	LETRA - 1	LETRA - 2
Coarse	Aumento	a	A
Coarse	Decremento	b	B
Medium	Aumento	c	C
Medium	Decremento	d	D
Fine	Aumento	e	E
Fine	Decremento	f	F
Nada	-	g	G
Reset	-	h	H
Temp	-	I	I
Fin TX	-	#	#
Cualquier otra	-	@	@

Tabla 8, Comandos de la placa desarrollada y sus respectivos caracteres

- Se definió una letra para cada estado de selección posible.
- Las letras minúsculas son las de la fuente 1 y las mayúsculas de la fuente 2.
- Dentro de cada tipo de paso (Coarse, Medium, Fine) se determinó también el sentido del mismo.
- La función "Nada" se mantuvo ya que con los comandos originales de la fuente uno podía quedar en una selección que no le da ninguna salida.
- La función reset, es una entrada, donde llega el valor que indica que la fuente está en condiciones de ser operada y desde un valor inicial de cero amperios (0A).
- La función "Temp" también trabaja como entrada y es disparada desde la fuente para poner en un estado de pausa al programa, sumado a esto nos provee una pantalla emergente en el software de control que nos comunique que entro en modo de protección.
- El carácter "#" es utilizado por la función "Fin" para determinar cuándo es el último carácter de llegada, y para obtener una validez de llegada de la información.

- El carácter "@" es devuelto cuando el software envía algún carácter que no está comprendido dentro de las anteriores posibilidades.

## Software (entorno de programación para la placa EDU-CIAA)

Entorno IDE para su programación en lenguaje C. El IDE (inglés. Integrated Development Environment = Entorno de Desarrollo Integrado) provee al desarrollador de Firmware (programador en C/C++) la posibilidad de trabajar en un ambiente amigable y plug & play.

Eclipse es el entorno base utilizado para desarrollo de aplicaciones, es decir, nuestro software IDE.

### Lenguaje C

C es un lenguaje de programación de propósito general que ofrece economía sintáctica, control de flujo y estructuras sencillas y un buen conjunto de operadores.

Este lenguaje ha sido estrechamente ligado al sistema operativo UNIX, puesto que fueron desarrollados conjuntamente. Sin embargo, este lenguaje no está ligado a ningún sistema operativo ni a ninguna máquina concreta. Se le suele llamar lenguaje de programación de sistemas debido a su utilidad para escribir compiladores y sistemas operativos, aunque de igual forma se puede desarrollar cualquier tipo de aplicación.

### Testing Software

Durante el desarrollo del software se realizó el proceso de control de calidad llamado testing.

Las pruebas de software (en inglés software testing) son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada. Es una actividad más en el proceso de control de calidad.

Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software. Dependiendo del tipo de pruebas, estas actividades podrán ser

implementadas en cualquier momento de dicho proceso de desarrollo. Existen distintos modelos de desarrollo de software, así como modelos de pruebas. A cada uno corresponde un nivel distinto de involucramiento en las actividades de desarrollo.

## Proceso del desarrollo Testing

Tenemos el proceso de desarrollo en Cascada, se denomina de este modo, ya que a cada salida de una etapa cae en la siguiente, es decir, las etapas se llevan a cabo una a continuación de la otra. Una de las peculiaridades de este proceso, es que no está previsto volver a una etapa anterior, es decir si se olvidó relevar algún requerimiento al comienzo, no tiene una alternativa para considerar este caso. Este proceso supone cada etapa independiente de las etapas anteriores.

También tenemos el proceso Incremental, se tiene las mismas etapas que en el Proceso de Desarrollo en Cascada, sin embargo, en este proceso, la etapa de relevamiento se divide en distintos sub conjuntos, y cada uno de estos sub conjuntos se construye de la misma forma que con el ciclo de vida en cascada. Se van desarrollando por partes que luego se integran, una vez finalizadas las mismas.

Otro Proceso de Desarrollo que tenemos es el Iterativo, en este tenemos las mismas etapas de desarrollo que los procesos anteriores, pero trabajamos sobre el todo, no necesariamente conocemos el comienzo todos los detalles del producto que queremos construir.

Y por último tenemos el Proceso de Desarrollo Ágil (Enfoque para la toma de decisiones en los proyectos de software), este es un proceso Iterativo e Incremental, se caracteriza por contar con iteraciones cortas y por no tener fases lineales, tipo Cascada en cada iteración. Existen distintas metodologías Ágiles, que entre las más conocidas y utilizadas encontramos "Scrum (Trabajo colaborativo)" y "Extreme Programming ("XP" metodología de desarrollo de software de Ingeniería)".

### 1. Pruebas estáticas

Son el tipo de pruebas que se realizan sin ejecutar el código de la aplicación.

Puede referirse a la revisión de documentos, ya que no se hace una ejecución de código. Esto se debe a que se pueden realizar "pruebas de escritorio" con el objetivo de seguir los flujos de la aplicación.

## 2. Pruebas dinámicas

Todas aquellas pruebas que para su ejecución requieren la ejecución de la aplicación. Las pruebas dinámicas permiten el uso de técnicas de caja negra y caja blanca con mayor amplitud. Debido a la naturaleza dinámica de la ejecución de pruebas es posible medir con mayor precisión el comportamiento de la aplicación desarrollada.

### Diagrama de flujo de software de control en la EDU-CIAA

El funcionamiento del programa se muestra en los diagramas de flujo de las Ilustración 12, Parte 1 de 3 Ilustración 12, Ilustración 13, Ilustración 14.

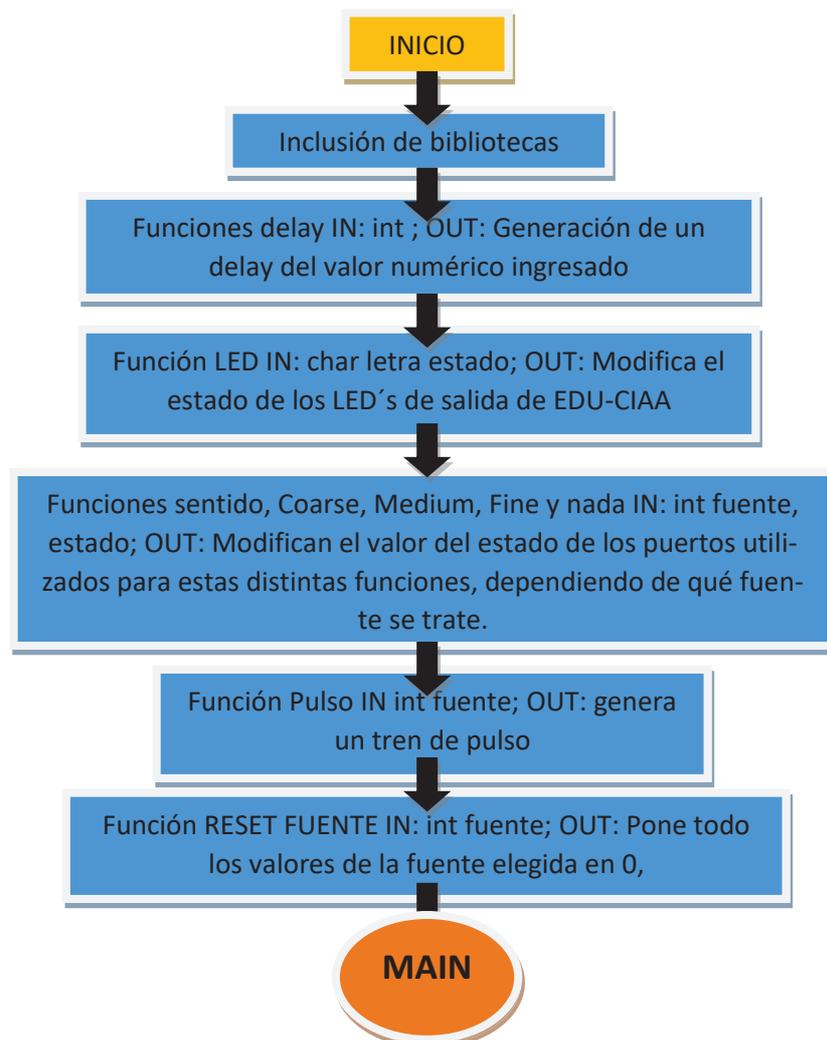
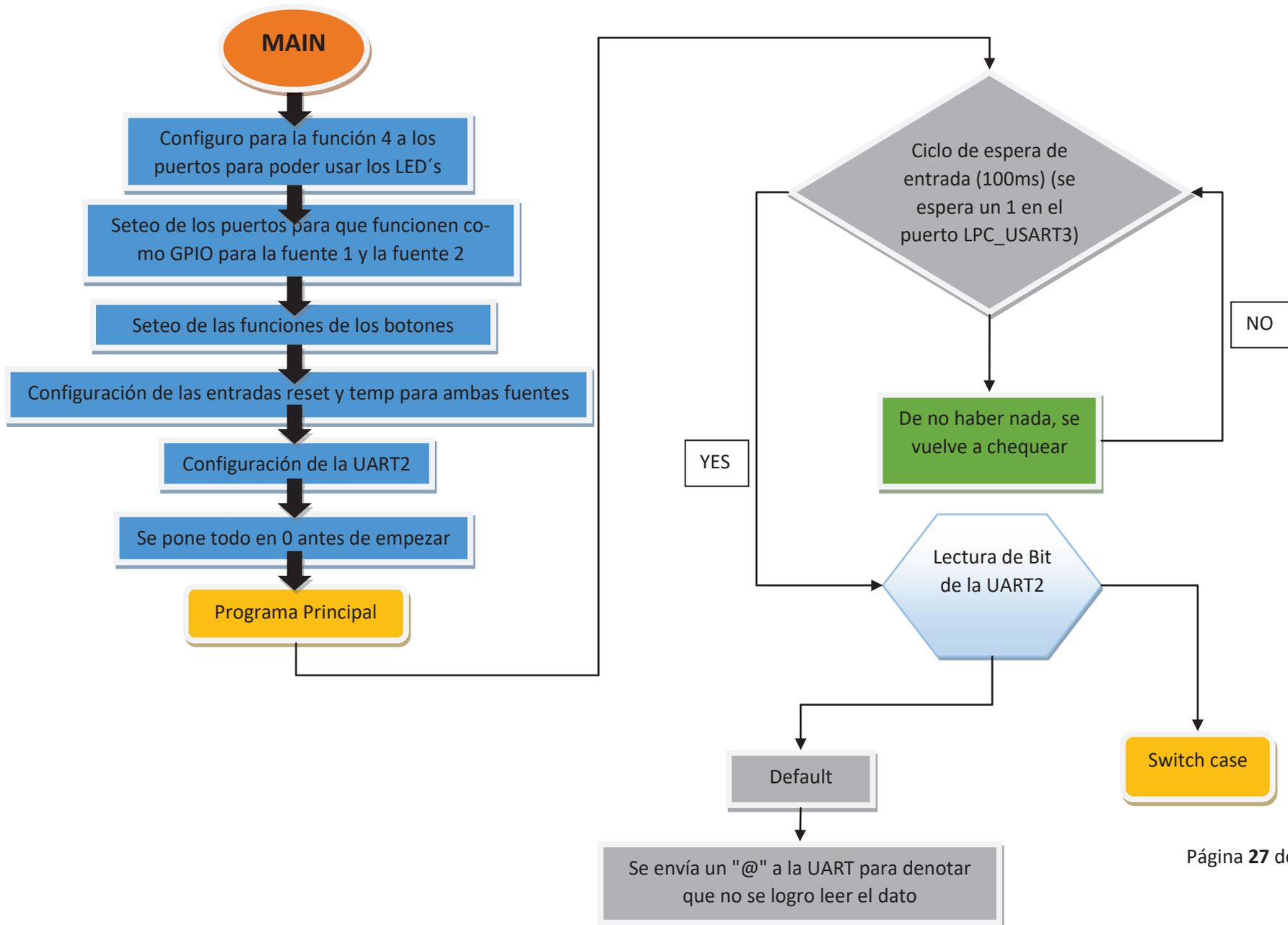


