Capítulo 6

Arquitectura de Software

6.1 Descripción del Firmware

La programación en firme (del ingles firmware) desarrollado fue pensando para facilitar el uso del sistema y minimizar los errores que suele introducir el usuario dentro de un sistema, elimina por software las inestabilidades del equipo básico y evita el uso de metodologías de caracterización no adecuadas que algunos usuarios por su formación primera conservan como validas.

La arquitectura principal del software de aplicación, es un menú de introducción de parámetros el cual inicia con la especificación del voltaje requerido y continua con la recepción de los parámetros de tratamiento deseados (ancho de pulso, tren de pulsos y número de pulsos), al finalizar la entrada de parámetros, el software pide la autorización al usuario para la ejecución del tratamiento, permitiendo de este modo que un tratamiento pueda ser abortado por el usuario en cualquier momento, regresando a la etapa de introducción de parámetros a la espera de nuevas aportaciones de tratamiento.

Por otro lado, una vez que los parámetros del tratamiento son aceptados, el usuario debe autorizar el tratamiento para que el sistema elabore todos los datos recibidos, despliegue los parámetros del tratamiento realizados y por último regrese a la etapa de introducción de nuevos parámetros.

De igual forma si se produce un error durante una operación ó los parámetros son incompatibles inmediatamente el software notifica al usuario del error desplegándolo en el display y generando una señal auditiva que le avisa del error.

Como se mencionó en el capítulo 5, la programación del firmware se hace en el entorno de programación de desarrollo MPLAB de microchip usando el compilador C18.

El código fuente se dividió en los siguientes módulos de acuerdo a su función:

- ✓ electroporador.c : Contiene el programa principal en donde se configuran los parámetros de control de electroporación, registro de eventos y datos y control global de errores.
- ✓ electroporador.h : Consta de todas las declaraciones de prototipos de función, declaración de los fuses del microcontrolador, declaración y asignación de variables a memoria del microcontrolador e inicializan todas las variables internas del programa.
- ✓ LCD_PIC.c : Modulo que especifica todas las funciones relacionadas con la escritura y borrado básico de caracteres a la pantalla LCD, además de establecer su inicialización.
- ✓ LCD_PIC.h: Modulo de cabecera que contiene todas las declaraciones y prototipos de función que manejan al LCD.

- ✓ teclado.c: Modulo que permite la introducción de parámetros a través de un teclado matricial.
- ✓ funciones.c: Modulo que especifica todas las funciones necesarias para el funcionamiento del programa principal "main" además de subrutinas de decisiones de control.
- ✓ boolean.h: función que define valores que pueden ser solo verdadero ó falso correspondiente a los dos estados lógicos de una variable digital (1 y 0).

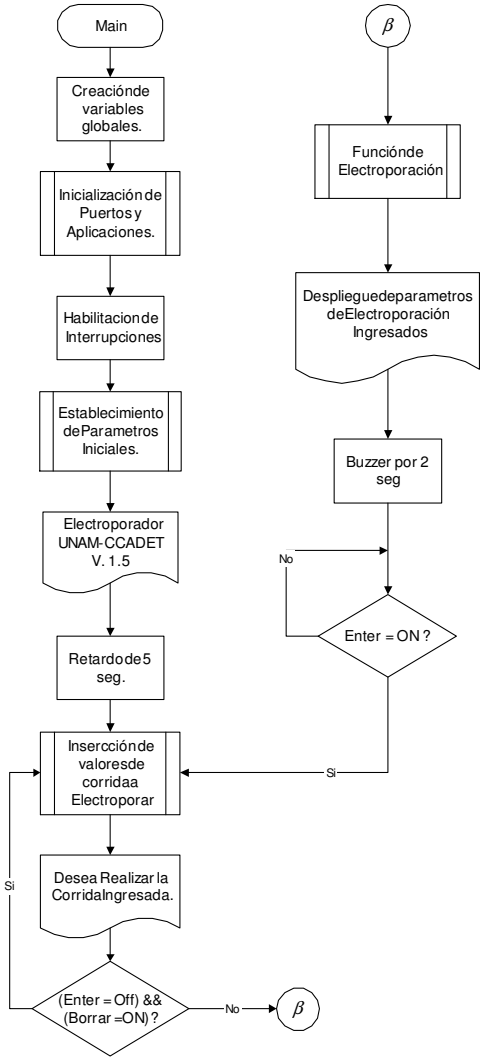
La programación en la tarjeta de control se efectúa con el programador PICKIT 2.

