

Interfaz de manejo y visualización de los datos de un contador Geiger-Müller

Documentación

Facultad de Ingeniería, Universidad de Buenos Aires
2019

Autores:

Núñez Frau, Federico	98211	fedenuñez32@gmail.com
Vidal, Gabriel	97772	vidalgabriel93@gmail.com
Goyret, Juan Pablo	99081	juanpablogoyret@gmail.com

Índice

1. Descripción	2
2. Diagrama de bloques	3
3. Diagramas de flujo	4
4. Esquemático	5
4.1. Pines del microcontrolador/Arduino y su función	11
5. Configuración de la interfaz	11
5.1. Parámetros configurables	11
5.2. Configuración por default	11
6. Comunicación	12
6.1. Comunicación con PC / serie	12
6.2. Protocolo SCPI	12
6.2.1. Ejemplo 1	14
6.2.2. Ejemplo 2	14
6.2.3. Ejemplo 3	14
6.2.4. Códigos	14
7. Firmware	15
7.1. Máquinas de estado y eventos	15
7.2. Timers	15
7.3. Registros	16

1. Descripción

El dispositivo consiste en una interfaz de manejo y visualización de los datos de un contador Geiger-Müller, el cual es un circuito analógico generador de pulsos de tensión a partir de las partículas radioactivas que son recibidas en un tubo Geiger- Müller. Este último es un tubo de vidrio entre cuyos bornes se debe crear una tensión en torno a los cientos de volts y que, al recibir una partícula radioactiva, genera un pulso de corriente entre su ánodo y su cátodo, el cual es luego transformado a un pulso de tensión por un circuito secundario. Dado que las partículas arriban al tubo en instantes de tiempo aleatorios, los pulsos también se generan de este modo.

Estos pulsos de tensión son contados por un microcontrolador Atmega328p embebido en una placa Arduino, el cual realiza la tarea de determinar cuantos de estos han sido detectados en una unidad de tiempo dada, al igual que registrar los tiempos de llegada de cada pulso, para luego informarlo al usuario.

Como se observa en el diagrama de bloques de la Figura 1, los pulsos de tensión son comunicados al microcontrolador a través de opto-acopladores rápidos, para aislar eléctricamente al circuito analógico del digital (además, ambos se encuentran en placas separadas). Se cuenta además con una salida del microcontrolador opto-acoplada para encender la fuente de alta tensión que excita el tubo Geiger-Müller.

El microcontrolador informa la cantidad de pulsos de tensión recibidos por unidad de tiempo al usuario por medio de un panel LCD 2x16; así como también a una PC por medio de USB, empleando el sistema de comunicación UART y utilizando el protocolo SCPI (*Standard Commands for Programmable Instruments*)¹. Este último medio es también utilizado para enviar los tiempos de llegada de cada pulso, permitiéndole al usuario realizar, por ejemplo (pero no contemplados en este trabajo), análisis estadísticos de la radiación recibida. Esta conexión permite, además, dar comienzo o detener mediciones a través de la PC.

Además, el sistema cuenta con un teclado conformado por 6 pulsadores: flechas en las 4 direcciones, un botón de cancelar y uno de aceptar; con las siguientes funciones:

- Controlar el largo del intervalo de tiempo en el cual se cuentan los pulsos antes de informar un valor al usuario o "ventana de tiempo". Este valor se modifica mediante un menú de configuración visible en el display LCD.
- Dar comienzo o detener las mediciones.
- Configurar un umbral de pulsos recibidos por unidad de tiempo, el cual, de ser superado, implica el encendido de un LED y/o la activación de una alarma.

El microcontrolador utilizado se encuentra embebido en una placa Arduino UNO Rev3. Sobre este se acopla mediante pines largos una placa aparte (utilizando el formato *poncho* o *shield*), el cual contiene conectores para dos cables dirigidos a 2 placas extra. Una de ellas posee el display LCD, el teclado, el LED y el buzzer correspondiente a la alarma sonora. Por otro lado, la otra placa cuenta con la fuente de alta tensión encargada de alimentar al tubo del contador, y comunicar los pulsos al Atmega328.

El circuito analógico será alimentado mediante un transformador de 220V a 6V, contando el bloque digital con una alimentación independiente.

En las Figuras 2 y 3 se muestran los diagramas de flujo del menú de control y del sistema contador de pulsos respectivamente.

¹"SCPI 1999.0".Internet: <http://www.ivifoundation.org/docs/scpi-99.pdf> [Mayo, 1999]