Ilustración 12, Parte 1 de 3

Ilustración 13, Parte 2 de 3



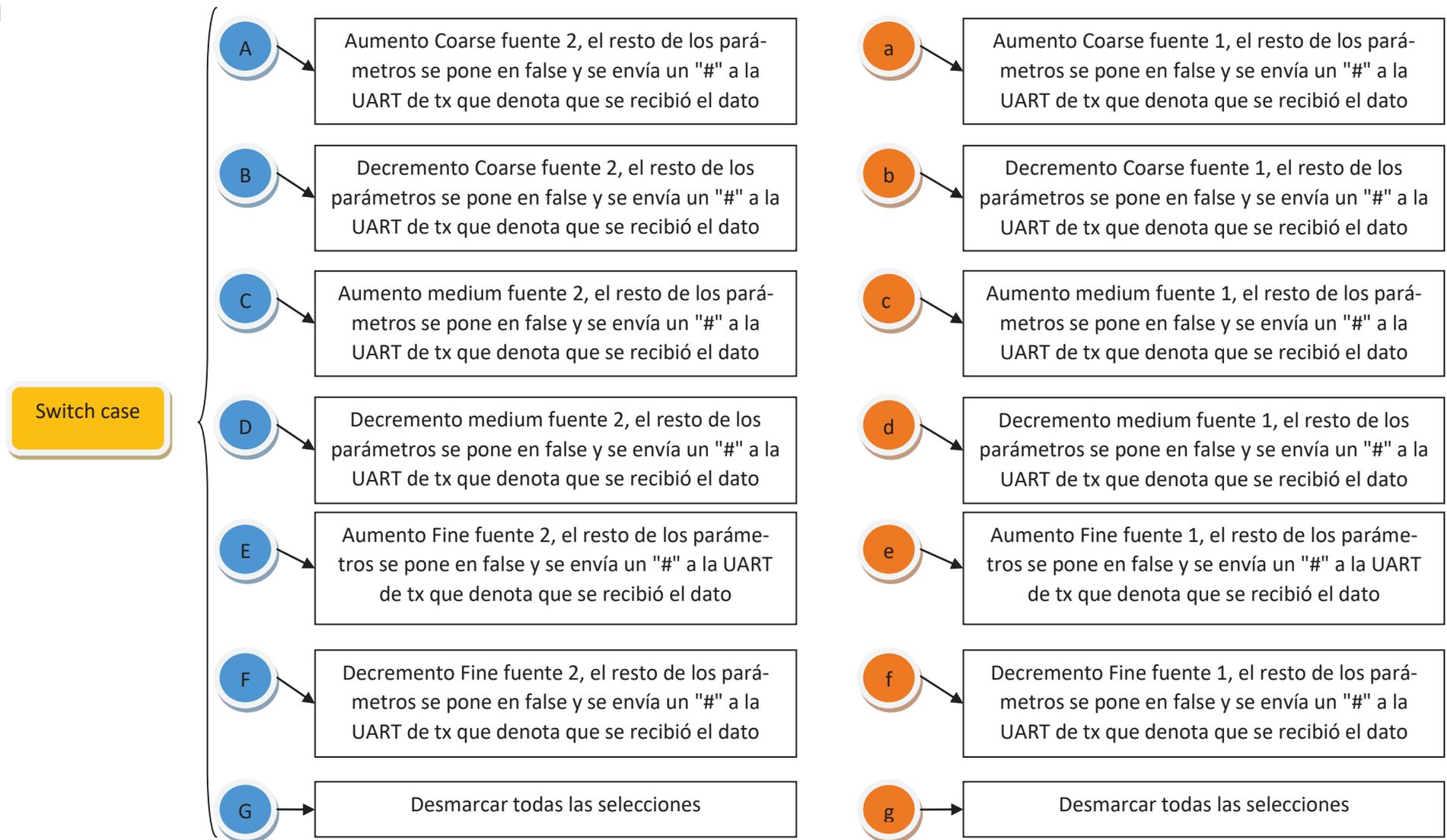


Ilustración 14, Parte 3 de 3

## Hardware (Placa interfaz entre EDU-CIAA y fuente)

### Detalle del diseño de la placa de control

El relevamiento de las señales del control original de las fuentes fueron censadas/medidas en su total funcionamiento.

A partir de los valores obtenidos de las mediciones, se llevo a cabo un entendimiento de los datos recolectados donde se llegó a la conclusión de cómo es la operación de la fuente durante el funcionamiento.

Los valores de selección de los estados fueron generados con salidas Digitales de la placa. Estos eran conectados a un transistor en modo emisor común en modo saturación, para poder de esta forma excitar el led de un opto-acoplador para generar una señal con el uno lógico (1) del nivel correspondiente y su negada en contrapartida.

En la Ilustración 15 se muestra la configuración de los transistores y opto-acopladores con la base de ellos conectados a la salida Digital de la placa a través de la resistencia R25 (Ejemplo).

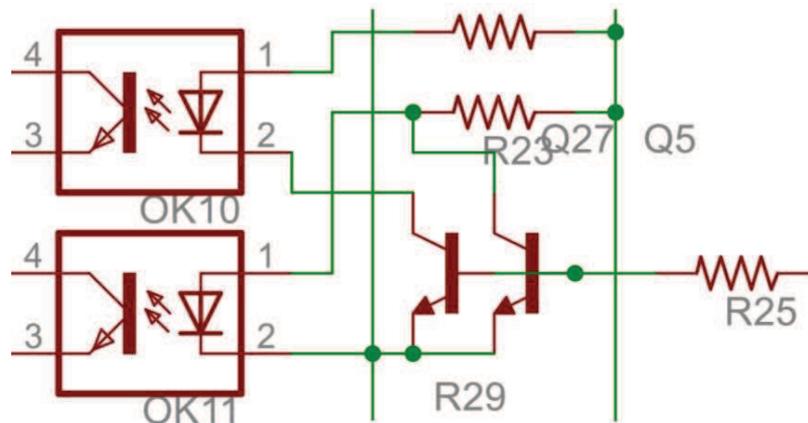


Ilustración 15, Salidas digitales

### Cálculos para las polarizaciones de componentes

Primero que nada, tenemos que considerar que el transistor que seleccionemos tiene una beta ó  $h_{fe}$  determinada. Este valor de ganancia de corriente lo podemos encontrar en la hoja de datos del BC548 (Adjunta).

