Universidad Nacional de Córdoba

Facultad de Matemática, Astronomía, Física y Computación
Lic. en Ciencias de la Computación

Control de plataforma de degradación acelerada de transistores de potencia

Autor: Brian Ezequiel Marchi
Director: Gonzalo Tomás Vodanovic

Córdoba, Argentina
Julio 2020

Agradecimientos

A Gonzalo Vodanovic, mi director, por haberme guíado, acompañado y ayudado en este arduo trabajo.

A Delfina Velez y Agustín Laprovitta, que también ayudaron en los experimentos realizados e hicieron la experiencia más agradable y llevadera.

Al jurado, Walter Zaninetti y Eduardo Romero, por tomarse el tiempo de leer el informe y evaluarme.

A mi madre, Andrea Osenda, por siempre apoyarme en las decisiones que he tomado a lo largo de la vida y por seguir haciéndolo.

A Alejandra Acosta, por haberme apoyado, ayudado y acompañado a lo largo de la carrera (y seguir haciéndolo). Sin ella no hubiese llegado a esta instancia.

A Rodrigo Fonseca, amigo, hermano y socio de toda la vida, que si bien estudia en otra facultad, los sentimientos sobre la institución son compartidos.

A Santiago Ruiz, Andrés Moro, Juan Martinez, Andrés Mateo, Joaquín Zarate, Tomás Romero y Martín Luna, por la amistad brindada por tantos años (algunos más años, otros menos), por tantas horas de vicio, risas, etc.

A Gustavo Giunta, Sofía Ruiz, Viviana Altenburger y Fernando Ruiz, mi segunda familia. Siempre estuvieron presentes y me brindaron consejos, ayuda, risas y un sin fin de cosas.

A Javier Lezama, mi profesor de la parte práctica de Álgebra y Matemática Discreta I, que con su ayuda pude arrancar bien el año y tomar riendas de la carrera.

A Nicolás Wolovick, Carlos Bederián y Daniel Penazzi por ser excelentes docentes y poner mucho esmero y ganas por enseñar.

Cada persona, las nombradas y las que no, contribuyeron de alguna forma a este logro, por lo que el título es de todos, no solo mio.

Resumen

El objetivo del presente trabajo es desarrollar una plataforma para acelerar, automatizar y emular el proceso de degradación al que se somete un transistor de efecto de campo metal-óxido-semiconductor (MOSFET) de potencia en un aparato de Resonancia Magnética Nuclear por ciclado rápido de campo (RMN-FFC). Este sistema debe ser capaz de controlar la degradación del transistor; analizar las condiciones de destrucción y reaccionar a ellas; observar los parámetros de funcionamiento durante los ensayos mediante gráficos y almacenar toda la información en una base de datos local.

Palabras clave: MOSFET, C++, Qt, Degradación, SOA, Regresor.

Abstract

The objective of the present work is to develop a platform to accelerate, automate and emulate the degradation process to which a power metal-oxide-semiconductor field effect transistor (MOSFET) is subjected in a fast field-cycling Nuclear Magnetic Resonance equipment (FFC-NMR). The system should be able to control degradation of the transistor; analyze destruction conditions and react to them; observe the operating parameters during the trials using graphics and saving all information in a local database.

Keywords: MOSFET, C++, Qt, Degradation, SOA, Regressor.

Índice general

Introducción

Los aparatos de Resonancia Magnética Nuclear de Ciclado Rápido de Campo (*FFC*, *Fast Field Cycling*) utilizan transistores para controlar el campo magnético que se genera en un electroimán. Estos dispositivos se encuentran sometidos a un severo estrés eléctrico, térmico y mecánico que disminuye progresivamente la vida útil de los mismos. En condiciones normales de funcionamiento la degradación de los transistores podría tomar meses en manifestarse, dificultando su análisis. En el presente trabajo se pretende automatizar una plataforma para degradar de forma acelerada transistores de potencia, emulando distintas condiciones de trabajo a las que son sometidos en un aparato de RMN-FFC.

El sistema realizado puede observarse en la figura 1, el cual está compuesto por un software que corre en una PC, un microcontrolador con su respectivo software, una etapa de adaptación de señales y una etapa de potencia donde se instala el transistor a degradar.

El software de la PC cuenta con una interfaz gráfica mediante la cual se seleccionan todos los parámetros configurables de degradación. Estos son enviados al microcontrolador, que se encarga del control de la degradación. Dado que las señales que el microcontrolador genera no son compatibles con el circuito de potencia, se incorpora una etapa de adaptación de señales que realiza la interfaz. Particularmente, en este trabajo se utilizara un transistor del tipo MOSFET, sin embargo, la plataforma podría utilizarse para cualquier transistor tipo FET. La amplitud de la tensión de *gate* del transistor a degradar se determina a partir de los parámetros configurados en el software de la PC. Este valor se comunica al microcontrolador y este genera la señal correspondiente que luego es acondicionada en la etapa de adaptación de señales para finalmente aplicarse al *gate* del transistor.

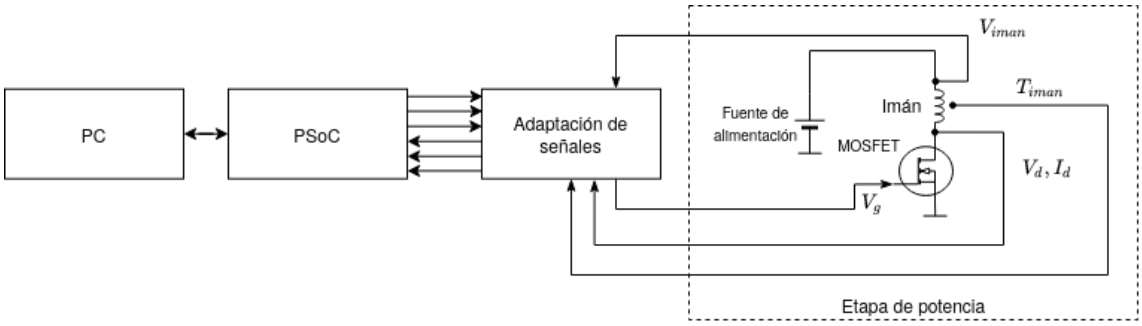


Figura 1: Plataforma de degradación de transistores

Por otro lado, las distintas mediciones (temperatura del imán T_{iman}, tensión de *drain* V_D, corriente de *drain* I_D y tensión del imán V_{iman}) se realizan mediante distintos métodos. Estas señales se ajustan a los valores permitidos del microcontrolador en la etapa de adaptación y son leídas por el microcontrolador para finalmente ser comunicadas a la PC para que puedan tomarse las decisiones correspondientes.

Para acelerar el proceso de degradación del MOSFET de potencia, preservando los demás componentes del aparato, se emulan las condiciones a las que se somete el transistor, en las circunstancias más desfavorables respecto de sus especificaciones de funcionamiento, y se controla

que no llegue al extremo de romperse. Sin embargo, los límites de corriente y temperatura de juntura pueden variar rápidamente ante ciertas circunstancias, por lo que puede darse el caso en que el dispositivo falle antes que el microcontrolador analice los parámetros y ejecute la acción correspondiente. Para resolver esto, se emplea un método simplificado de pronóstico a corto plazo, mediante regresión local, para pronosticar si el siguiente ciclo de degradación llevará al transistor a trabajar fuera de sus límites de operación segura y detener el proceso antes de que esto ocurra.

La automatización de este proceso permite que las degradaciones de los MOSFETs sean **replicables** y **precisas**. Esto ayuda a la comprensión de las causas de rompimiento o degradación y se pueden tomar acciones correctivas para evitarlo. También es flexible, ya que, el software ofrece diversos parámetros de degradación modificables por el usuario, como el tipo de degradación, frecuencia de pulsos de tensión sobre el *gate*, tensión aplicada al *gate*, la inductancia del imán, el período de trabajo de los pulsos, etc.

1. MOSFET de Potencia

Los MOSFETs son transistores que pertenecen a la familia de los **FET** (Field Effect Transistor). El control de corriente en estos dispositivos depende de un campo eléctrico que se genera a partir de la creación de un canal entre el *drain* y el *source* al aplicarse una tensión en el *gate* para controlar la conductividad de este. Éste puede funcionar como un amplificador o conmutador, que puede variar dependiendo de qué tipo de configuración se utilice.