2. Diagrama de bloques

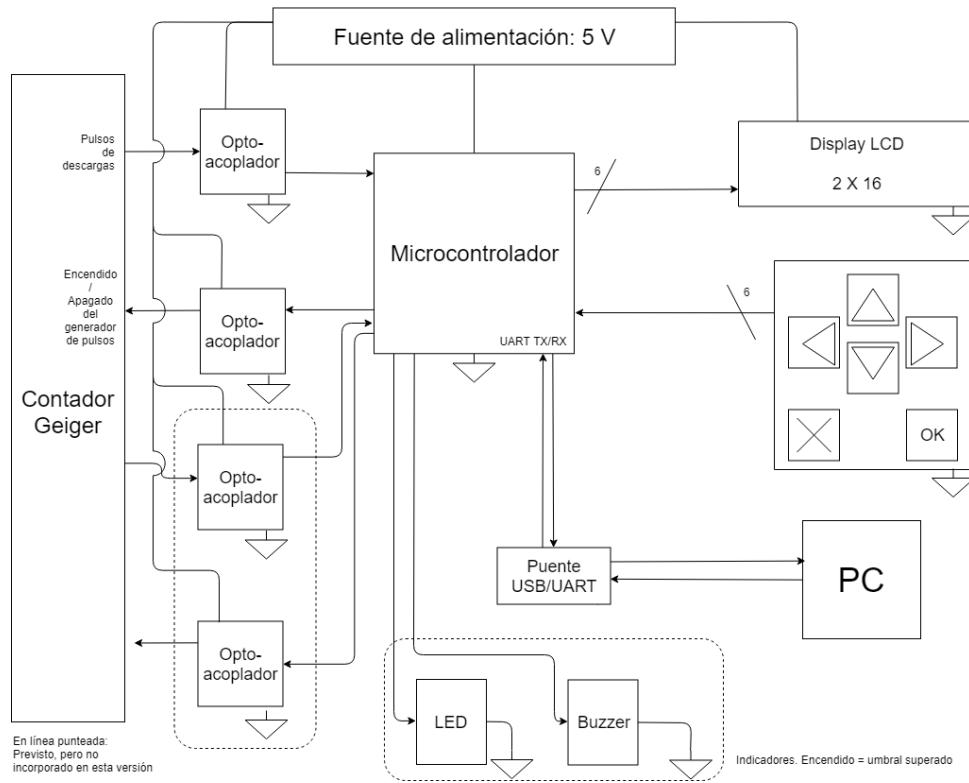


Figura 1: Diagrama de bloques.

3. Diagramas de flujo

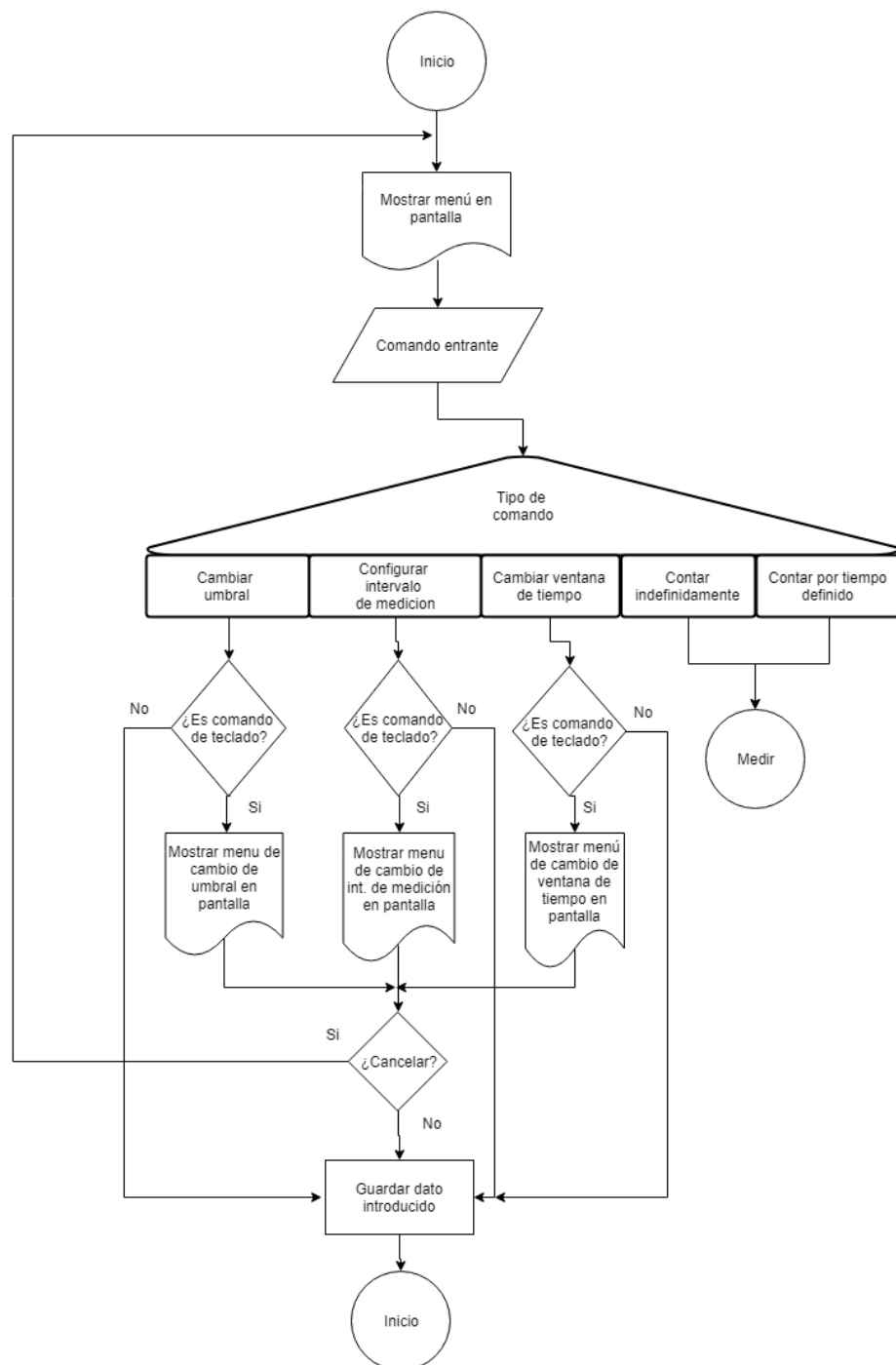


Figura 2: Diagrama de flujo del menú de control del contador.