$V_{CC} = 3.3V$  – La fuente de alimentación.

$V_{BB} = 3.3V$  – La fuente de alimentación de control (EDU-CIAA).

$hfe = 200$  – Ganancia de corriente del transistor.

El transistor se encuentra en la región de corte cuando  $I_C = 0$ . Para dejar en cero la corriente de colector, se requiere tener en cero la corriente de base  $I_B$ . Ésta será cero cuando  $V_{BB} = 0$ . El transistor se encuentra en saturación cuando la tensión de colector emisor sea de cero  $V_{CE} = 0$ . Cuando  $V_{CE} = 0$ , podemos determinar el valor de la corriente de saturación de colector  $I_{Csat} = \frac{V_{CC}}{RC}$ .

$$I_{Csat} = 20mA$$

Para

$$RC = \frac{V_{CC}}{I_{Csat}} = \frac{3.3V}{20mA} = 165\Omega$$

Entonces

$$I_{Csat} = H_{fe} \times I_B$$

Por lo tanto:

$$I_B = \frac{I_{Csat}}{H_{fe}} = \frac{20mA}{200} = 1mA$$

Para asegurar el estado de saturación, evitando efectos intrínsecos del transistor podemos multiplicar la ganancia por 5.

$$I_B = 5 \times \frac{I_{Csat}}{H_{fe}} = 5 \times \frac{20mA}{200} = 5 \times 1mA = 5mA$$

Considerando la malla de la entrada, podemos determinar el valor adecuado para  $R_B$ .

$$V_{BB} = I_B \times R_B + V_{BE}$$

$$R_B = \frac{V_{bb} - V_{BE}}{I_B}$$

Sustituyendo:

$$R_B = \frac{3.3V - 0.7V}{5mA} = 520\Omega$$

Por lo tanto la configuración nos queda:

$$R_B = 520\Omega$$

$$RC = 165\Omega$$

Como detalle se midió físicamente la tensión base emisor  $V_{BE}$  para que el análisis analítico sea el correcto arrojando como resultado lo calculado.

### Región de corte y saturación

Para determinar la corriente de saturación, consideramos la tensión colector emisor de la malla de salida igual a cero. Por lo tanto:

$$V_{CC} = I_C \times RC + V_{CE} \rightarrow V_{CC} = I_C \times RC + 0$$

$$I_C = \frac{V_{CC}}{RC}$$

Para determinar el corte, consideramos que la corriente de base es igual a cero, por lo tanto la corriente de colector es igual a cero:

$$V_{CC} = I_C \times RC + V_{CE} \rightarrow V_{CC} = 0 \times RC + V_{CE}$$

$$V_{CE} = V_{CC}$$

Con estos dos puntos determinamos la recta de carga del transistor. La región central se llama, región activa. Las regiones del extremo son regiones de saturación y de corte. Ilustración 16

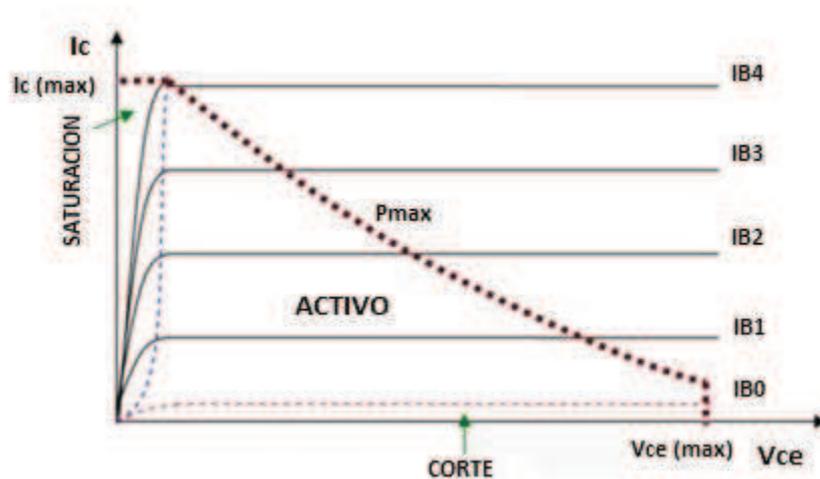


Ilustración 16, Curva de tensión  $V_{CE}$  vs  $I_C$

El valor Reset y Temperatura (Ilustración 17) son capturados por una entrada analógica que es acondicionada con un Transistor y un opto-acoplador.

El transistor está en una configuración buffer para no consumir corriente de la parte de control de la fuente y el opto-acoplador sirve para aislar de forma galvánica.

Además, el transistor se encuentra trabajando en corte y saturación. Con la salvedad que la excitación de la base es por medio de 5V.

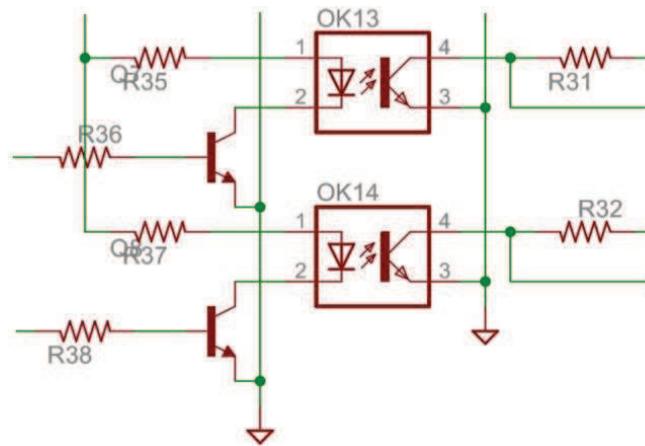


Ilustración 17, Entradas

### Aislación eléctrica

Dicha placa se realizó con la idea de que sea completamente aislada de forma eléctrica entre la EDU-CIAA y la fuente, evitando problemas de falsas tierras.

Un Opto-acoplador es un circuito integrado muy básico, compuesto generalmente por un diodo LED y un fototransistor unidos de tal forma que cuando una señal eléctrica circula a través del LED haciendo que brille, la luz que este emite es recibida por la base del fototransistor, que empieza a actuar en modo saturación. Vea la Ilustración 18

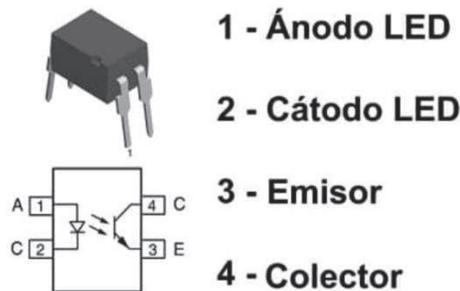


Ilustración 18, Opto-acoplador

Se puede utilizar este dispositivo a modo de interfaz entre dos circuitos, de tal forma que quedarían unidos ópticamente, lo que a efectos de protección del circuito, se traduce en colocar una resistencia de un valor muy alto ( $\gg M\Omega$ ), lo que lo hace especialmente útil para proteger contra los picos de tensión.

Como la luz que emite el LED varía en función de la tensión y la corriente que circulan por él y esta luz a su vez modifica el comportamiento del transistor, la

señal eléctrica a la salida (en el transistor) dependerá de la señal que tengas a la entrada, es decir, de cómo ataques el LED.

En la práctica esto se traduce en que si regulas bien el comportamiento de tu Opto-acoplador (y los componentes necesarios para su funcionamiento), puedes conseguir que tu circuito aislador sea invisible en la práctica, es decir, no cambie el comportamiento de tu circuito global. Otra posibilidad es aprovechar la ganancia que te proporciona el fototransistor y lo utilices para amplificar la señal.

## Diseño de emulador de pulsos de encoder

Encoder: Es un potenciómetro por pasos con una cantidad de 128 pasos por vuelta y esta genera a su salida una tren de pulsos que dependiendo de la velocidad y del ángulo de giro, nos provee una cantidad de pulsos por unidad de tiempo. También, dependiendo del sentido de giro, un valor (sentido) es el encargado de dar este dato de forma digital, siendo 1 (5v) para sentido de las agujas del reloj y 0 (0v) en sentido contrario.

La implementación de esto fue a través del código del programa de la placa EDUCIA que dependiendo del carácter que le llega de la PC, esta lo transforma a la salida correspondiente del tren de pulsos del Conector A, la señal que contiene la información de la cantidad de pasos a realizar y en el pin del "sentido" la información de en cual sentido.