Estos son utilizados enormemente en la industria donde este involucrado cualquier tipo de circuito electrónico, por lo que comprender las causas de deterioro y/o destrucción ayuda a realizar los circuitos para prevenirlas o, en caso de no poder prevenirlo, determinar el tiempo en el que será necesario cambiar el recurso físico antes que se destruya por completo.

Este capítulo trata sobre los transistores en general para luego centrarse en los MOSFETs de potencia [1] y las características que nos importan de éste para el experimento.

1.1 Qué es un transistor

Como se dijo anteriormente, un transistor puede funcionar como amplificador y/o conmutador. Si bien hay varios tipos de transistores, lo más utilizados son los **FETs**, mencionados anteriormente y los **BJT** (Bipolar Junction Transistor) [2]. Los transitores del tipo BJT, cuentan con tres pines llamados base, collector y emitter (ver figura 1.1), mientras que los tipo *FET* cuentan con tres pines llamados *gate*, *drain* y *source*.

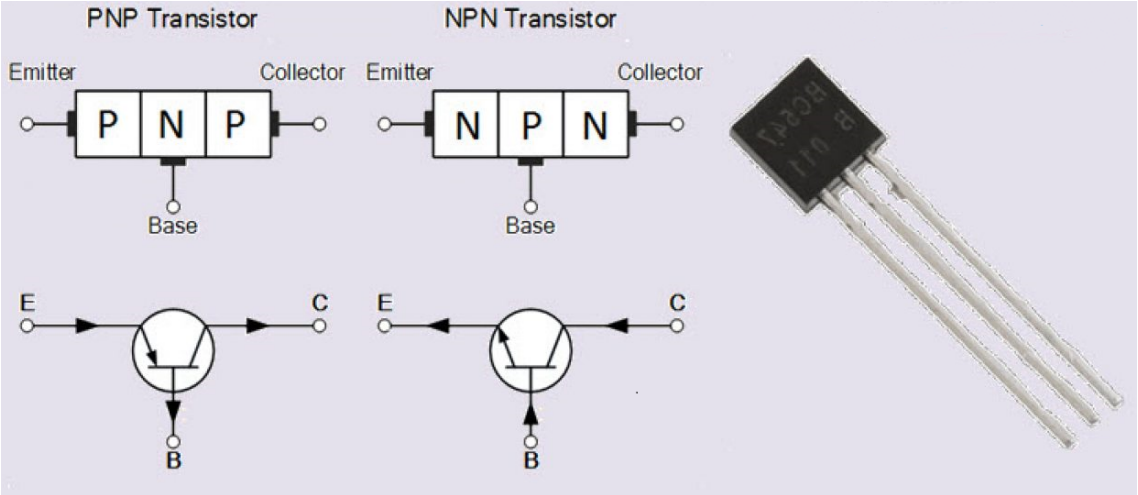


Figura 1.1: Transistores NPN y PNP

En general, su funcionamiento se basa en aplicar alguna corriente (o tensión en caso de los MOSFET) sobre alguno de los pines hasta superar un umbral, especificado en la hoja de datos del fabricante, para hacer circular corriente. Los más utilizados son los transistores **NPN**, (Ver figura 1.1). La diferencia básica entre los dos tipos de transistores es el hecho de que los BJT son controlados por **corriente** mientras que los FET son controlados por **tensión**. Al mismo tiempo, dentro de las familias de los FET, se pueden mencionar los MOSFETs, mencionados anteriormente, y los JFET (Junction Field Effect Transistor)[2].

1.1.1. Transistores tipo MOSFET

Los MOSFETS se clasifican en:

- Según el tipo de portadores del canal: CANAL-N y CANAL-P, con similitudes al NPN y PNP, respectivamente.
- Según el modo de formación del canal: Acumulacion y Enriquecimiento.
- Según las diversas geometrías usadas en la implementación: VDMOS, V-GROOVE ó TRENCH y LATERAL POWER MOS.

En el presente trabajo, se utilizó un **MOSFET de potencia** con las siguientes características:

- Canal N
- Enriquecimiento
- TRENCH

Un MOSFET de potencia se diferencia de uno regular, por los altos niveles de tensión y corriente que puede soportar, pero su funcionamiento básico es el mismo. El fabricante logra extender los límites de ruptura mediante la elección de la forma y los materiales del transistor.

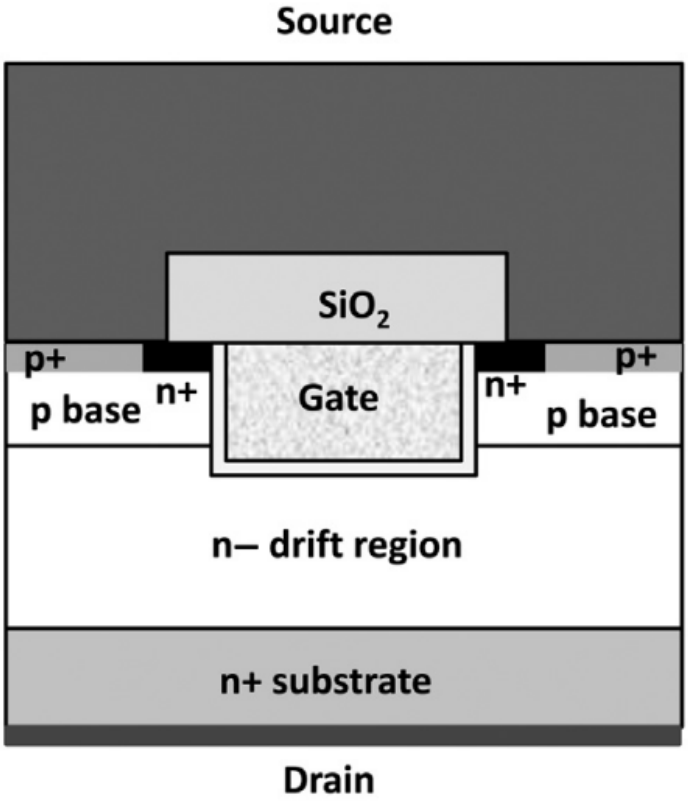


Figura 1.2: Estructura del MOSFET de potencia utilizado. [3]

En un MOSFET de Canal N, como el que puede verse en la figura 1.2, al aplicarse una tensión positiva en la terminal del *gate* se genera un campo eléctrico entre el *drain* y *source*, provocando

que los electrones se sientan atraídos al *gate*. Cuando la tensión supera determinado umbral (V_{th}), el campo eléctrico será lo suficientemente intenso, formándose una región rica en electrones denominada canal y permitiendo el paso de corriente del *drain* al *source*. El valor de V_{th} queda definido por los materiales utilizados y el proceso de fabricación. Aunque este es determinado por diseño, existe una variación considerable entre transistores del mismo tipo.

1.2 Temperatura de juntura

La temperatura a la cual se encuentra trabajando el semiconductor se denomina temperatura de juntura. Ésta tiene un máximo, el cual se encuentra especificado en la hoja de datos [4], y que al sobrepasarse puede influir en la degradación y, en algunos casos, en la rotura del dispositivo. Es por ello que será uno de los **parámetros a controlar en la degradación.**

La temperatura de juntura [5] se calcula mediante la ecuación 1.1, donde T_c es la temperatura de la cápsula del dispositivo, P_w es la potencia disipada [5] y Rth_{JC} es la resistencia térmica entre la juntura y la cápsula, definida en la hoja de datos del fabricante. Cabe destacar que la potencia disipada está fuertemente relacionada con la corriente de *drain* y la tensión entre el *drain* y el *source*, por lo que la variabilidad de cualquiera de estos 2 parámetros hará que ésta cambie.

$$T_j = T_c + (P_w * R_{th_{JC}}) \tag{1.1}$$

1.3 Efecto avalancha

El circuito de degradación del MOSFET de potencia (figura 1) posee una carga inductiva, de características similares al imán que se encuentra en el equipo de resonancia magnética nuclear. Cuando hay circulación de corriente a través del transistor ($V_{gate} > V_{th}$) se almacena energía en dicha carga. Durante la conmutación de apagado del MOSFET, cuando la señal de *gate* se lleva a $0V$, la energía almacenada en la bobina genera un pico de tensión en V_{ds}. En caso que ese pico supere la tensión de ruptura del transistor, comienza a circular corriente a través del mismo, disparando el consumo de potencia y generando un aumento repentino de la temperatura interna. Este evento se denomina “avalancha” [6].