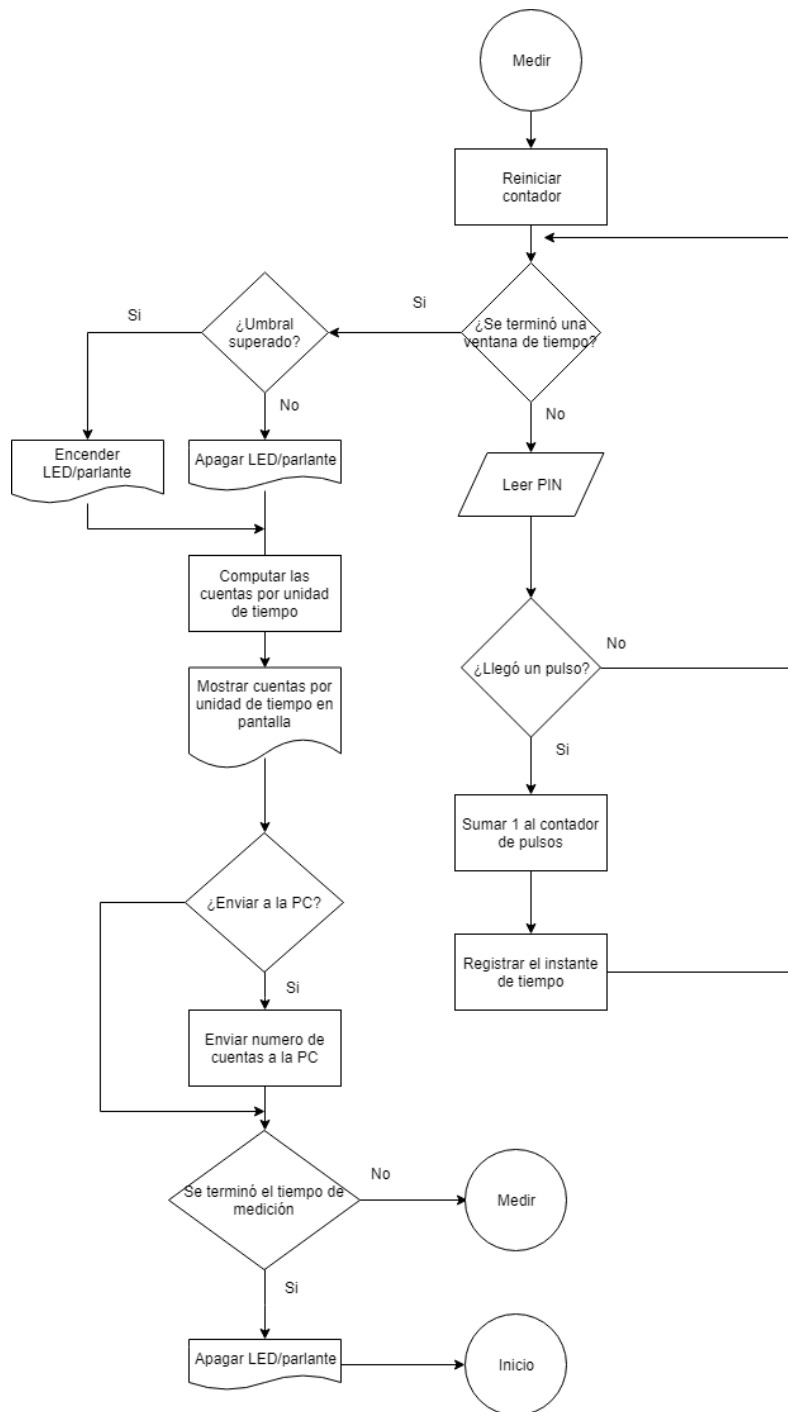
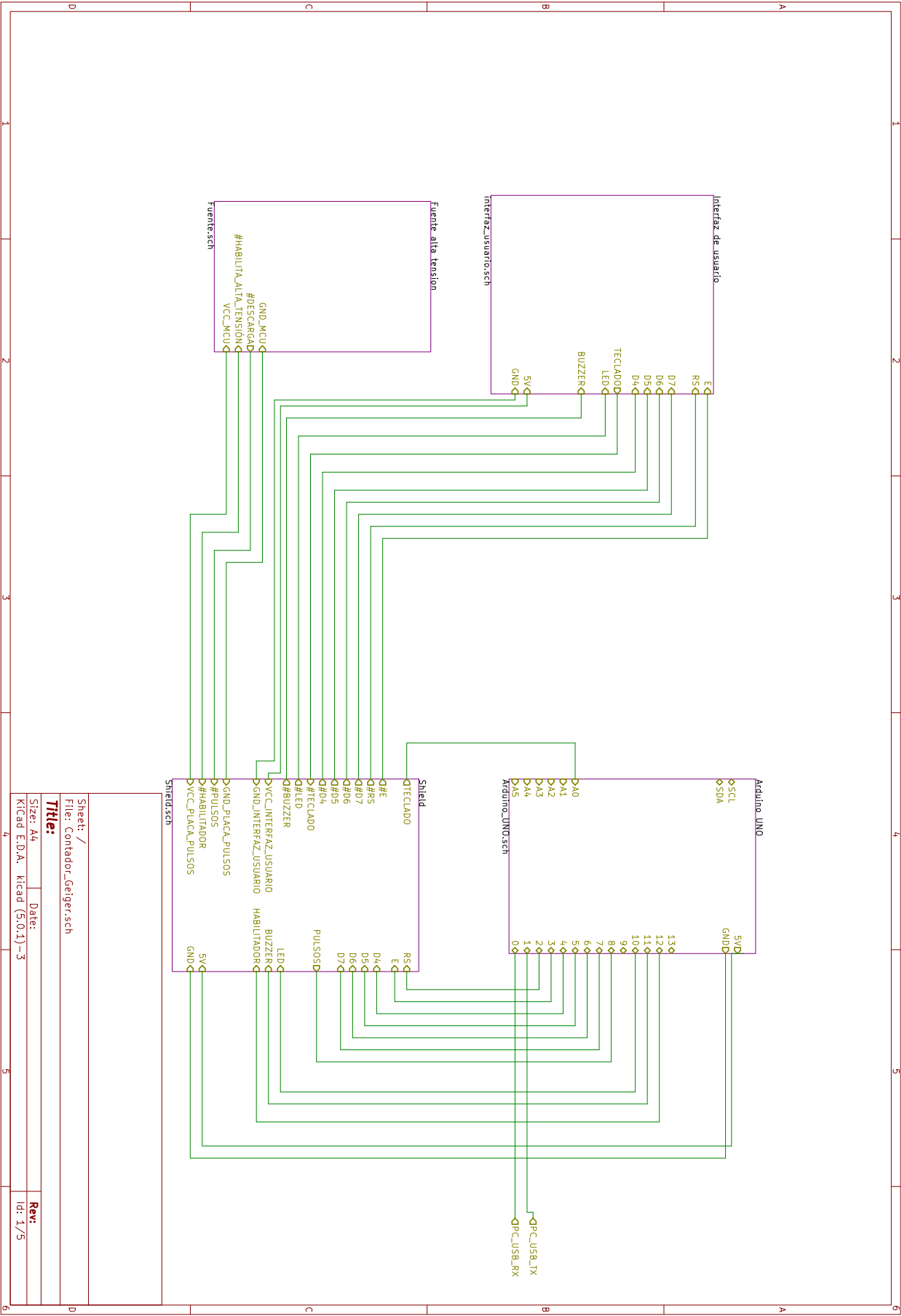