## Diseño y fabricación

### Criterios de diseño eléctrico y físico

Para lograr la adaptación de la placa controladora EDU-CIAA y las señales que ingresan a la fuente, fueron acondicionadas con niveles de tensión TTL (0-5V) a través de componentes electrónicos a fines. Utilizando acoplamiento óptico, se aisló la parte de control con la fuente para tener una total independencia de tensiones.

El circuito de adaptación y control fue diseñado, para ser de fácil remoción y acoplamiento con la EDU-CIAA. El mismo se puede ver en la siguiente Ilustración 19.

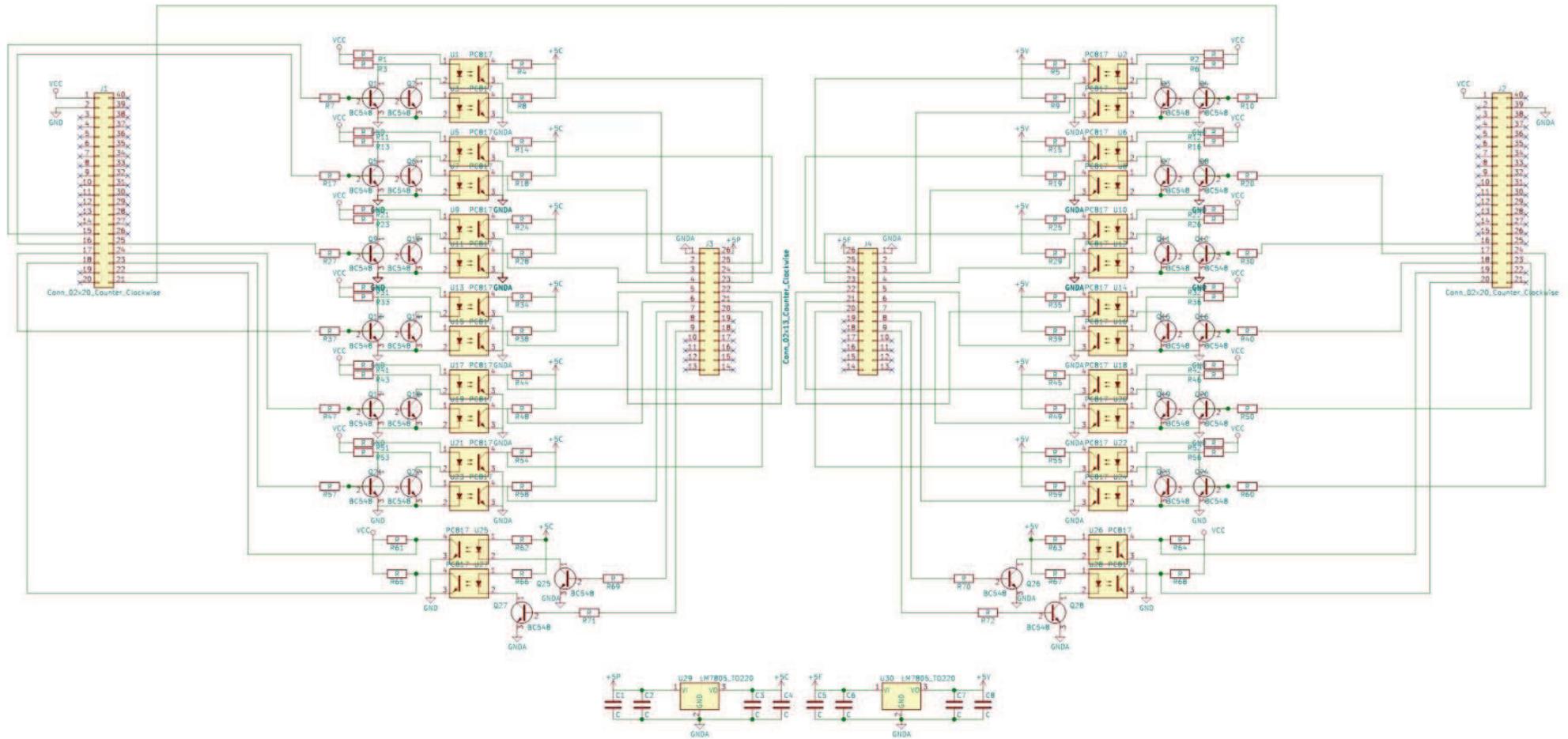


Ilustración 19, Esquemático de placa

En la figura anterior se pueden ver todos los componentes usados junto con sus conexiones y los nombres de los mismos. Esto es el Esquemático y da el inicio del desarrollo en el programa de diseño. Arriba se puede ver la fuente de tensión que regula la misma para el lado de la fuente de corriente, ya que para el lado de la EDU-CIAA, la misma tiene su fuente incluida en la placa.

### Software de diseño

Todo el desarrollo de la placa fue llevado a cabo con software libre (KICAD).

KiCad es un paquete de software de código abierto (Libre) para la automatización del diseño electrónico (EDA). Los programas manejan Captura esquemática y Diseño de PCB con salida de Gerber. La suite se ejecuta en Windows, Linux y macOS y está licenciada bajo GNU GPL v3.

Este software permite el diseño/planeamiento del esquemático.

### Diseño de Placa (Circuito impreso)

Una vez realizado se puede continuar con el diseño de la placa física en un estilo CAD basándose en el circuito esquemático.

Para tal circuito fueron utilizados todos componentes de comercialización nacional, para no depender de las importaciones. También se pensó en la facilidad al momento de hacer el "PCB" (Placa de circuito impreso) con componentes "Through-Hole" (agujero pasante) para una facilidad al momento de fabricarla y soldarla.

En las siguientes imágenes (Ilustración 20, Ilustración 21, Ilustración 22, Ilustración 23) se pueden ver el Ruteo (Trazado de pista) sobre la placa en dos tonos distintos ya que se realizó en doble faz. Esto quiere decir que es una placa que tiene ruteo de ambos lados de la misma. Estas caras se las identifica como TOP y BOTTON (Superior, Inferior)

A ambas caras de la placa se les genero un plano asociado a GND de la EDU-CIAA para evitar posibles interferencias.