Resulta de interés en el presente trabajo verificar que la energía disipada durante el pico de avalancha (Eas) no supere la máxima soportada por el MOSFET que se somete a degradación. Y también analizar el aumento de la temperatura de juntura durante la transición de apagado del mismo, al someterlo simultáneamente a alta tensión y corriente. El parámetro E_{as} depende de la duración de la avalancha (t_{av}) y la potencia pico disipada durante la misma ($P_{av(pk)}$). La potencia pico de avalancha depende de la corriente (Id_{av}) y la tensión (Vds_{av}) en el MOSFET durante la avalancha (ecuación 1.2), mientras que el tiempo de avalancha (ecuación 1.3) depende, además, de la inductancia de la carga inductiva (L). Luego, se calcula E_{as} con la ecuación 1.4.

$$t_{av} = (L * Id_{av})/Vds_{av}. \tag{1.2}$$

$$P_{av(pk)} = Id_{av} * Vds_{av} \tag{1.3}$$

$$E_{as} = (P_{av(pk)} * t_{av})/2 \tag{1.4}$$

Para obtener la temperatura de juntura T_j, se calcula en mediante la ecuación 1.5 la potencia promedio disipada ($P_{av(R)}$) como el producto entre la E_{as} y la frecuencia de los pulsos aplicados al *gate* del MOSFET (f). Luego, con la ecuación 1.6 se obtiene el aumento de temperatura debido a la avalancha, el cual depende de $P_{av(R)}$ y la resistencia juntura-cápsula del MOSFET (Rth_{JC}). Finalmente se obtiene $T\,j_{av}$ como la suma de la temperatura de juntura inicial ($T\,j_{inicial}$) y el aumento de temperatura debido a la avalancha (ecuación 1.7).

$$P_{av(R)} = E_{as} * f \tag{1.5}$$

$$\Delta T_j = P_{av(R)} * R_{th_{JC}} \tag{1.6}$$

$$T_{j_{av}} = T_{j_{inicial}} + \Delta T_j \tag{1.7}$$

Notar que el cálculo de la potencia promedio (ecuación 1.5) depende de la frecuencia de los pulsos aplicados al *gate* del MOSFET (f).

1.4 Área de operación segura (SOA)

El área de operación segura de un MOSFET se define como los valores que puede soportar de corriente y tensión para determinadas condiciones de operación: tiempo de pulso y temperatura, para que el mismo no esté en riesgos de rotura. Los límites del dispositivo en particular, se pueden obtener de la hoja de datos [4].

Puede observarse en la figura 1.3 que a medida que aumenta el tiempo de encendido del pulso aplicado al *drain*, la corriente que circula por el dispositivo debe ser menor para evitar que el mismo se rompa.

1.5 El MOSFET a degradar

El modelo del MOSFET a degradar es **IXTN660N04T4**. La tensión V_{ds} máxima en este dispositivo es de $40\,V$, mientras que el V_{th} se encuentra entre $2\,V$ y $4\,V$. La corriente máxima soportada son 200 A. La tensión V_{gs} máxima que puede aplicarse es $+/-15\,V$. La temperatura de juntura máxima es $175\,°C$ y la mínima de $-55\,°C$. **Cualquier violación de estos parámetros durante algún periodo de tiempo implica degradación o rotura del MOSFET**. El SOA de este dispositivo en particular puede verse en la figura 1.3

1.5.1. Las curvas de SOA del IXTN660N04T4

Para verificar si un punto está fuera del SOA [7], se utilizaron las curvas que se encuentran en la hoja de datos del transistor [4]. Sin embargo, dado que el fabricante no provee las ecuaciones asociadas a estas curvas, únicamente se cuenta con el gráfico proporcionado, por lo tanto, a partir del mismo se realizó una aproximación de las funciones asociadas. Para ello, se eligieron 2 puntos del gráfico proporcionado por el fabricante para cada curva y se aplicó regresión lineal utilizando el algoritmo de C++ (ver sección 4.1) sobre ellos para obtener la función potencial asociada.

- $R_{dsON} = {V_{ds}}^{0.834} * 10^{2.737}, V_{ds} \in [0.1\,V, 0.3\,V]$

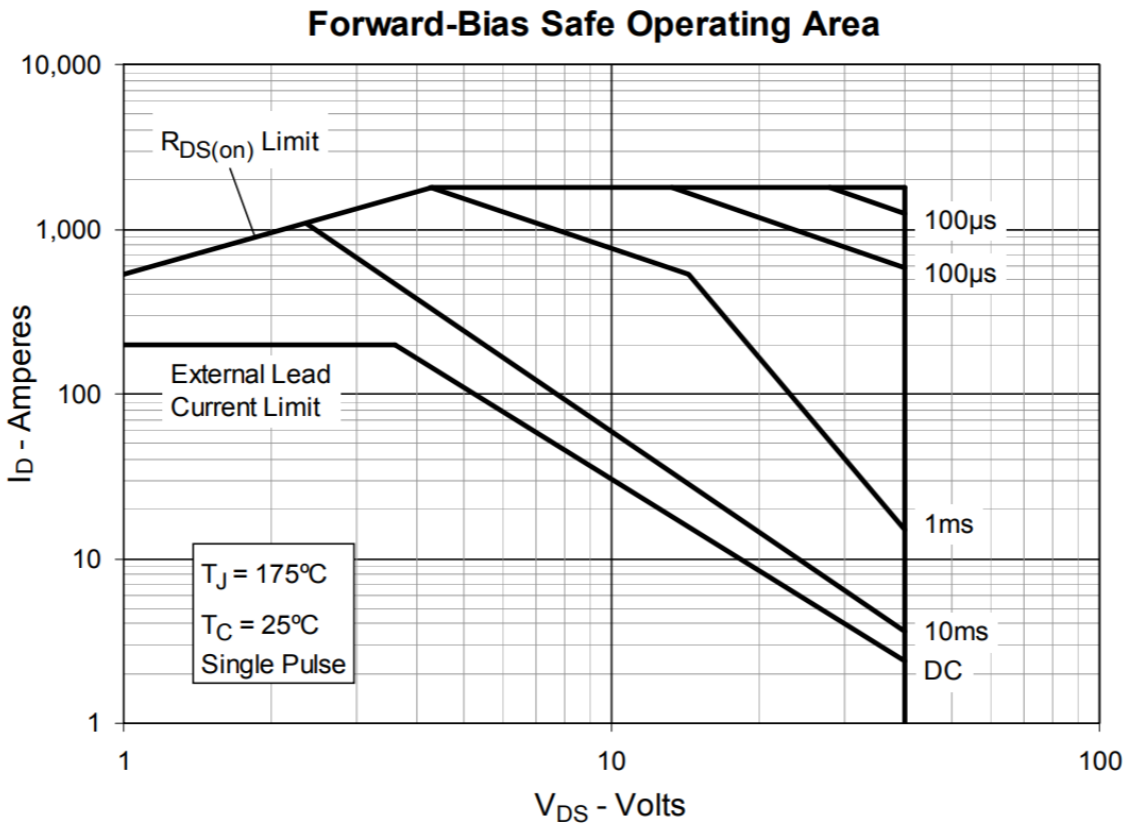


Figura 1.3: Área de operación segura de tensión entre el drain y el source vs corriente circulante

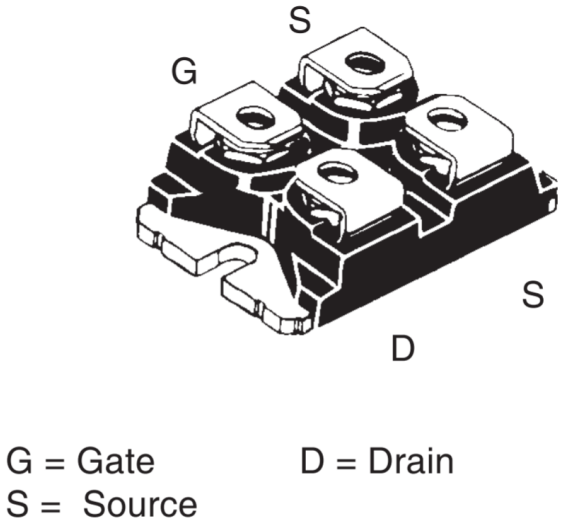


Figura 1.4: MOSFET IXTN660N04T4

- Encapsulado $= 200, V_{ds} \in [0.3V, 40V)$
- Deriva térmica
  - $DC = {V_{ds}}^{-1.84} * 10^{3.33}, V_{ds} \in [3V, 40V)$
  - $10ms = {V_{ds}}^{-2.04} * 10^{3.8}, V_{ds} \in [4.8V, 40V)$
  - $1ms = {V_{ds}}^{-3.415} * 10^{6.647}, V_{ds} \in [18V, 40V)$

- $V_{dsBR} = 0, V_{ds} = 40V$

En la figura 1.5 puede verse un gráfico con estas curvas aproximadas. En el eje horizontal se encuentra el valor de tensión *drain-source* y en el eje vertical la corriente de *drain*. Este gráfico representa la relación que hay entre ambos parámetros. Cuanto mayor sea la duración del pulso y la tensión *drain-source*, menor será la corriente que soporte el MOSFET.