A continuación se muestra los diagramas de flujo que describen la arquitectura del firmware del microcontrolador.

6.2 Diagrama de bloques del Firmware

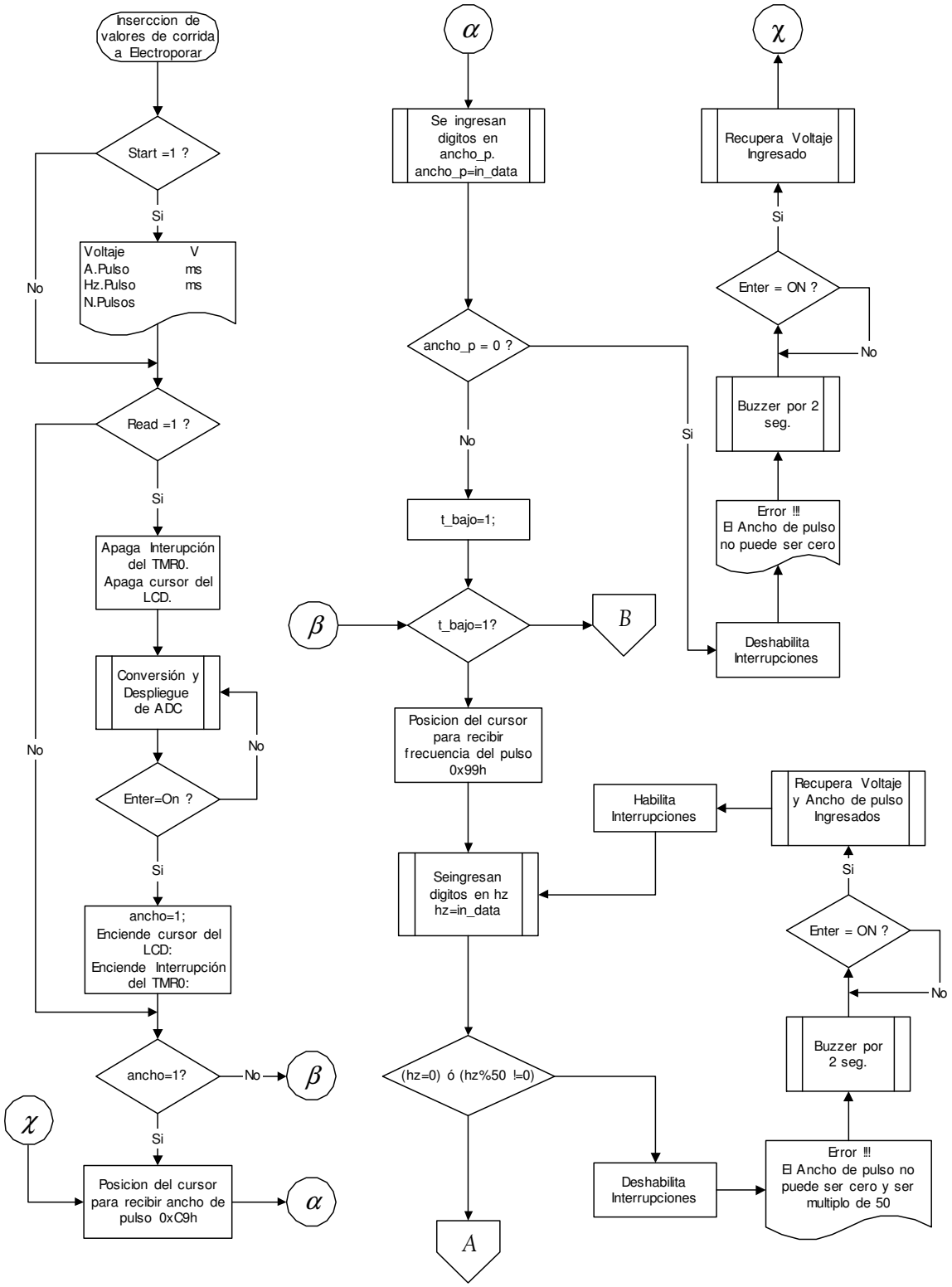
6.2.1. Función main

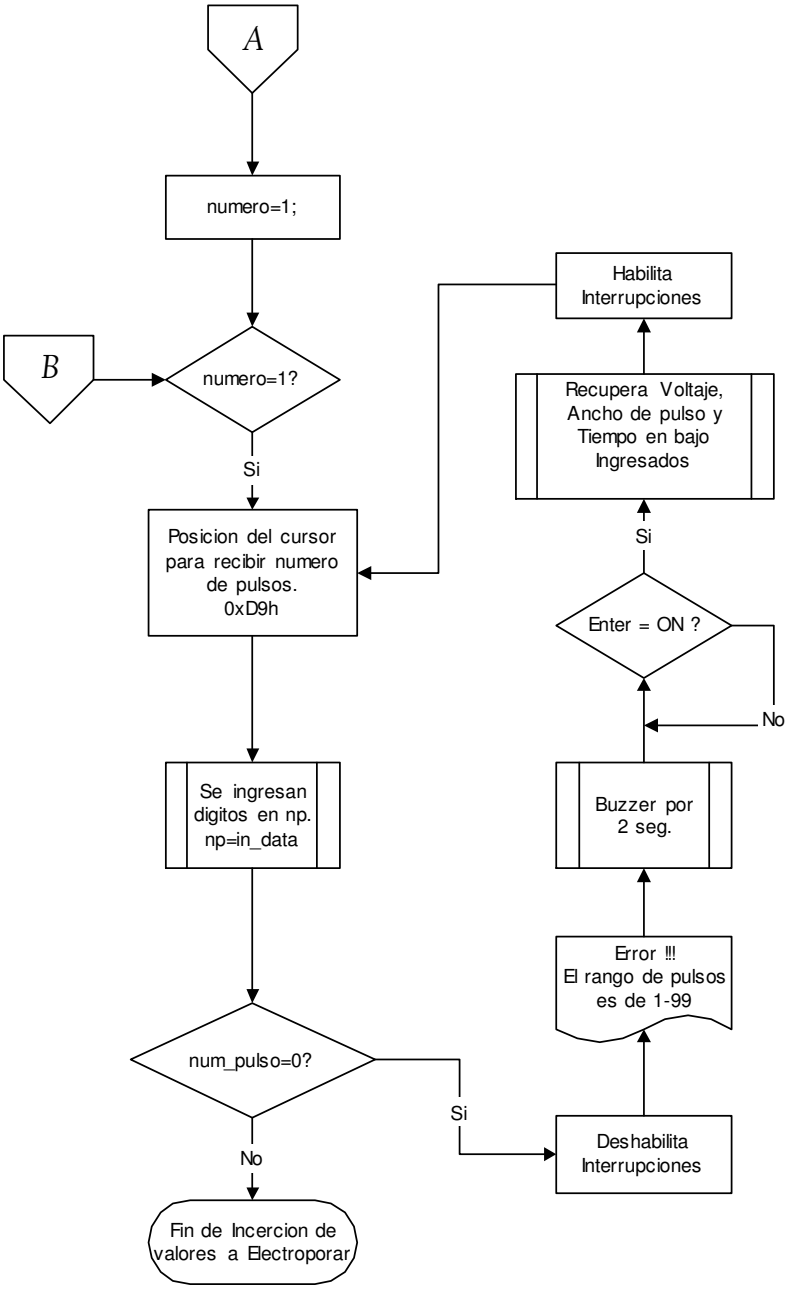
Función principal que especifica el comportamiento del sistema de electroporación.