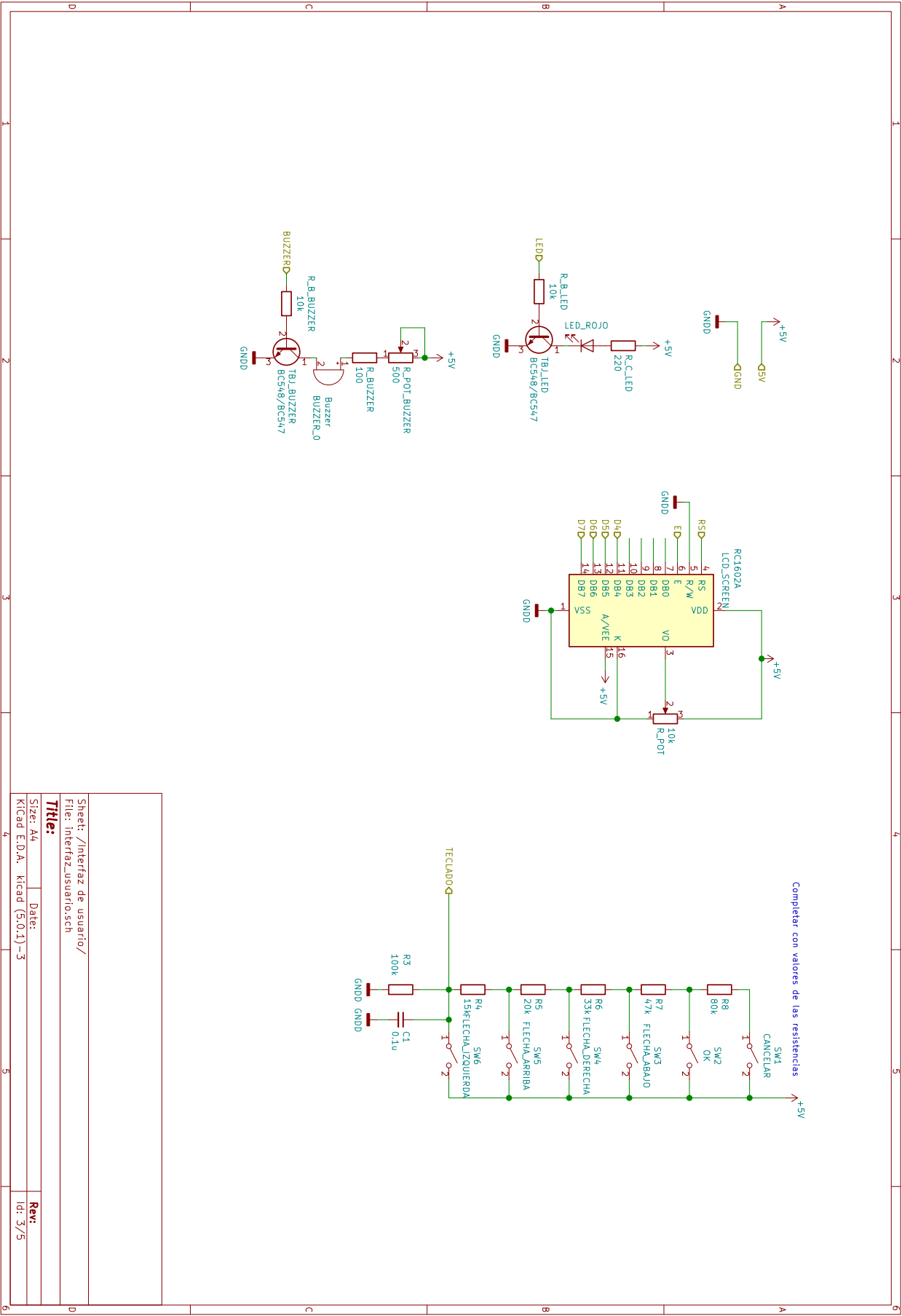
Figura 3: Diagrama de flujo del algoritmo de medición.