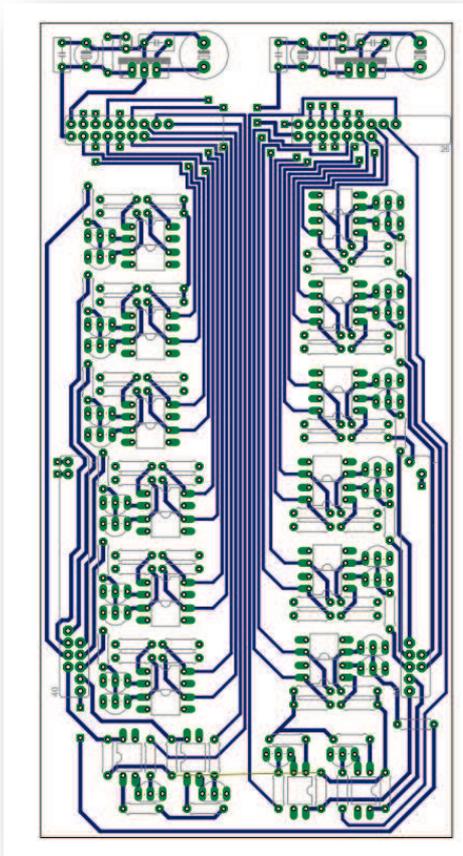


Ilustración 21, Capa superior

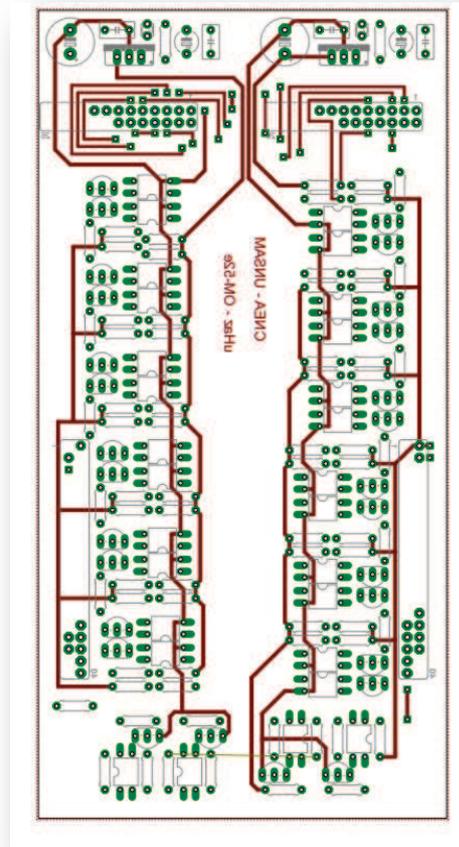


Ilustración 20, Capa inferior

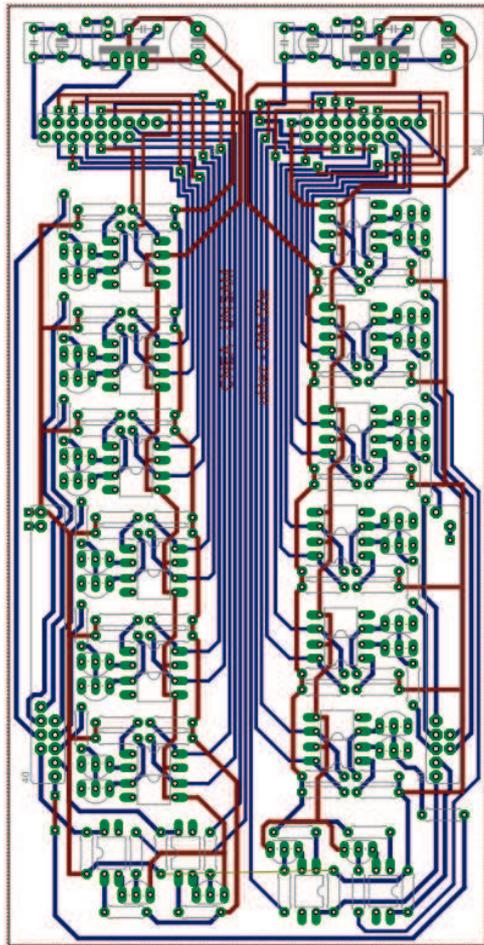


Ilustración 23, Capa Superior TOP

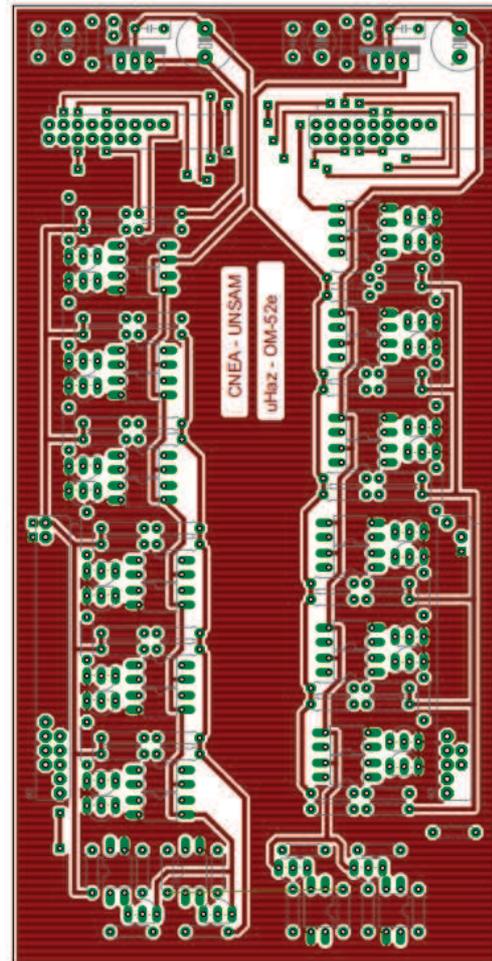


Ilustración 22, Capa inferior TOP

En las siguientes ilustraciones (Ilustración 24, Ilustración 25, Ilustración 26, Ilustración 27, Ilustración 28) se puede observar las vistas en 3D “dimensiones” de la placa desarrollada.