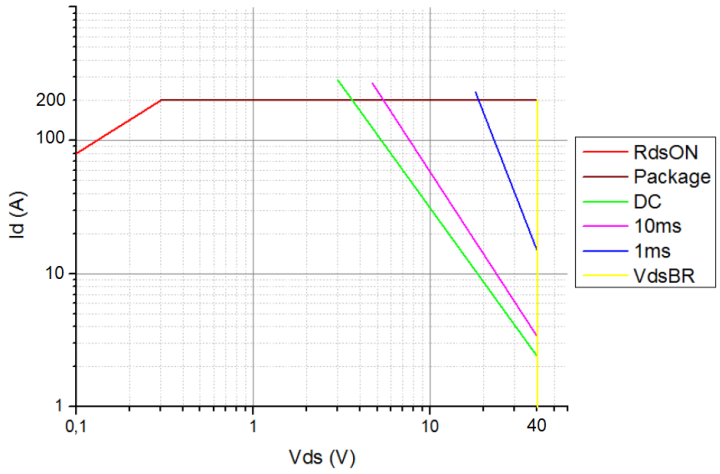


Figura 1.5: Funciones asociadas a cada curva de SOA

En las funciones de límite por deriva térmica, el tiempo indicado hace referencia a la duración del pulso del *gate*. En particular, la curva de 1 *ms* se linearizó para facilitar el cálculo ya que la diferencia se considera despreciable. Con el objeto de tener un margen de error, la curva que se utiliza para detectar si un punto se encuentra fuera del SOA es de un orden mayor en términos de tiempo, es decir, si el pulso de *gate* es de 5 *ms*, se utilizará la curva de 10 *ms*. Esto es debido a que el fabricante no provee una buena precisión en la hoja de datos.

A esto debe sumarse que estas curvas dependen de la temperatura de juntura del transistor, y por lo tanto, es necesario corregirlas cuando esta temperatura varíe. Para poder calcular este factor de corrección, primero se define la potencia máxima soportada del MOSFET, donde se asume que $T_j = T_c = 25\ ^\circ C$ como puede verse en la ecuación 1.8

$$P_{D25} = \frac{T_{j_{max}} - T_j}{Rth_{JC}} = \frac{175°C - 25°C}{0.144°C/W} = 1041.67W \tag{1.8}$$

Luego, se obtiene la potencia disipada del transistor a la temperatura de juntura actual (esto puede verse en la ecuación 1.9).

$$P_{D_{TMOSFET}} = \frac{T_{j_{max}} - T_j}{Rth_{JC}} \tag{1.9}$$

A partir de lo cual, se procede a calcular el factor de reducción de la potencia máxima soportada siguiendo la ecuación 1.10.

$$DeratingFactor = \frac{P_{D25}}{P_{D_{TMOSFET}}} \tag{1.10}$$

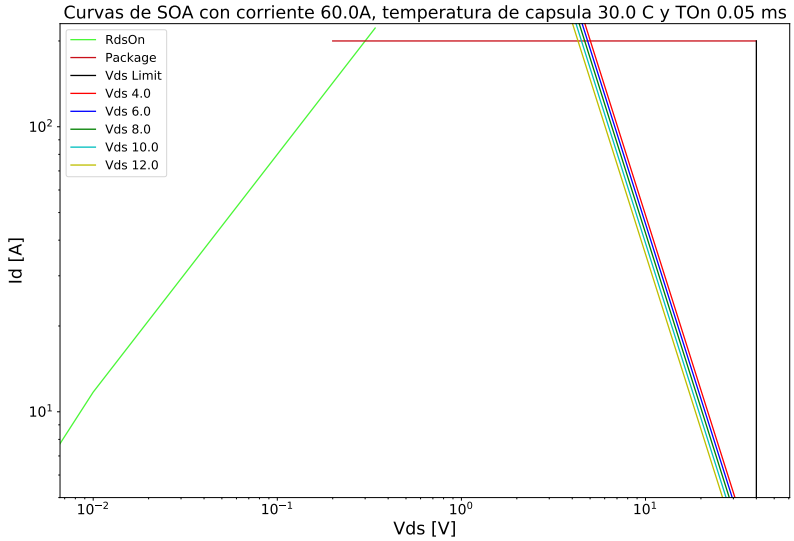
Finalmente, con el $DeratingFactor$, se obtiene el factor de corrección para ajustar la ordenada

al origen de la recta 1.12.

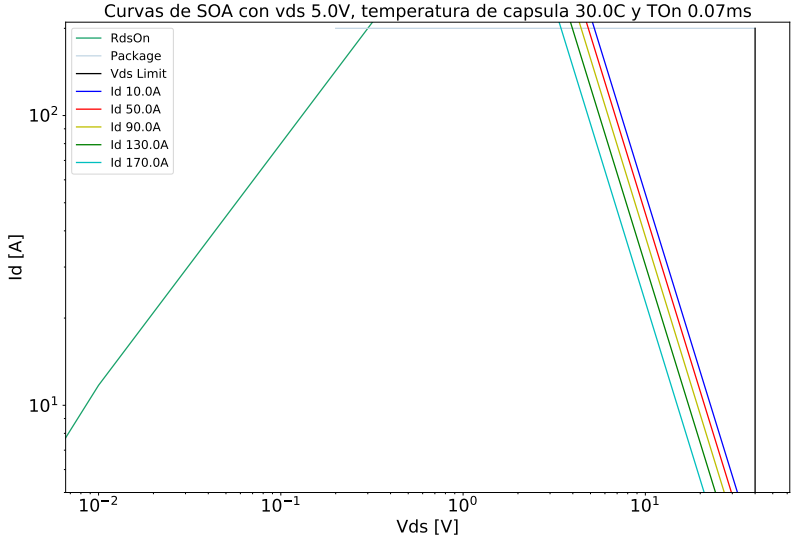
$$y = x^m * B, B = 10^b \tag{1.11}$$

$$B_{Derated} = B/DeratingFactor \tag{1.12}$$

Como se explico en la sección 1.2, la potencia disipada depende de la corriente de *drain* y de la tensión aplicada entre el *drain* y el *source*. Por lo tanto, en forma indirecta afectan al factor de corrimiento de las curvas SOA. Para analizar esta relación, se simularon y graficaron las ecuaciones previamente descriptas. La figura 1.6 muestra el ajuste de una curva de SOA para corriente variable y tensión de *drain* fijo y viceversa. Puede observarse como las curvas se mueven sobre la ordenada al origen dependiendo de cual parámetro sea el variable, alterando el valor máximo de corriente circulante que el MOSFET puede soportar para una determinada tensión en el *drain*.



(a) Curvas de SOA con tensión de *drain* variable con corriente y temperatura fijas



(b) Curvas de SOA con tensión de *drain* y temperatura fija pero con corriente variable

Figura 1.6: Corrimiento de curvas de SOA

1.6 Degradación del MOSFET de potencia

Mediante este trabajo se intenta emular el proceso de degradación que sufre un transistor MOSFET en un equipo de RMN-FCC. Un transistor de potencia puede presentar varios mecanismos de falla, por lo que es importante identificar cuáles son los principales en estas condiciones particulares en que se esta utilizando el dispositivo. En los aparatos RMN-FCC, se generan ciclos con distintas intensidades de campos magnéticos, como los que pueden verse en la figura 1.7. Estos distintos niveles de campos magnéticos, son directamente proporcional a la corriente que circula por el electroimán, siendo el transistor de potencia el encargado de controlar la intensidad de la corriente circulante y la conmutación rápida entre los distintos niveles. Para generar estos pulsos el transistor opera en modo lineal, con tiempos que equivalen a operación en corriente continua (DC).