4. Esquemático

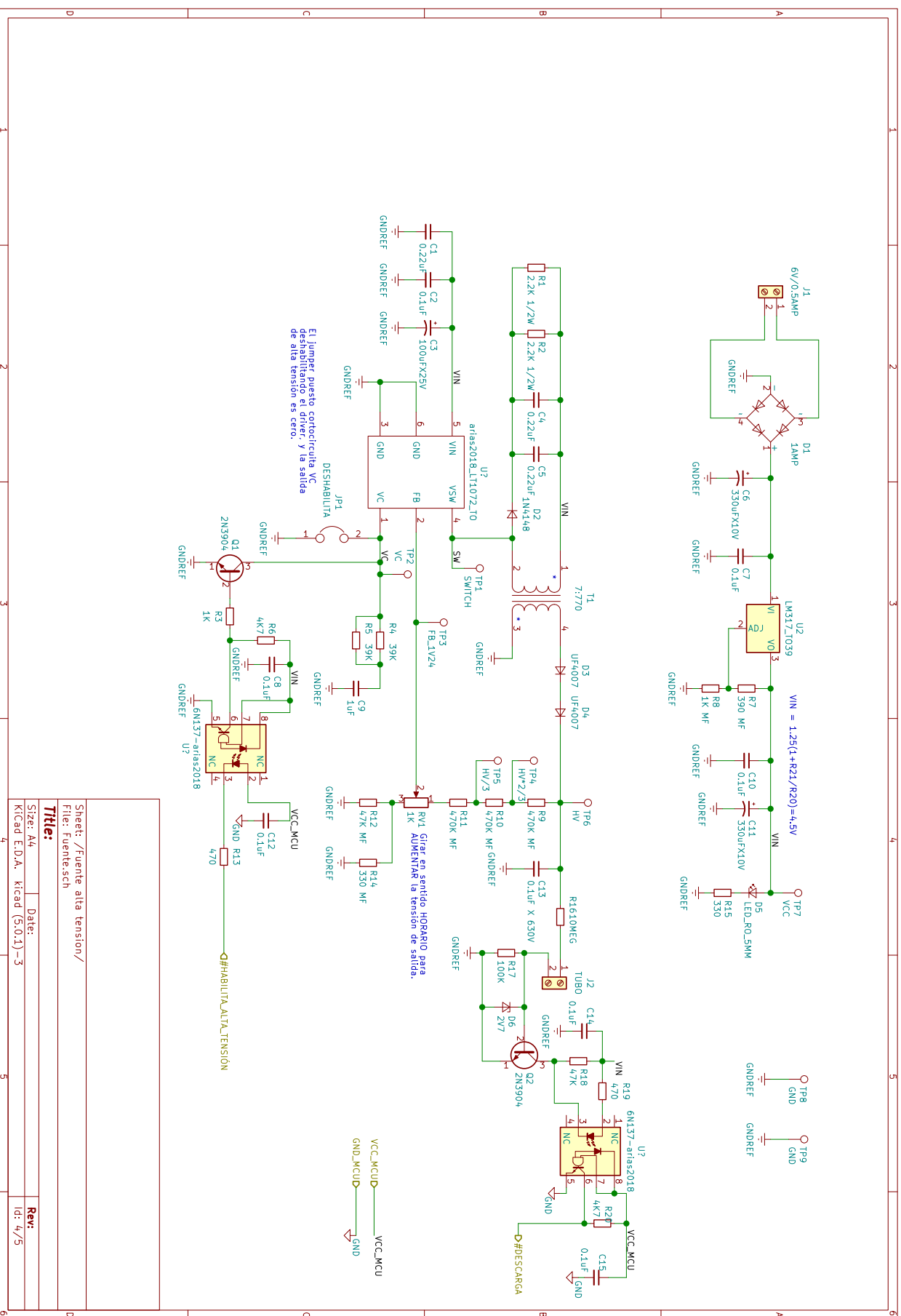
A continuación se incluye el esquemático del proyecto elaborado en *Kicad*. Este contempla el microcontrolador Atmega328p embebido en la placa Arduino, su *shield*, la placa del contador y la del panel LCD/teclado.