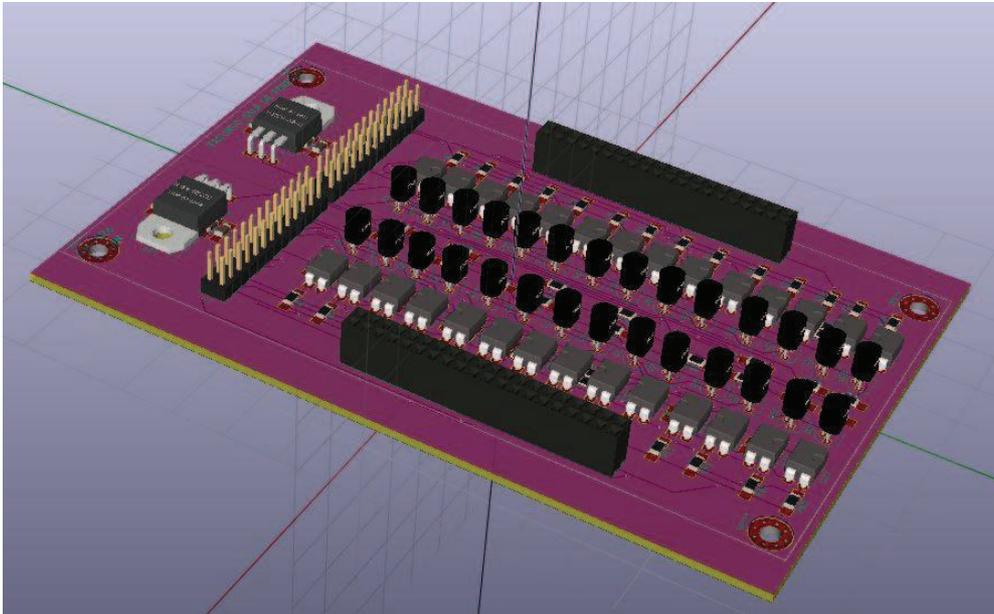


Ilustración 24, Modelo 3D

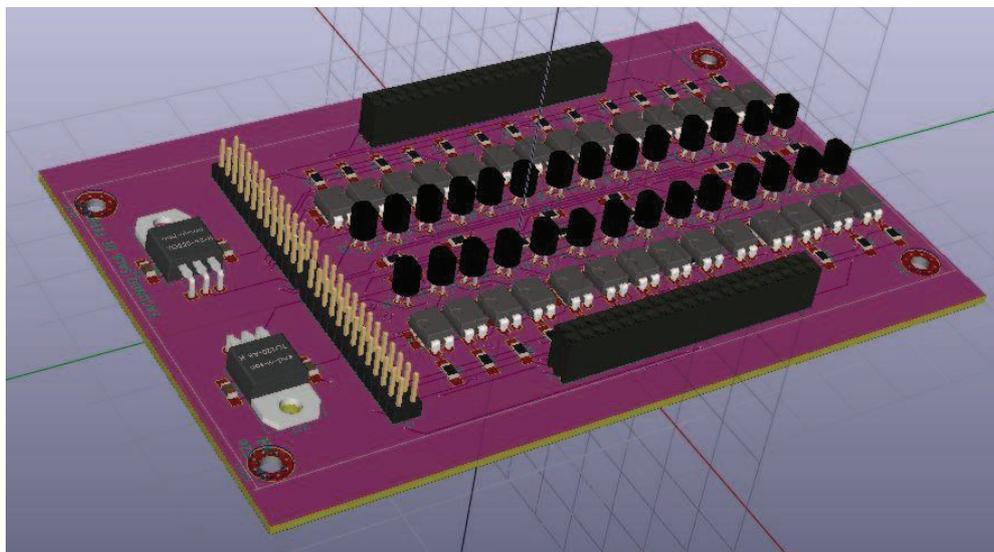


Ilustración 25, Modelo 3D

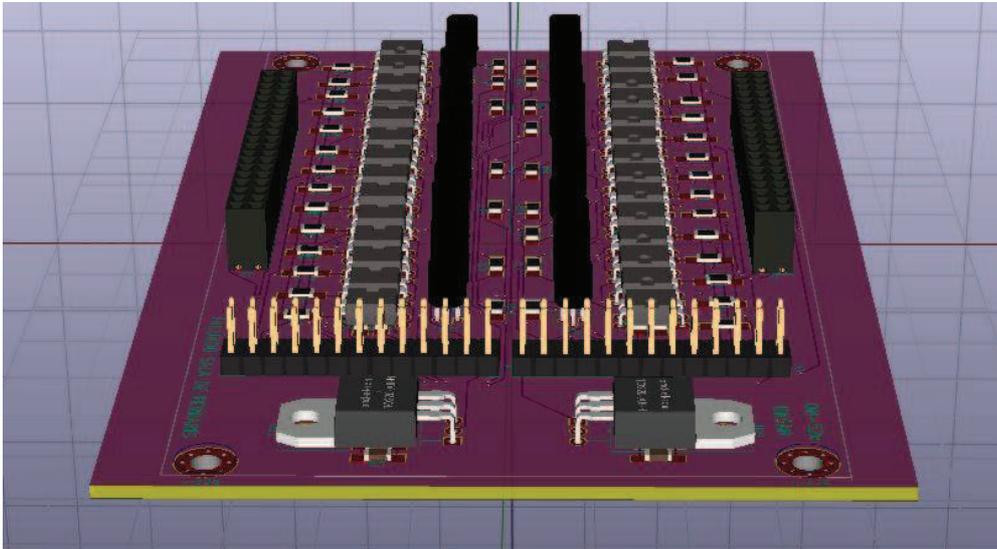


Ilustración 26, Modelo 3D

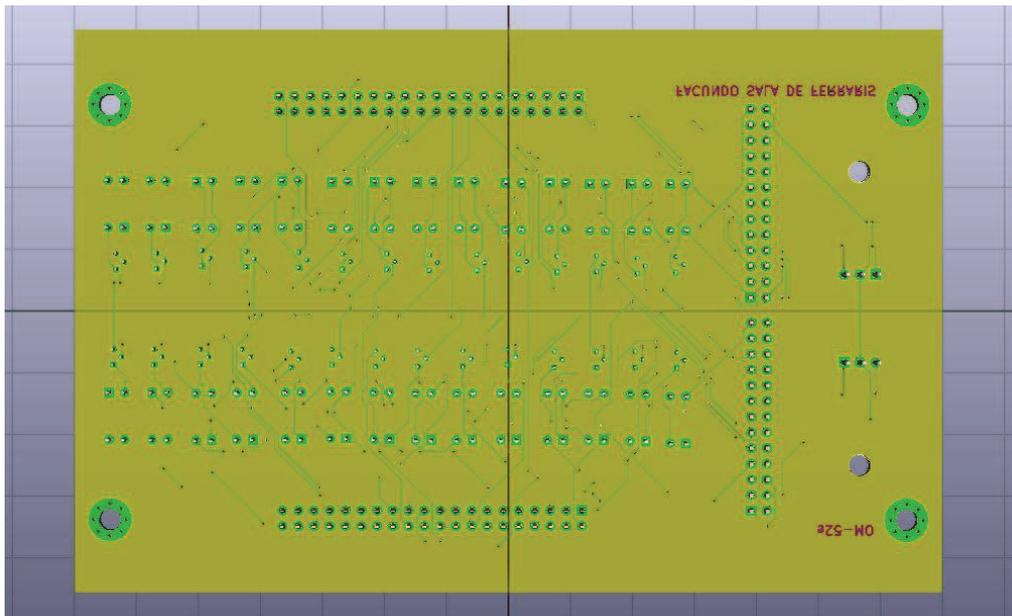


Ilustración 27, Modelo 3D Cara inferior

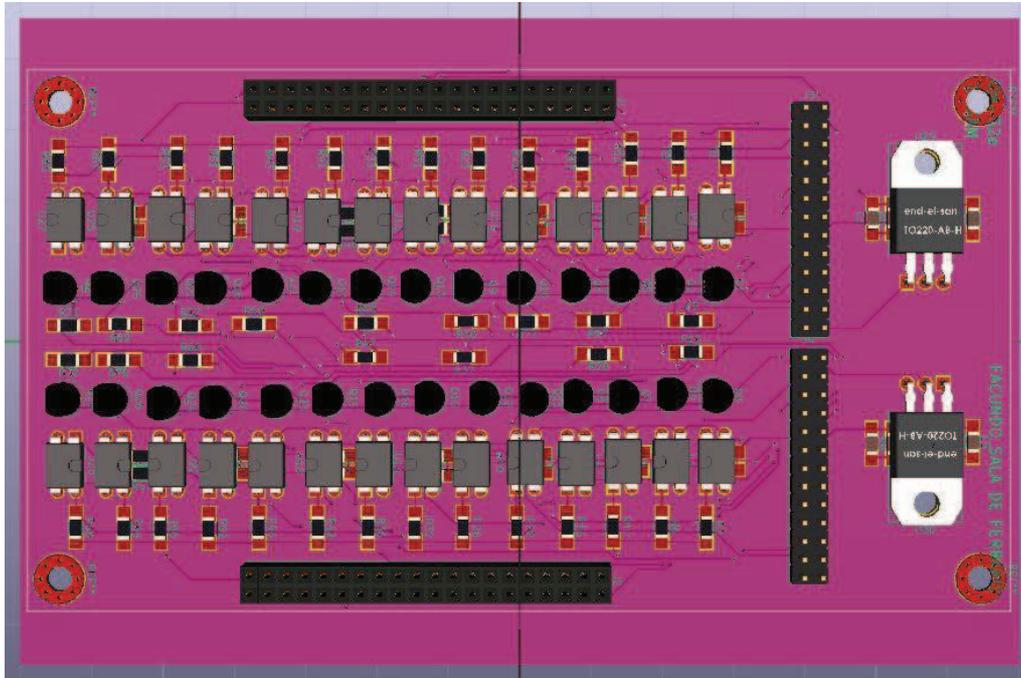


Ilustración 28, Modelo 3D Cara Superior

### Placa y modulo de control terminado

La Ilustración 29 muestra el dispositivo diseñado de control de fuente de corriente por dentro. En esta se aprecia la EDU-CIAA y la placa interface de control de la fuente conectada por debajo de la anterior. También se aprecian algunas de las conexiones de la placa como el USB y los conectores que vas hacia los Db25. En la Ilustración 30 podemos verlo armado con su cable conexión USB y sus dos conectores de la fuente.



Ilustración 29, Muestra el modulo de control con su placa EDUCIA y de Control



Ilustración 30, Muestra el modulo de Control terminado con su conexión externa para cada fuente y el conector USB

## Software de control HMI

### Entorno de desarrollo usado

Las HMI (también conocidas como interfaces gráficas de usuario o interfaces de usuario) permiten un control sencillo (con uso de mouse) de las aplicaciones de software.

GUIDE (entorno de desarrollo de GUIDE) proporciona herramientas para diseñar interfaces de usuario para Apps (Aplicaciones) personalizadas. Mediante el editor de diseño de GUIDE, es posible diseñar gráficamente la interfaz de usuario. GUIDE genera entonces de manera automática el código de MATLAB para construir la interfaz, la cual se puede modificar para programar el comportamiento de la app.

### Interface de control de fuente

En la Ilustración 31 se muestra una captura de pantalla de la aplicación de control.