6.2.2. Función de Inserción de Parámetros

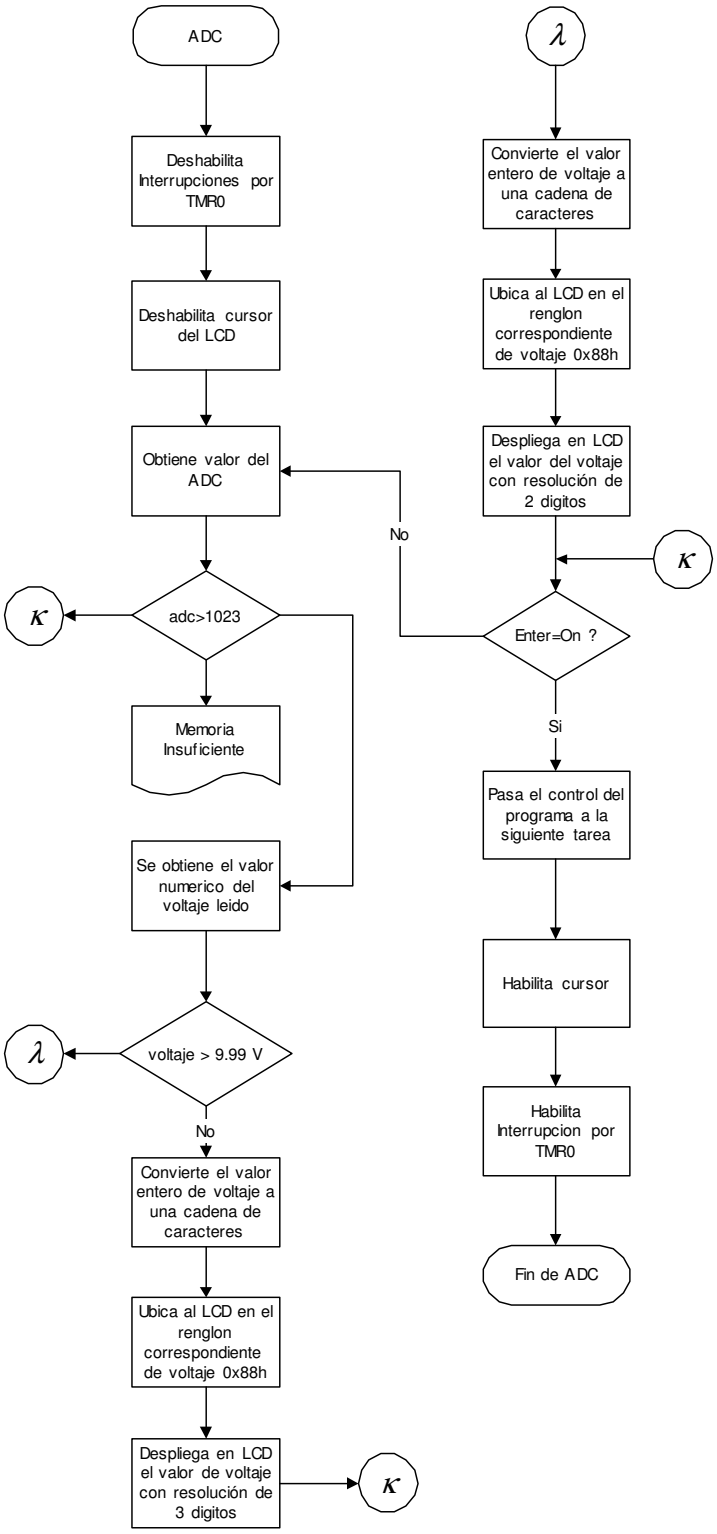
Esta subrutina se encarga de ingresar en distintas variables los valores del tratamiento ingresado para su posterior manejo y control dentro del proceso de elctroporación.