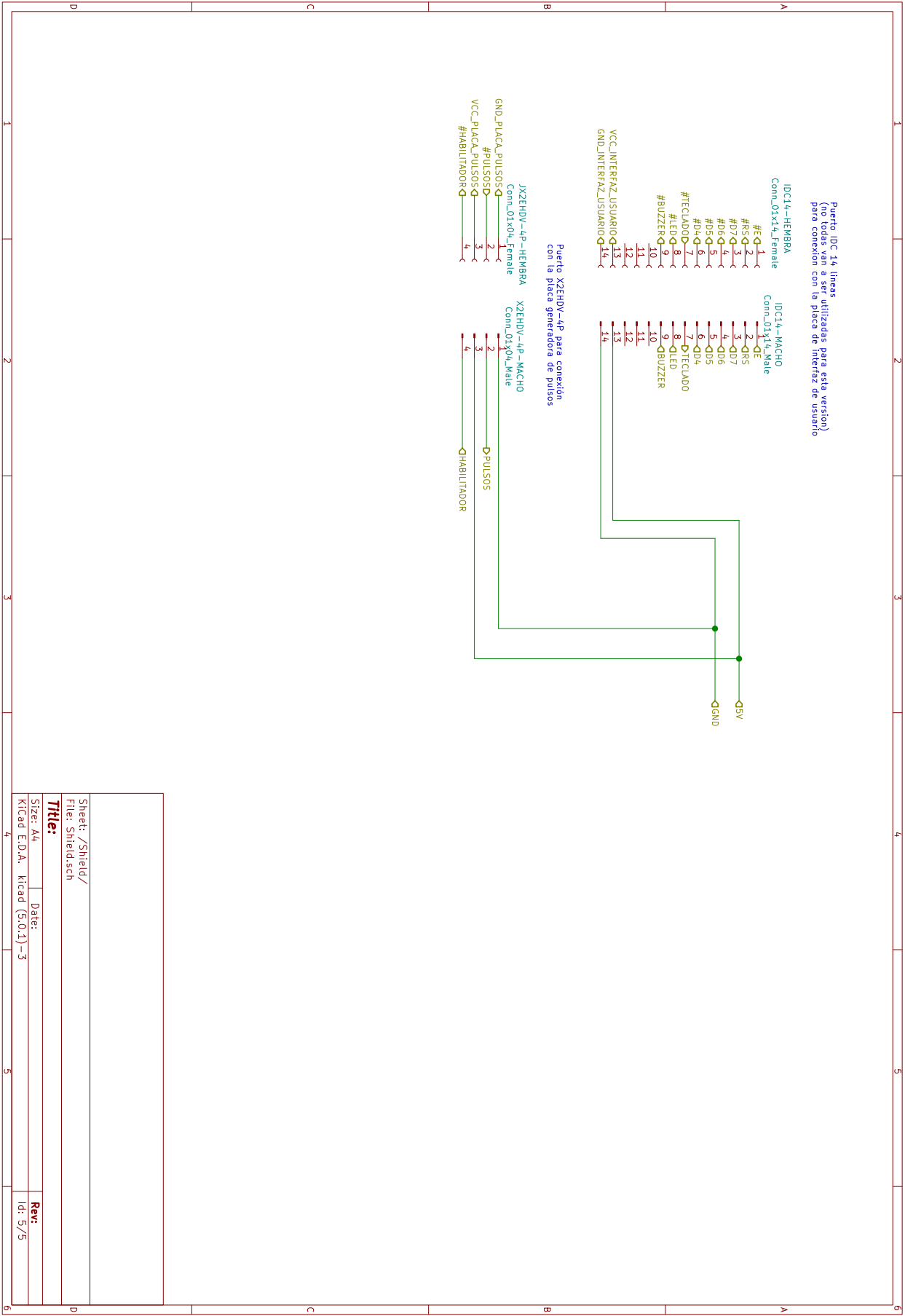


Sheet: /	
File: Contador_Geiger.sch	
Title:	
Size: A4	Date:
Kicad E.D.A. Kicad (5.0.1) - 3	
Rev:	
ID: 1/5	



Sheet: /Interfaz de usuario/ File: Interfaz_usuario.sch	
Title:	
Size: A4	Date:
KiCad E.D.A. KiCad (5.0.1) - 3	Rev:
4	5





4.1. Pines del microcontrolador/Arduino y su función

A continuación, en la Tabla 1 se especifica la funcionalidad de cada pin del microcontrolador Atmega328p para el contador Geiger, junto con su correlación con las salidas del Arduino UNO R3.

Pin Arduino UNO R3	Pin Atmega328p	Funcionalidad
Digital 0	PD0	Comunicación UART - RX
Digital 1	PD1	Comunicación UART - TX
Digital 2	PD2	Pantalla LCD: RS
Digital 3	PD3	Pantalla LCD: E
Digital 4	PD4	Pantalla LCD: datos
Digital 5	PD5	Pantalla LCD: datos
Digital 6	PD6	Pantalla LCD: datos
Digital 7	PD7	Pantalla LCD: datos
Digital 8	PB0/ICP1	Recepción de pulsos del tubo
Digital 12	PB4	Habilitación de la fuente de alta tensión
Analógico 0 (A0)	PC0	Teclados
Analógico 4 (A4)	PC4	Encendido de buzzer indicador
Analógico 5 (A5)	PC5	Encendido del LED indicador

Cuadro 1: Listado de los pines del Arduino UNO R3/Atmega328p con su respectiva funcionalidad para el contador Geiger.

5. Configuración de la interfaz

5.1. Parámetros configurables

La interfaz de control y gestión contempla la posibilidad de configurar los siguiente parámetros de medición:

- Cantidad de ventanas de tiempo a medir: entre 1 y 99999.
- Duración de las ventanas de tiempo a medir: 1ms, 10ms, 100ms o 1s.
- Umbral por encima del cual se activa una señal luminosa o sonora: entre 0 y 99999.
- Activación o desactivación de la señal sonora.
- Activación o desactivación de la fuente de alta tensión del tubo del contador Geiger-Müller.
- Activación del modo para retornar los tiempos de llegada de las partículas radioactivas al tubo.

A su vez, también es posible guardar la configuración del dispositivo en memoria EEPROM (mediante el comando *SAV por UART). De hacerselo, al resetearse este tomará automáticamente la configuración guardada.

5.2. Configuración por default

Los siguientes valores son los correspondientes a la configuración por default del dispositivo.

- Tiempo de la ventana: 1s.
- Número de ventanas: 10.
- Umbral de la señal luminosa/sonora: 1.
- Estado de la fuente: desactivada.
- Envío de tiempo de llegada de cada pulso: desactivado.

El dispositivo puede ser retornado a su configuración por default ejecutando el comando *RST.

6. Comunicación

6.1. Comunicación con PC / serie

La comunicación serie por USB con la PC se realiza empleando la USART del Atmega328p en modo UART. Para la conexión física se utiliza el hardware del propio Arduino como adaptador de serie a USB.

La comunicación por UART se encuentra configurada del siguiente modo:

- Baud rate: 76800 (se genera un 2 % de error).
- Tamaño de la palabra: 8 bits.
- Bit de paridad: no.
- Cantidad bits de stop: 1

Las cadenas de texto con comandos SCPI a enviar deben terminar con el caracter de retorno de carro `\n`. Por ejemplo, para comenzar una lectura enviar: `READ?\n`. De otro modo, el dispositivo nunca identificará que el mensaje enviado ha terminado y lanzará un error por timeout, devolviendo por consola el código -365.