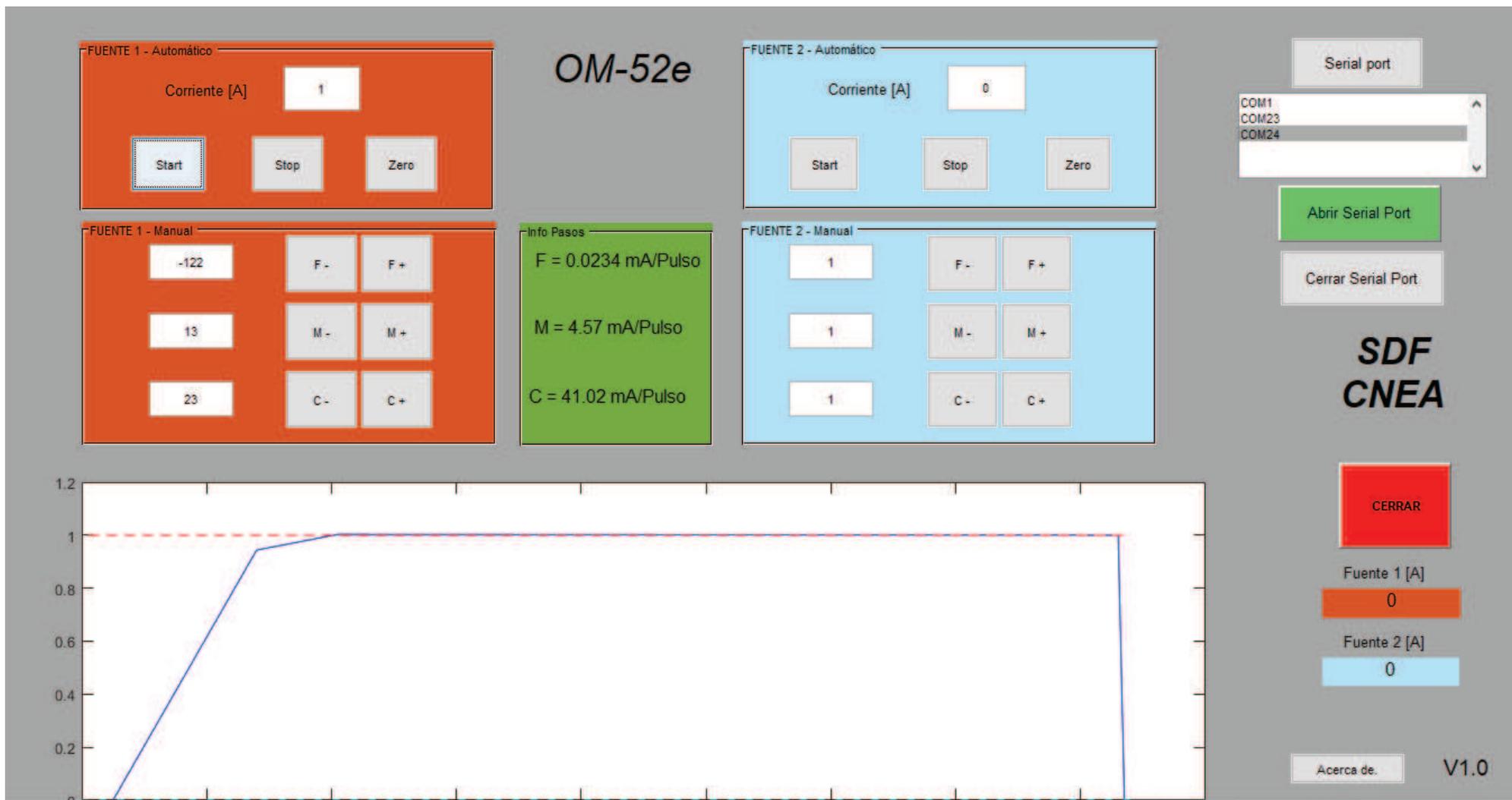


Ilustración 31, Panel de control V 1.0

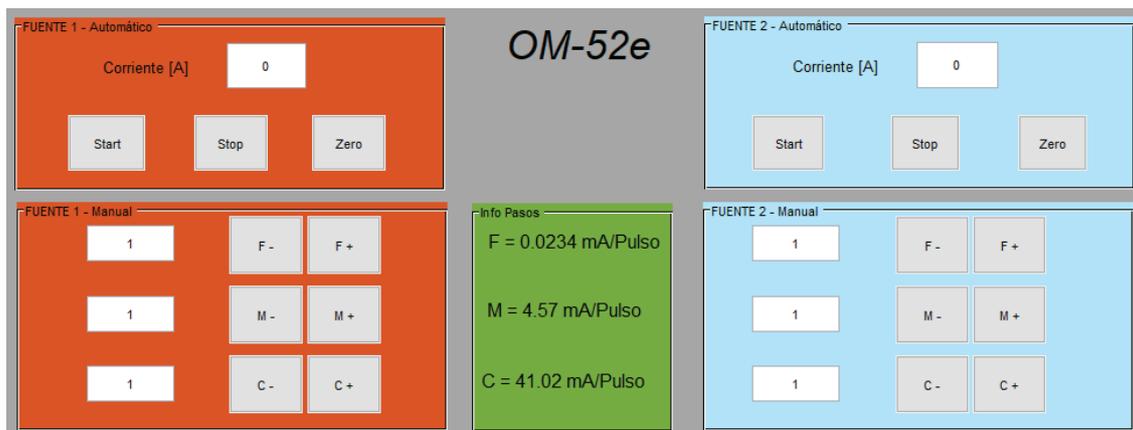


Ilustración 32, Mandos de set de corriente

Se muestra en la Ilustración 32 los comandos de las fuentes de forma independiente, su seteo de corriente deseada de alcance automática y su control manual

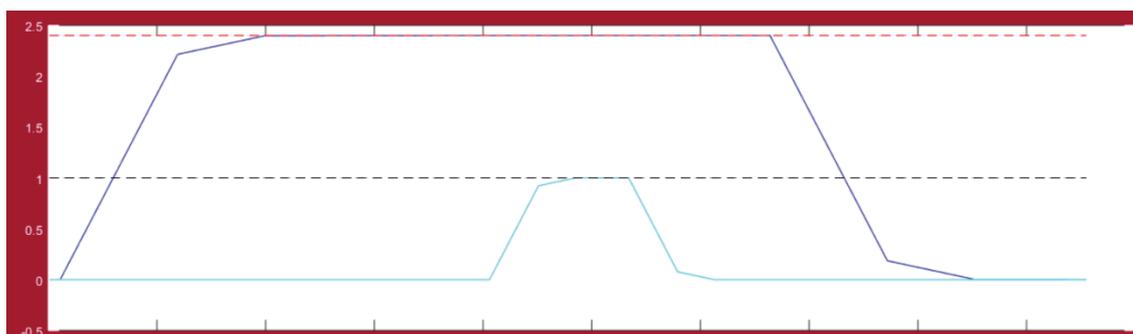


Ilustración 33, Grafica temporal de Corriente

En la Ilustración 33 podemos ver la grafica temporal de Corriente en función de Pasos. Ambas graficas se exhiben sobre el mismo grafico en dos colores diferentes. Para esta prueba se le ordenó a la fuente 1 una corriente de 2.5 A, mientras que a la fuente 2 se le seteo una corriente de 1 A. Ambas después de terminada la prueba, fueron devueltas al valor 0 (CERO).

A continuación se muestran los cuadrantes de información de estado de corriente a la que se encuentra cada fuente (Ilustración 34).



Ilustración 34, Estado de corriente actual

En la siguiente imagen (Ilustración 35) se puede apreciar la forma de selección del puerto Serial, este número depende de la PC y de la cantidad de puertos que tenga conectados.

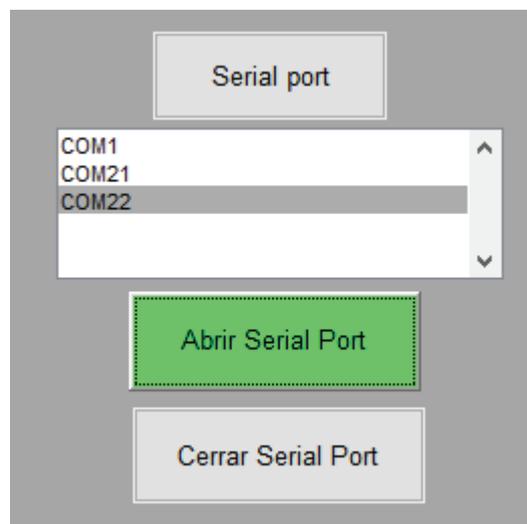


Ilustración 35, Selección de puerto



Ilustración 36, Salida del programa

## Pruebas de funcionamiento

Las primeras pruebas se realizaron sin la fuente. Se utilizó un analizador lógico para capturar en tiempo real todas las salidas de la placa desarrollada (Salida) y del software HMI (Entrada). Se conectó el analizador lógico a las salidas del conector A para capturar los pulsos en ausencia de la fuente. De esta manera analizaremos el funcionamiento.

En la Tabla 9 se muestran las señales que se capturaron con su respectiva función asociada a ese pin y el numero de pin asociado al Conector A.

	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
Fun- ción	Senti- do	<u>Senti- do</u>	Pul- so	Coar- se	<u>Coar- se</u>	Me- dium	<u>Me- dium</u>	Fi- ne	<u>Fi- ne</u>	Na- da	<u>Na- da</u>
Pin Db25	2	15	3	4	17	5	18	6	19	7	20

Tabla 9, Señales representadas en el analizador lógico

En la Ilustración 37 e Ilustración 38 podemos observar la señal capturada del analizador durante una corrida de 28 segundos, donde se le ordenó un crecimiento de 1A.

Al seleccionar la corriente, el HMI nos determina los pulsos para cada paso. Para el Coarse unos 45, en Medium 34 y en Fine -46. Esto quiere decir que según el signo que antecede al número de pulsos, nos determina si es de incremento o decremento.

La cuenta que determina estos valores son:

$$\begin{aligned} Coarse &= \text{Redondeo} \left( 0.92 \times \frac{1A}{5.25 \frac{A}{Vuelta}} \times \frac{128 \text{pulsos}}{Vuelta} \right) \\ &= 45 \text{ pulsos} \end{aligned}$$

$$\begin{aligned} Medium &= \text{Redondeo} \left( 1A - \left( \frac{5.25A}{\frac{128 \text{pulsos}}{Vuelta}} \times Coarse \right) \times \frac{0.98}{0.585A} \times \frac{128 \text{pulsos}}{Vuelta} \right) \\ &= 34 \text{ pulsos} \end{aligned}$$

$$Fine = \text{Redondeo} \left( 1A - \left( \frac{5.25A}{\frac{128 \text{pulsos}}{Vuelta}} \times Coarse \right) - \left( \frac{0.585A}{\frac{128 \text{pulsos}}{Vuelta}} \times Medium \right) \times \frac{1}{0.003A} \times \frac{128 \text{pulsos}}{Vuelta} \right) = -46 \text{ pulsos}$$

MATLAB

```
datos.S_Coarse=5.25;
datos.S_Medium=0.585;
datos.S_Fine=0.003;

datos.iter_c=ceil(0.92*datos.I_deseada_1/datos.S_Coarse*256);
datos.iter_m=ceil((datos.I_deseada_1-(datos.S_Coarse/256*datos.iter_c))*0.98/datos.S_Medium*256);
datos.iter_f=ceil((datos.I_deseada_1-(datos.S_Coarse/256*datos.iter_c)-
datos.S_Medium/256*datos.iter_m)*1/datos.S_Fine*256);

% ceil es una función de redondeo
```