A
numero=1;
B
numero=1?
Si
Posicion del cursor para recibir numero de pulsos. 0xD9h
Se ingresan digitos en np. np=in_data
num_pulso=0?
Si
No
Fin de Incercion de valores a Electroporar
Deshabilita Interrupciones
Error !!! El rango de pulsos es de 1-99
Buzzer por 2 seg.
Enter = ON ?
No
Si
Recupera Voltaje, Ancho de pulso y Tiempo en bajo Ingresados
Habilita Interrupciones

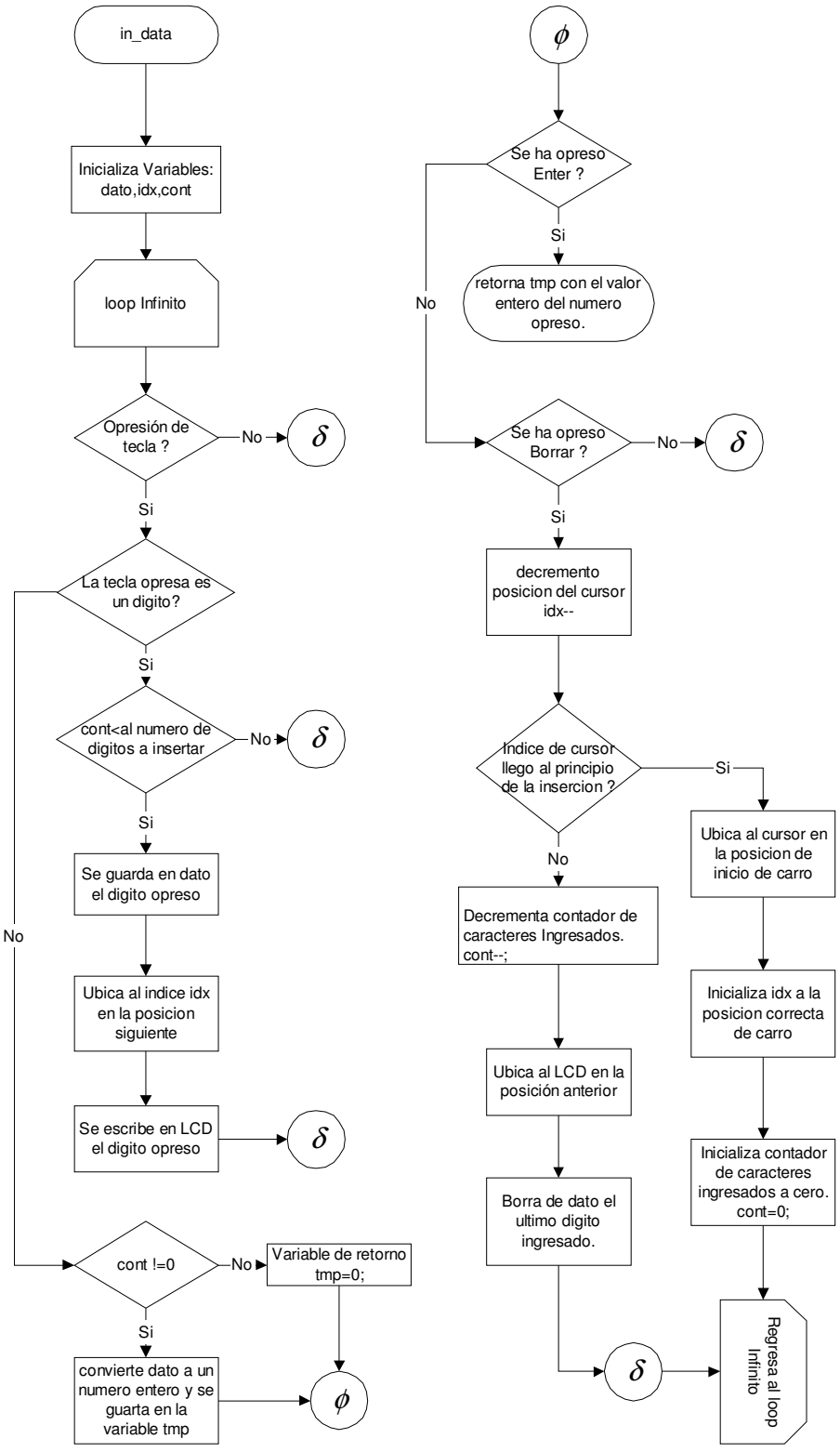
6.2.3. Función ADC

Esta función se encarga de censar el voltaje de la UE y mostrarlo en pantalla.