Para comunicarse desde una PC puede utilizarse, por ejemplo, el programa PuTTY (al momento de realizar este trabajo, su última versión es la 0.7) disponible para Windows y Unix. Además de configurar el *baud rate*, el tamaño de palabra, el bit de paridad y el número de bits de *stop* como ya fue indicado, se debe configurar la pestaña "terminal" del modo en el que se muestra en la Figura 4

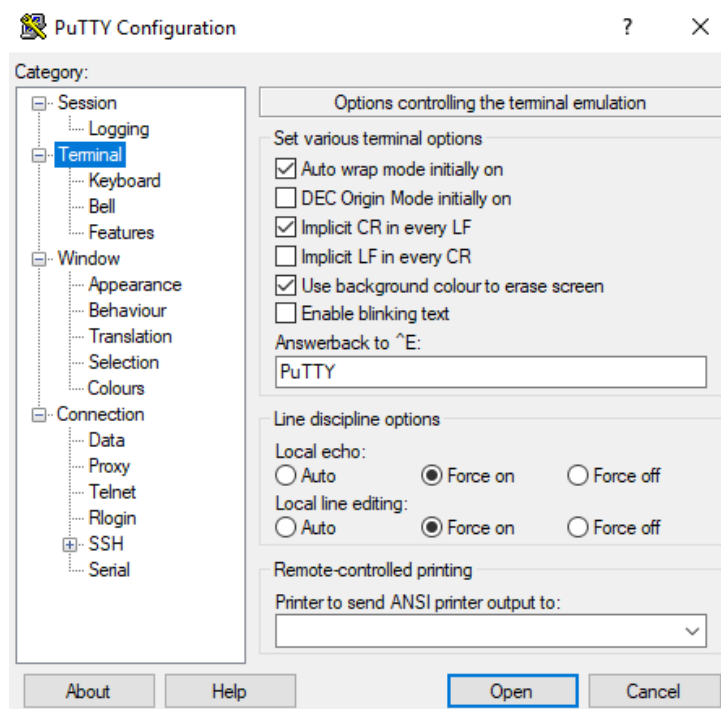


Figura 4: Configuración de la pestaña terminal de PuTTY.

6.2. Protocolo SCPI

Como ya se indicó, la comunicación serie se realiza utilizando el protocolo SCPI. Los comandos implementados son los que se muestran en el Cuadro 3.

Comando	Parámetro	Acción
*IDN?	N/C	Devolver un mensaje de identificación del dispositivo
*RST	N/C	Devolver la configuración del dispositivo a sus valores por default (ver 5.2)
READ?	N/C	Iniciar una medición utilizando la configuración actual del dispositivo
ABORt	N/C	Detener una medición
CONFigure?	N/C	Devolver la configuración de todo el dispositivo
CONFigure:WINDow <N>	Tamaño de la ventana. Valores posibles de N: 0 (1ms), 1 (10ms), 2 (100ms) y 3 (1s)	Configurar la duración de las ventanas de tiempo
CONFigure:WINDow?	N/C	Devolver el tamaño de la ventana de tiempo
CONFigure:COUNT <N>	Entre 0 y 99999	Configurar la cantidad de ventanas a medir
CONFigure:COUNT?	N/C	Devolver el número de ventanas a medir
CONFigure:TRIGger <N>	Entre 0 y 99999	Configurar el valor del umbral a partir del cual se enciende el LED o la alarma sonora (de encontrarse activada)
CONFigure:TRIGger?	N/C	Devolver el valor del umbral
CONFigure:DATA <BOOL>	0 (desactivado) o 1 (activado)	Configurar el retorno de los tiempos de los pulsos que arribar al contador
CONFigure:DATA?	N/C	Devolver si el el retorno de los tiempos de los pulsos está activado o no
CONTrol:APOWer <BOOL>	0 (desactivado) o 1 (activado)	Encender o apagar la fuente de alta tensión
CONTrol:APOWer?	N/C	Devolver si el tubo se encuentra recibiendo alta tensión o no
SYSTEM:BEEPer:STATe <BOOL>	0 (desactivado) o 1 (activado)	Encender o apagar la alarma sonora
SYSTEM:BEEPer:STATe?	N/C	Devolver si la alarma sonora se encuentra activada
*SAV	N/C	Guardar la configuración del dispositivo en memoria EEPROM para retomarla cuando este se reinicie

Cuadro 2: Comandos SCPI implementados para comunicación serie. <N>es un parámetro representado por un número entero, mientras que <BOOL>uno representado por un booleano.

Varios comandos mostrados en el Cuadro 3 tienen una forma extendida y una forma reducida. La forma extendida comprende todos los caracteres del comando tal y como se lo muestra en la columna izquierda de la tabla. Por otro lado, la forma reducida se ve representada solamente por la **letras en mayúscula y los símbolos como ":z l"**. Tanto la forma reducida como la extendida son válidas.

Por ejemplo, si se tiene el comando `CONF:Figure:DATA?"`, su forma reducida sería `CONF:DATA?"`. Del mismo modo, con `CONF:Figure:WINDow <N>`, su forma reducida es `CONF:WIND <N>`. En ambos casos ejemplo se obtendrá la misma respuesta tanto para el comando extendido como para el reducido.

A continuación se muestran algunos ejemplos. Recordar que a todos los comandos, tal y como son presentados en dichos ejemplos, se les debe agregar el caracter de retorno de carro (`\n`) de modos que sean correctamente interpretados por la interfaz.