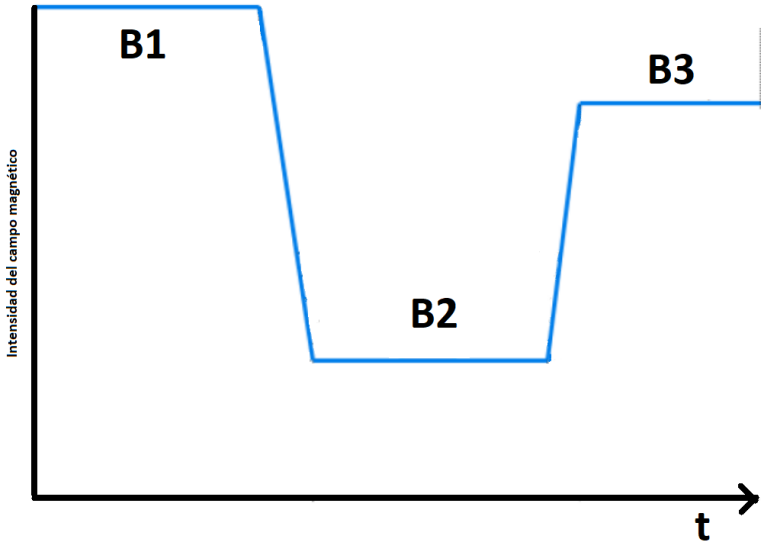


Figura 1.7: Secuencia típica de un ciclo en un aparato RMN-FFC

Por lo tanto, se pretende analizar la degradación del transistor debido a dos principales factores:

- Funcionamiento en modo lineal
- Conmutación de una carga inductiva

1.6.1. Degradación debido al funcionamiento en modo lineal

Se denomina modo lineal a aquel donde el transistor opera parcialmente encendido. En principio, este modo de trabajo es sencillo. Requiere aplicar una tensión entre el *gate* y el *source* (V_{gs}) cuya amplitud se mantenga dentro de los límites indicados en la hoja de información del MOSFET y que los valores de tensión *drain-source* (V_{ds}) y corriente en el transistor (I_d) permanezcan dentro del área de operación segura (SOA, Safe Operating Area). Sin embargo, operar en modo lineal es una de las aplicaciones de potencia más complicadas. Ya que la alta disipación de potencia puede provocar un rápido aumento de la temperatura de juntura (T_j) y el consecuente embalamiento térmico del dispositivo: la condición inestable que ocurre cuando T_j aumenta sin control.

Para emular este tipo de degradaciones en un transistor de potencia, se aplica una señal cuadrada de baja frecuencia en el *gate* del transistor, de forma en que se lleva al dispositivo a trabajar en modo lineal. En la figura 1.8 puede verse un diagrama simplificado del circuito utilizado para este tipo de degradación. Un parámetro a configurar es el nivel de corriente (alta o baja) que se utiliza para el proceso, también es posible modificar la frecuencia de la señal de *gate*, el ciclo de trabajo, entre otros parámetros.

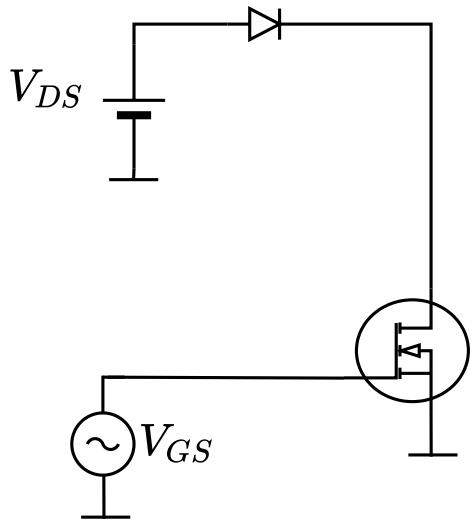


Figura 1.8: Circuito relacionado a la degradación por alta/baja corriente.

1.6.2. Degradación debido a la conmutación de una carga inductiva

Cada vez que un MOSFET se somete a un ciclo de encendido y apagado se induce un estrés electro-térmico que tiende a debilitar sus materiales. Este estrés es intensificado cuando se utiliza una carga inductiva por el proceso de "avalancha" explicado en la sección 1.3. Para emular este proceso de degradación, se utiliza un circuito similar al descripto en la sección anterior pero con el agregado de un inductor similar al utilizado en un equipo RMN-FFC, este puede verse en la figura 1.9. La señal de *gate* que se utiliza en este tipo de degradación, es una señal cuadrada cuya frecuencia puede configurarse a valores superiores al de degradación en modo lineal.

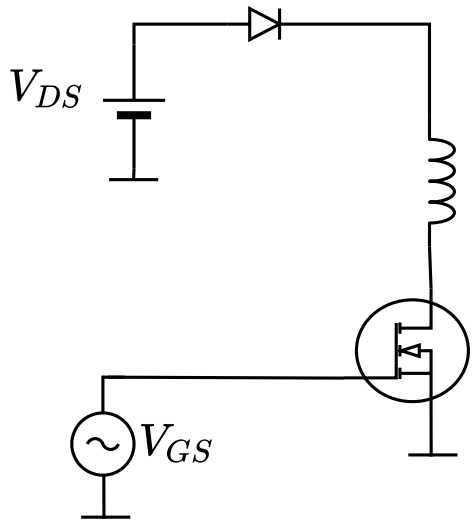


Figura 1.9: Circuito relacionado a la degradación por conmutación.

2. Hardware utilizado

Para poder realizar las degradaciones mencionadas en la sección 1.6, se requiere de múltiples recursos además de un transistor de potencia. Es necesario generar una señal para aplicar en el *gate* del MOSFET cuyas características dependen del tipo de degradación que se desea efectuar y de otros parámetros configurables como la frecuencia, ciclo de trabajo, amplitud, etc. y por lo tanto, es necesario contar con un microcontrolador que reciba esta configuración y elabore la señal de *gate* correspondiente. Además, es necesario controlar que el ensayo se encuentre dentro de ciertos límites de operación, para esto, es necesario utilizar un conjunto de sensores para medir la temperatura de cápsula T_c, la corriente de *drain* I, la tensión de *gate* V_{gs} y la tensión de *drain* V_{ds}. Finalmente, es necesario contar con algunas medidas de protección para reaccionar cuando el ensayo viola los limites establecidos: ventilador para poder reducir la temperatura del MOSFET, interruptor para apagar la fuente y así detener el ensayo, etc. Los elementos que se incorporan a los circuitos de las figuras 1.8 y 1.9, pueden verse en el circuito final de la figura 2.1.

Como se mencionó en la introducción de este informe, la plataforma de degradación acelerada se puede dividir en cuatro partes principales:

- Software de la computadora
- Microcontrolador, con su respectivo software
- Etapa de adaptación de señales
- Etapa de potencia

El software de la PC cuenta con una interfaz gráfica mediante la cual se configuran los parámetros de degradación, que luego son enviados al microcontrolador programable por USB.

El microcontrolador utilizado ofrece una extensa gama de recursos de hardware para realizar una amplia variedad de operaciones, tales como acondicionamiento de señales, conversión analógica a digital, digital a analógica y tratamiento digital de señales en tiempo real, por lo que resulta altamente efectivo para implementar la instrumentación en cuestión. El mismo se encarga del control del proceso y es la interfaz entre los valores medidos en el circuito de potencia y la computadora, permitiendo recopilar información relevante de las condiciones de trabajo, almacenarla en una base de datos local y observar gráficamente parámetros críticos durante todo el ensayo.

La etapa de adaptación de señales se encarga de amplificar las señales que ingresan a la etapa de potencia y de reducir las señales que ingresan al microcontrolador, con el objeto de que se encuentren en los niveles requeridos por cada dispositivo. Dentro de la etapa de potencia se cuenta, principalmente con una carga inductiva que emula el imán del equipo de RMN-FFC (sólo para degradación en conmutación), el dispositivo a degradar y una fuente de alimentación. Además, el sistema cuenta con un ventilador para acelerar el enfriado del transistor que es controlado por el microcontrolador.