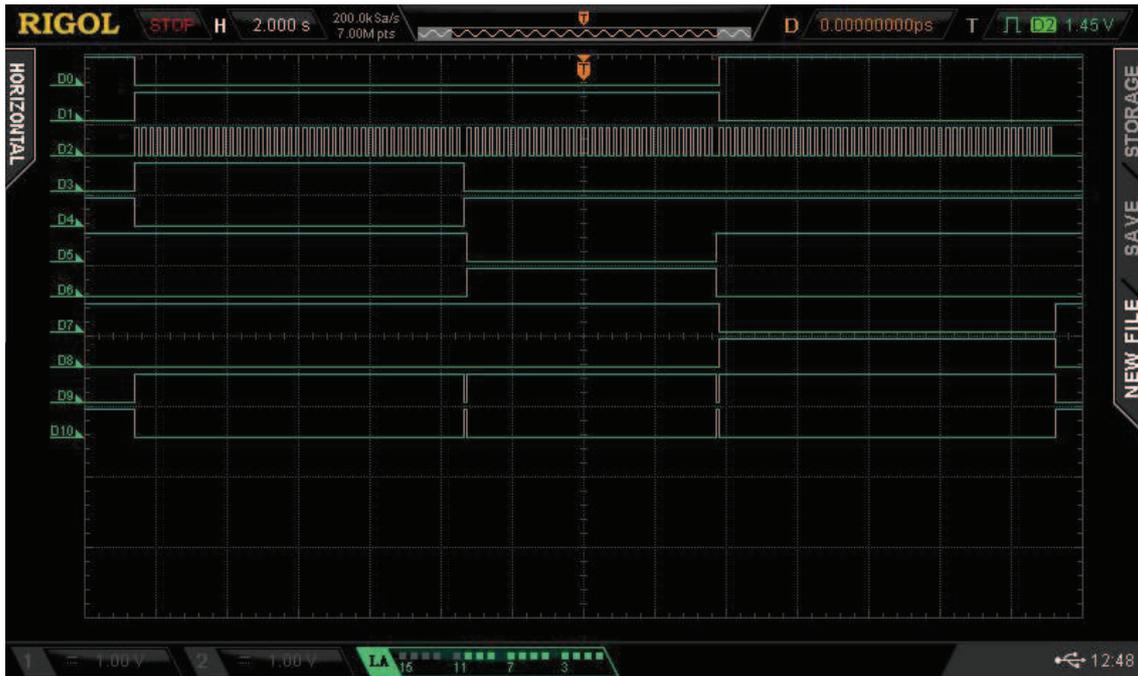


Ilustración 37, Captura de analizador lógico de fuente 1

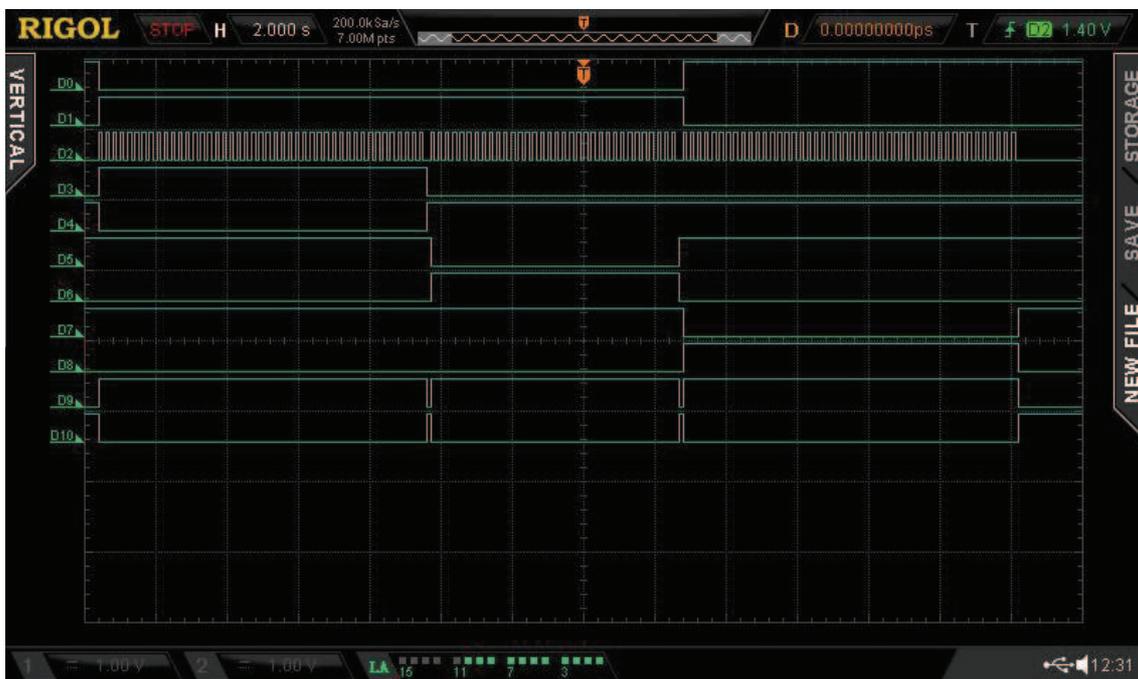


Ilustración 38, Captura de analizador lógico de fuente 2

Vemos en la Ilustración 37 e Ilustración 38 que la cantidad de pulsos que se ven en D2 se corresponden con la cantidad de pulsos que se describieron anteriormente. También se observa como la línea D0 más la negada D1, cambian de sentido al llegar a la etapa de Fine ya que su signo era diferente al que teníamos para Coarse y Medium.

En la D3 y su señal análoga negada D4, tenemos la selección de Coarse. A su vez, para Medium tenemos la señal D5 y la señal negada D6, Mientras que para el Fine contamos con D7 y su negada D8.

En la señal D10 tenemos la selección de "Nada" que ocurre cada vez que termina una de las funciones o cuando el sistema se encuentra en reposo.

## Conclusiones

El sistema de control de fuente se diseño sin realimentación ya que no es viable, dado el tipo de corriente que maneja. Se debería censar la misma para una realimentación de estado. Esto me obligo a realizar un fuerte análisis del sistema en su conjunto, para obtener y estimar minimizando el de error. Esta aproximación se realizo, relevando la curva de salida (Corriente vs señal de control), el estudio y corrección de las alinealidades de la fuente.

Se pudo poner en práctica varios conceptos de electrónica analógica para el diseño de la placa de control.

También se implemento programación en lenguaje C y JAVA.

Para el uso de software de diseño de Circuito impreso, se utilizó una herramienta de distribución gratuita (KiCad) para poder continuar con la línea de desarrollo de la placa EDUCIAA.

Se puede considerar que los tiempos establecidos para el proyecto fueron estimados en forma correcta. Los riesgos que se tuvieron en cuenta no fueron suficientes ya que el mayor riesgo (exámenes adeudados) de todos no fue contemplado, hecho por el cual se demoró la entrega del mismo y el cumplimiento del plazo.

La continua participación del operador de la fuente fue indispensable para el entendimiento y el alcance del proyecto, como de sus requisitos.

Dado que fue un proyecto heredado de una materia anterior (digitales 2), fue indispensable y muy valorada la documentación de aquel entonces para continuar con dicho proyecto.

## Bibliografía

*Libro, Electrónica analógica 2da Edición (Fernando Ramírez Rojas)*

*Libro, An Engineers Guide to MATLAB 3rd Edition (The MathWorks)*

[https://en.wikipedia.org/wiki/Bipolar\\_junction\\_transistor](https://en.wikipedia.org/wiki/Bipolar_junction_transistor)

<http://www.microbeams.co.uk/om52e.html>

<http://www.ingmecafenix.com/automatizacion/encoder/>

<http://kicad-pcb.org/>

<http://www.proyecto-ciaa.com.ar/devwiki/doku.php?id=desarrollo:edu-ciaa:edu-ciaa-nxp>

<https://es.wikipedia.org/wiki/JavaScript>

<https://es.mathworks.com/discovery/matlab-qui.html>

[https://es.wikipedia.org/wiki/Universal\\_Asynchronous\\_Receiver-Transmitter](https://es.wikipedia.org/wiki/Universal_Asynchronous_Receiver-Transmitter)

<https://ferqarciar.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>

<http://mrelbernitutoriales.com/corte-y-saturacion-bjt/>

[https://es.wikipedia.org/wiki/Pruebas\\_de\\_software](https://es.wikipedia.org/wiki/Pruebas_de_software)

## Referencias externas

CNEA. (s.f.). [https://www.argentina.gob.ar/comision-nacional-de-energia-atmica.](https://www.argentina.gob.ar/comision-nacional-de-energia-atmica)

EDU-CIAA. (s.f.). [http://www.proyecto-ciaa.com.ar/.](http://www.proyecto-ciaa.com.ar/)

JAVA. (s.f.). [https://www.java.com/.](https://www.java.com/)

M4, A. c. (s.f.). [https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/lpc-cortex-m-mcus:LPC-ARM-CORTEX-M-MCUS.](https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/lpc-cortex-m-mcus:LPC-ARM-CORTEX-M-MCUS)