6.2.1. Ejemplo 1

Suponiendo que se quisiera encender la alimentación de alta tensión del tubo del contador Geiger, se podría ejecutar el siguiente comando:

`CONTrol:APOWer 1`

O, en su forma, reducida:

`CONT:APOW 1`

Y también se podría enviar, por ejemplo:

`CONT:APOWer 1`

o bien,

`CONTrol:APOW 1`

En todos los casos, la fuente de alta tensión será encendida.

6.2.2. Ejemplo 2

Si se quisiera configurar la ventana de tiempo en 10ms y que el dispositivo mida 100 ventanas se podrían enviar los siguientes comandos:

`CONF:WIND 2`
`CONF:COUN 100`

6.2.3. Ejemplo 3

Si se quisiera saber el número de pulsos recibidos en la ventana de tiempo configurada tal que la señal luminosa o sonora se encienda (umbral superado), se podría ejecutar el siguiente comando:

`CONF:TRIG?`

Y el dispositivo devolverá el valor correspondiente

6.2.4. Códigos

La implementación del protocolo SCPI en la interfaz de gestión y control contempla algunos códigos de error o de información, que son los que se presentan en el Cuadro ...

Código	Significado
-100	El comando recibido es erróneo
-365	Timeout en la recepción de un comando. Se ha tardado demasiado tiempo en recibir un comando
-363	Comando demasiado extenso (>60 caracteres). Overflow en el buffer de recepción de comandos
-284	Dispositivo ocupado. No se puede recibir un comando en este momento

Cuadro 3: Comandos SCPI implementados para comunicación serie. <N>es un parámetro representado por un número entero, mientras que <BOOL>uno representado por un booleano.

7. Firmware

7.1. Máquinas de estado y eventos

El firmware consiste principalmente de 3 máquinas de estado, cada una asociada a funcionalidades distintas de la interfaz de control y gestión. Estas son:

- Realización de las mediciones.
- Recepción y envío de datos por UART.
- Menú mostrado en la pantalla LCD del dispositivo.

Los estados de las primeras dos máquinas se encuentran almacenados en el registro ESTADO (R24), mientras que el de la tercera lo hace en ESTADOS_LCD (R22). Ambos son registros reservados que no pueden ser accedidos por ningún otro proceso.

Las máquinas cambian de estado o ejecutan acciones dependiendo de los eventos que ocurran en el microcontrolador, como la llegada de un carácter por UART o un pulso por el pin ICP1. Dichos estados se almacenan en el registro reservado EVENTO (R25), y en el registro en RAM EVENTO2. Este último fue creado debido a la falta de registros de acceso común disponibles para ser reservados a una tarea en específico.

7.2. Timers

La interfaz de control y gestión utiliza los tres timers disponibles en el microcontrolador Atmega328P. Sus funciones son:

- Timer 0:
 - Identificar un timeout (o exceso de tiempo) en la recepción de una cadena de caracteres por UART. Se genera una interrupción si ha transcurrido un cierto tiempo desde la llegada de un primer carácter sin que se haya recibido un carácter de fin de trama.
 - Debouncer para detectar la pulsación de una tecla en el teclado mediante el comparador y el ADC.
- Timer 1: guardar los instantes de tiempo de los pulsos capturados.
- Timer 2: generar los delays del LCD.

7.3. Registros

En el Cuadro 4 pueden verse los registros de uso común R0 a R31 catalogados según su uso. Si se encuentran reservados para alguna tarea en específico (no pueden ser utilizados en ninguna otra función que no sean aquellas relacionadas con el proceso para el cual se encuentran justamente reservados), entonces se aclara su motivo.

Registro	Reservado	Uso	Definición
R0	Si	Multiplicaciones	N/C
R1	Si	Multiplicaciones	N/C
R2	Si	Contador del timer 0	CONT_TIMER0_LOW
R3	Si	Contador del timer 0	CONT_TIMER0_HIGH
R4	No	Uso general	N/C
R5	No	Uso general	N/C
R6	No	Uso general	N/C
R7	No	Uso general	N/C
R8	Si	Puntero al LSB del buffer de transmisión de UART	PTR_TX_L
R9	Si	Puntero al MSB del buffer de transmisión de UART	PTR_TX_H
R10	Si	Puntero al LSB del buffer de recepción de UART	PTR_RX_L
R11	Si	Puntero al MSB del buffer de recepción de UART	PTR_RX_H
R12	Si	Cantidad de caracteres almacenados en el buffer de transmisión de datos de UART	BYTES_TRANSMITIR
R13	Si	Cantidad de caracteres almacenados en el buffer de recepción de datos de UART	BYTES_RECIBIDOS
R14	No	Uso general	N/C
R15	Si	Multiplicador para extender el tamaño de la ventana de tiempo	MUL_DE_VENTANA
R16	No	Uso general	N/C
R17	No	Uso general	N/C
R18	No	Uso general	N/C
R19	No	Uso general	N/C
R20	No	Uso general	N/C
R21	No	Uso general	N/C
R22	Si	Estados de la máquina de estados del LCD	ESTADOS_LCD
R23	Si	Lleva la cuenta de la cantidad de pulsos medidos	CONTADOR_DE_PULSOS
R24	Si	Almacenamiento de los estados de las máquinas de estado de la comunicación serie y del sistema de recepción de los pulsos	ESTADO
R25		Almacenamiento de eventos que alteran los estados de las máquinas de estado	EVENTO
R26	Si	Puntero X	N/C
R27	Si	Puntero X	N/C
R28	Si	Puntero Y	N/C
R29	Si	Puntero Y	N/C
R30	Si	Puntero Z	N/C
R31	Si	Puntero Z	N/C

Cuadro 4: Comandos SCPI implementados para comunicación serie. <N>es un parámetro representado por un número entero, mientras que <BOOL>uno representado por un booleano. En la última columna se aclara la definición del registro en el código si es que se lo ha redeclarado con ".def".