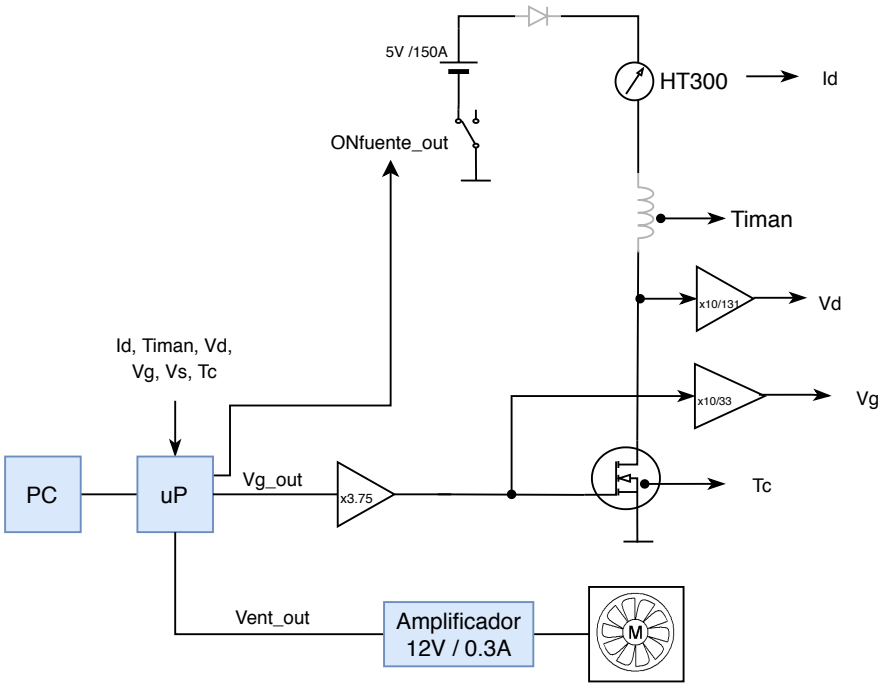


Figura 2.1: Circuito de la planta del MOSFET.

2.1 Etapa de adaptación de señales

Las señales que salen del microcontrolador a la etapa de potencia deben ser amplificadas para que se encuentre en los niveles requeridos por la etapa de potencia, a su vez, las señales que ingresan al microcontrolador proveniente de los sensores, deben ser reducidos para que este no sea dañado. Estos acondicionamientos de señales, se realizan en la etapa de adaptación de señales.

En concreto, como puede verse en la figura 2.1, las señales que salen del microcontrolador y que necesitan ser amplificadas son:

- V_{g_out}: Señal aplicada al *gate* del transistor
- $Vent_{out}$: Encendido del ventilador para refrigerar el transistor (figura 2.2)

Se considera necesario aclarar que $ONfuente\_out$, la señal de encendido de la fuente de alimentación no requiere ningún tipo de acondicionamiento. Por otro lado, al microcontrolador ingresan las tensiones V_g y V_d. Ambas necesitan ser reducidas al rango de trabajo de los conversores analógico-digital.

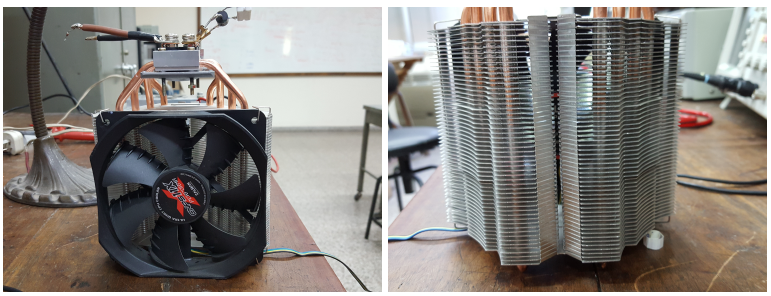


Figura 2.2: Ventilador y disipador Zalman CNPS11X Performa

Existen múltiples maneras de sensar corriente [8]. En este caso, para medir la corriente que

circula por el transistor se utiliza un sensor de efecto Hall **HT300** debido a los altos niveles de corriente que se requiere medir. La relación que hay entre la tensión medida y la corriente de *drain* puede verse en la ecuación 2.1. Esta relación se obtuvo experimentalmente. Cabe aclarar que vale para tensiones mayores a 0. Cuando la tensión es 0, el valor de corriente sensado es 0.

$$Id = 3 + 88.5 * V_{Hall}, V_{Hall} > 0 \tag{2.1}$$

Para la adecuación de todas estas señales se utilizan amplificadores operacionales, los cuales se describen a continuación.

2.1.1. Amplificador

Un amplificador operacional (Op Amp), es un dispositivo que tiene como función principal amplificar una diferencia de tensión entre sus entradas. La tensión de salida se obtiene como se ve en la ecuación 2.2.

$$V_{out} = A_v * (V_+ - V_-) \tag{2.2}$$

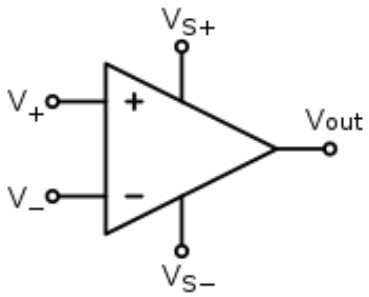


Figura 2.3: Diseño de amplificador operacional

Asimismo, dicho amplificador, como se puede apreciar en la figura 2.3, posee un único terminal de salida V_{out} y dos terminales de entrada (V_+ o entrada no inversora y V_- o entrada inversora). Además presenta dos terminales de alimentación ($\pm Vs$) y un terminal de **masa**. Dependiendo de como se realicen las conexiones, se obtendrán distintas configuraciones, como por ejemplo: Amplificador inversor, Amplificador no inversor, Integrador, Derivador, etc. En particular, el **amplificador no inversor** es el utilizado en la placa de adaptación de señales, ya que únicamente amplifica la señal de entrada (ver figura 2.4) y las tensiones del experimento son positivas. Este recurso físico se utiliza para obtener señales significantes cuando la tensión de entrada es del orden de los milivoltios.

Estos son necesarios para amplificar las tensiones producidas por el PSoC, ya que este trabaja con hasta un máximo de 3.3 V, valor que está por debajo de las tensiones de *gate* del transistor de potencia requeridas en los experimentos. Además, en la placa de adaptación de señales, se encuentran operacionales que reducen las tensiones que ingresan al PSoC. Esto es necesario para **no saturar** o incluso **romper** el microcontrolador por proveer tensiones superiores a 3.3 V.

Como puede verse en la figura 2.1, existen tres amplificadores diferenciales en modo amplificador no inversor. El amplificador de la tensión de *gate* producida por el PSoC tiene una ganancia de **3.75**, mientras que el coeficiente divisor del amplificador que reduce la tensión de *drain* será de

13.1 y el que reduce la tensión de *gate* leída por el PSoC es de **3.3**

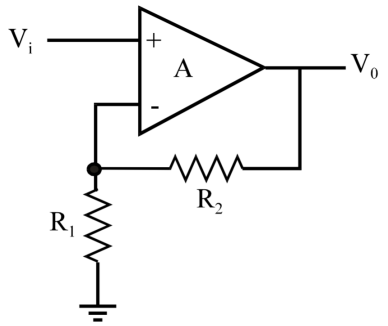


Figura 2.4: Diseño de amplificador no inversor

2.2 Etapa de potencia

La etapa de potencia esta conformada, principalmente, por la fuente de alimentación, el imán y el transistor de potencia. Dado que, mediante esta plataforma se pretende emular la degradación a la que un transistor de potencia es sometido en un aparato de RMN, para algunos tipos de degradaciones se utiliza un imán de características similares al utilizado en estos equipos para generar el campo magnético [9]. Éste posee una inductancia de 5 μH y una resistencia de 19 $m\Omega$. La fuente de alimentación es del tipo *switching* y su encendido es controlado por un relé con bobina de 12 V y contactos que soportan hasta 220 V, con 10 A de corriente. Éste es activado o desactivado por el PSoC en situaciones específicas, que será detallado más adelante.

2.3 Microcontrolador

Para poder tener un control sobre las señales que actuarán sobre la etapa de adaptación de señales y el MOSFET, fue necesario utilizar algún microcontrolador capaz de comunicarse con la PC y que pueda manejar señales analógicas. Se decidió utilizar un microcontrolador de la empresa *Cypress semiconductor* denominado PSoC. Este es programable en C, puede comunicarse por USB y brinda un conjunto de herramientas fáciles de utilizar para la conexión e interacción de circuitos analógicos.