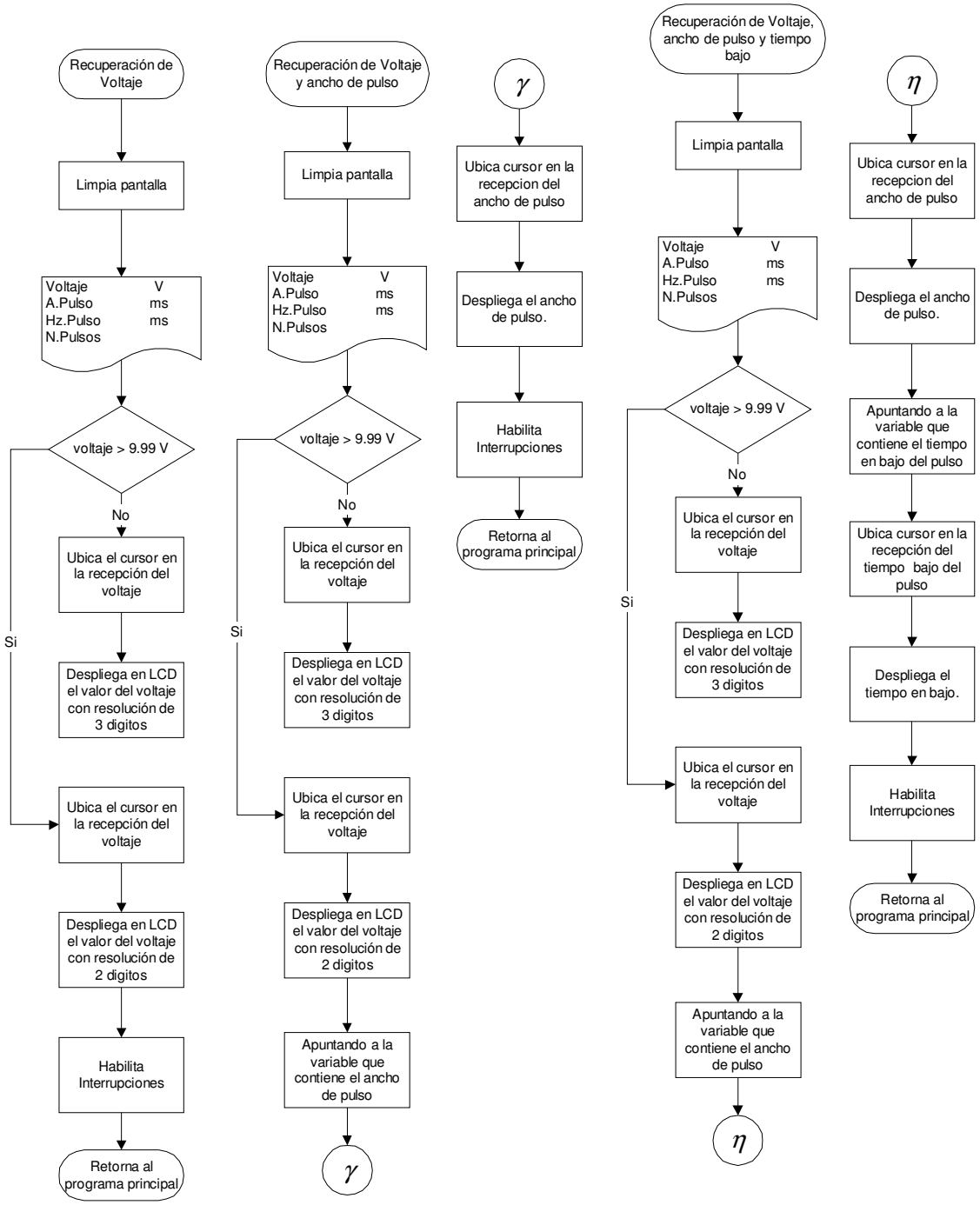
6.2.4. Función in_data

Esta función recolecta los dígitos que son tecleados, los despliega en LCD y guarda su valor como ASCII en una variable temporal.

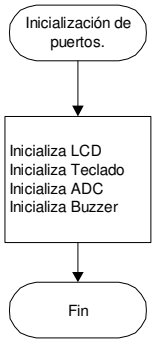


6.2.5. Funciones de Recuperación

Cada vez que se genera un error el sistema avisa del mismo, desplegando en pantalla dicho error, las funciones de recuperación despliegan en pantalla los valores ya ingresados por el usuario antes que se introdujera un error en el sistema.



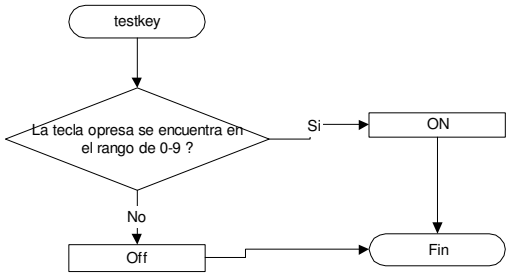
6.2.6. Función Inicialización de puertos



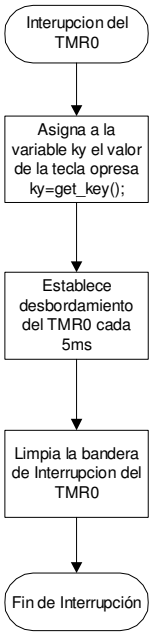
Esta función inicializa puertos, registros y funciones principales para el funcionamientos del sistema.

6.2.7. Función testkey

Esta función regresa un valor booleano dependiendo de la tecla opresa si es un numero retorna un 1 de lo contrario retorna un Ø