Se optó por utilizar este microcontrolador porque tiene facilidad para trabajar con señales analógicas y es lo suficientemente potente y rápido para enviar y recibir datos en cuestión de microsegundos. Además, su entorno de desarrollo ya tiene gran parte de lo necesario para poder interactuar con el MOSFET.

El PSoC® (Programmable System-on-Chip) es un dispositivo en el cual se incorpora un sistema configurable digital, uno analógico y un microcontrolador dentro de un único chip. A su vez, su arquitectura interna se puede dividir en tres grandes partes: bloques analógicos y digitales configurables, una CPU y un sistema de enrutamiento programable y de interconexión (ver figura 2.6). Cabe destacar que se pueden reconfigurar dinámicamente las entradas y salidas de los recursos analógicos y digitales. Algunas características de la arquitectura de estos microcontroladores son:

- Reloj tanto interno como externo.
- El voltaje de referencia puede ser variado para actuar con distintos sensores.

- Voltaje de funcionamiento de 5 voltios ó 3.3 voltios.

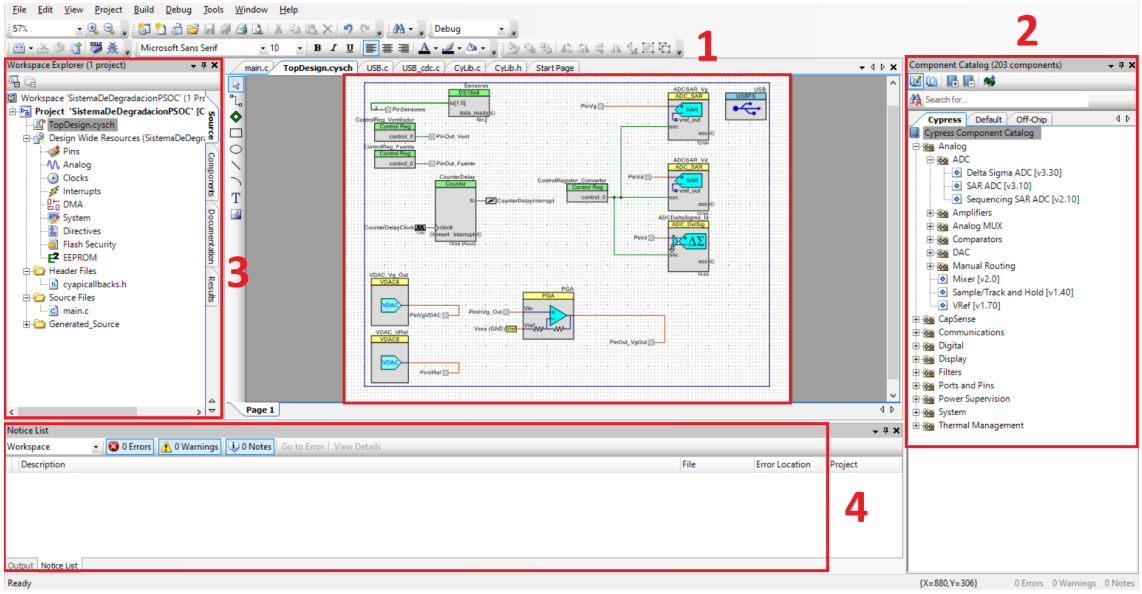


Figura 2.5: (1) Diagrama de bloques que tienen una representación física en la placa. (2) Bloques definidos por Cypress. (3) Organización del proyecto, configuración de pines, frecuencias de clocks, etc. (4) Errores y advertencias del compilador.

Los PSoC, dependiendo de la familia que se esté utilizando, pueden utilizar para su programación diferentes entornos de desarrollo. Para alguno de los modelos, uno de ellos el PSoC 5LP, ofrece un entorno de desarrollo llamado PSoC Creator, utilizado para programar el *SoC*. En la figura 2.5 se muestra la interfaz gráfica del software utilizado para configurar y programar el PSoC 5LP. Esta herramienta cuenta con distintos bloques digitales y analógicos completamente implementados por el fabricante (ver cuadro 2 de figura 2.5), de forma en que el usuario solo tiene que instanciarlos. Por otro lado, los procesadores van cambiando según la familia con la que se esté trabajando. Por ejemplo:

- Familia PSoC 1: Procesador M8C que alcanza los 24MHz.
- Familia PSoC 3: Procesador de ciclo individual 8051.
- Familia PSoC 5: Procesador ARM Cortex-M3 de 32-bit.

2.3.1. El modelo utilizado

Se utilizó el PSoC 5LP CY8C5868AXI-LP035, que contiene un Cortex M3 de 67 *Mhz* como CPU y trabaja con 3.3 *V* (aunque puede ser configurado para trabajar a 5V). Este microprocesador es de 32 bits y no soporta operaciones de punto flotante de forma nativa. El compilador debe adaptar cualquier declaración de una variable de punto flotante para el microprocesador. Algunas características relevantes del procesador Cortex es que tiene un pipeline de 3 etapas y un predictor de saltos condicionales, además de instrucciones especiales para sincronización de memoria (no permitir escribir/leer en memoria si hay otra instrucción accediendo a ella). Posee 16 registros de 32 bits, aunque solo 13 de ellos pueden usarse libremente. Los otros 3 son el Stack Pointer, Link Register y Program Counter. No contiene una caché de datos, solo de instrucciones.

El PSoC tiene **100 pins**. Más información puede encontrarse en la hoja de datos o en la página oficial [10].

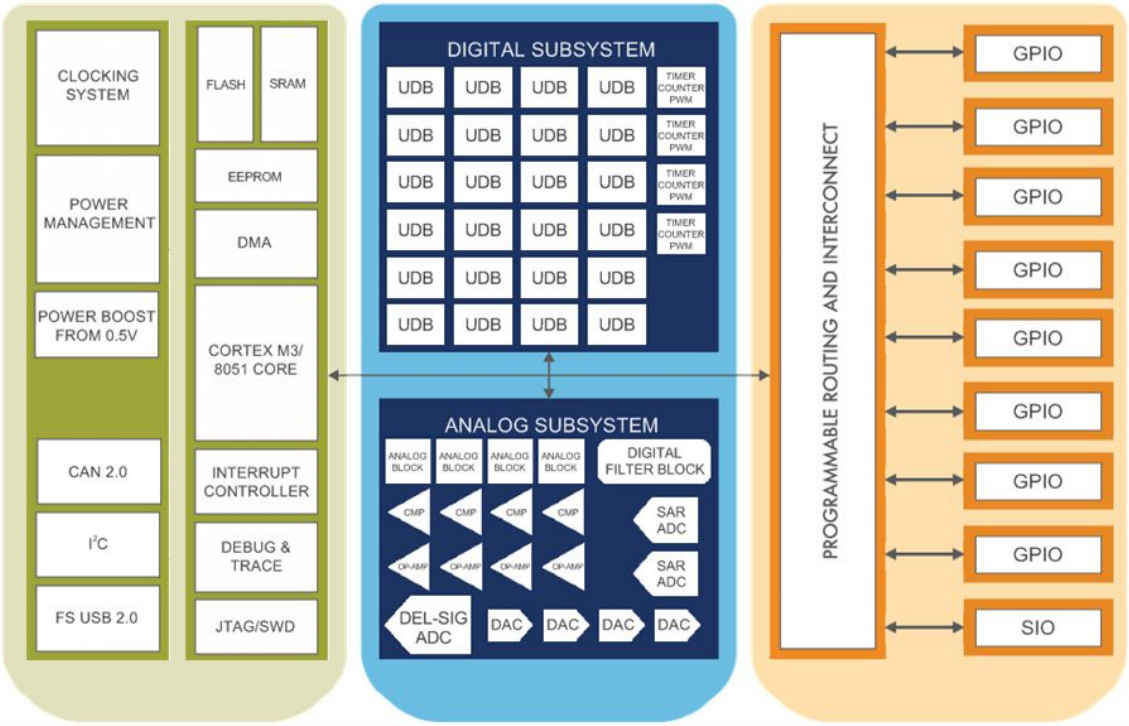


Figura 2.6: Subsistemas del PSoC

El *SoC* puede verse dividido en 3 partes conectadas entre si, como puede verse en la figura 2.6, denominados subsistema de CPU, subsistema digital y analógico y subsistema de interconexión programable. Este ultimo recurso se encarga de configurar las conexiones internas entre los circuitos analógicos y/o digitales, además de permitir la interacción con el exterior mediante puertos *GPIOs*. A continuación se detallan el subsistema de la CPU y el digital y analógico.