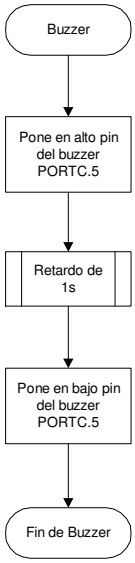


6.2.8. Función interrupción del TMR0



Esta función realiza interrupciones por hardware a través del TMR0 cada 5 ms para la lectura del teclado matricial.

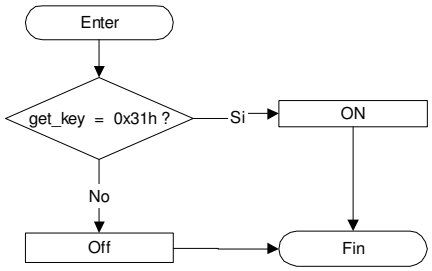
6.2.9. Función Buzzer



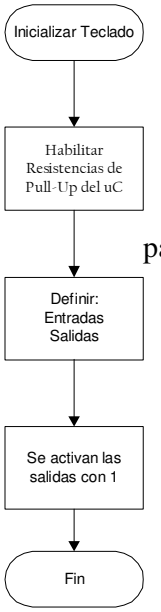
Esta función genera una señal auditiva que avisara al usuario de algún error y finalización de tratamiento.

6.2.10. Función Enter

Esta función regresa un valor booleano si la tecla Enter ha sido opresa.



6.2.11. Funcion Inicializacin del Teclado



Esta función inicializa el puerto B, así como los resistores de jalón del puerto para la utilización del teclado matricial.

6.2.12. Función get_key

Esta función escanea el teclado y regresar el valor de la tecla pulsada a una variable para su posterior manejo