2.3.2. El subsistema de la CPU

En esta parte se encuentra el microprocesador *Cortex* M3, para poder ejecutar las instrucciones que el usuario haya programado utilizando C. Tiene 3 tipos de memoria, una EEPROM, una SRAM de 64KB y una memoria FLASH (no volátil) de 256KB.

El sistema de *clocking* provee 4 *clocks* internos y 2 externos que utilizan osciladores de cristales para una mayor velocidad. Además, provee múltiples divisores para poder decrementar la frecuencia de ellos. Cabe destacar que el *clock* de USB es totalmente independiente del resto de los *clocks*.

El **controlador de interrupciones** soporta hasta 32 interrupciones donde se puede variar la prioridad (entre 0 y 7 inclusive, siendo 0 la mayor y 7 la menor) y avisará a la CPU que debe atender la interrupción, ya sea provocada desde software o provocado por algún componente físico. CAN 2.0, I2C y FS USB2.0 son los bloques encargados de la comunicación de datos entre el microcontrolador y dispositivos de entrada y salida que soporten alguno de esos 3 protocolos.

2.3.3. El subsistema digital y analógico

Como puede verse en la figura 2.6, el subsistema analógico cuenta con 2 conversores analógico-digital **SAR** y uno del tipo **Delta Sigma**, así como 4 conversores digital-analógico. Los bloques analógicos son programables, tal que con ellos se pueden utilizar *PGAs*, *Mixer*, *Sample & Hold* y *Trans-Impedance Amplifier*.

En el subsistema digital, se puede observar que los únicos bloques que ya están implementados en el *PSoC* son 4 **timers** que pueden ser utilizados como **PWMs**. Todos los otros bloques son **UDBs** (Universal Digital Block).

Un UDB es un bloque programable un poco más complejo que la PLD (*Programmable Logic Device* [11]. Con ellos se puede implementar cualquier función que involucre un circuito digital. Con el entorno que Cypress ofrece, puede hacerse uso de dichos bloques con funciones ya implementadas, o crearse las propias [12], aunque raramente se necesite crear una nueva funcionalidad ya que en el uso común, la mayoría de los recursos ya están implementados.

2.3.4. Bloques y elementos utilizados en el PSoC

En la figura 2.7 puede observarse el circuito final de bloques utilizado en el PSoC.

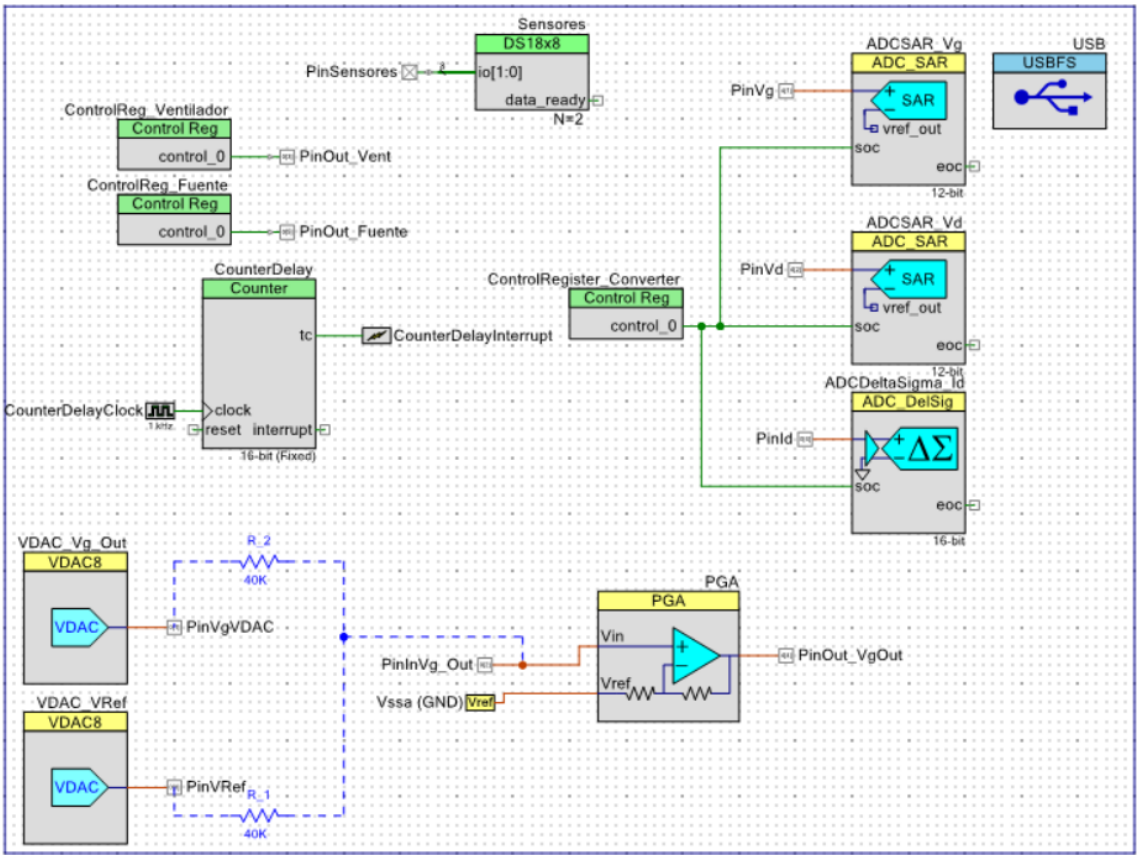


Figura 2.7: Circuito desarrollado en PSoC Creator para la degradación

Para simplificar las mediciones y por una falta de recursos físicos por parte del PSoC (ya que se utilizan todos los conversores analógico-digital disponibles), la tensión asociada al *source* no se mide. Las conexiones físicas para medirla fueron hechas, pero para un trabajo futuro.

A continuación se menciona el uso de cada bloque para luego detallarlos y mostrar su configuración. El nombre de cada recurso físico puede observarse en la parte superior de cada bloque.

- ControlReg_Ventilador: Señal de control que indica el encendido o apagado del ventilador para enfriar el MOSFET.
- ControlReg_Fuente: Señal de control que enciende o apaga el relé que controla la fuente switching.
- ControlRegister_Converter: Señal de control que sincroniza la conversión de todos los conversores analógicos-digitales.
- Sensores: Bloque de comunicación con los sensores de temperatura digitales para medir la temperatura de cápsula del MOSFET e imán del circuito.
- ADCSAR_Vg: Sensor analógico-digital que convierte en binario la tensión en el *gate* del MOSFET para ingresar al PSoC y enviarlo a la PC.
- ADCSAR_Vd: Sensor analógico-digital que convierte en binario la tensión del *drain* del MOSFET para ingresar al PSoC y enviarlo a la PC.
- ADCDeltaSigma_Id: Sensor analógico-digital que convierte en binario la corriente circulante del circuito para ingresar al PSoC y enviarlo a la PC.
- USB: Bloque que permite la comunicación USB con la PC utilizando el protocolo UART.
- VDAC_Vg_Out: Conversor digital-analógico de tensión que será utilizado sobre el pin de *gate*.
- VDAC_VRef: Conversor digital-analógico de tensión que será utilizado sobre el pin de *gate*.
- PGA: Amplificador de ganancia variable configurado para actuar como sumador de las tensiones de VDAC_VRef y VDAC_Vg_Out.
- CounterDelay: Contador para medir el tiempo de espera después de mandar un paquete de datos a la PC.

2.3.5. Conversores analógicos-digitales (ADC)

El conversor analógico-digital convierte una determinada tensión V en un valor binario para que pueda ser interpretado por un circuito digital. Es necesario definir una tensión de referencia, que represente el “0”, y la tensión máxima posible, además de la resolución (cantidad de bits para representar el valor máximo). A mayor resolución, más preciso será el conversor, pero puede ser mucho más costoso y/o lento. Se utilizaron 2 tipos de conversores analógicos-digitales ya que son los que provee el PSoC que serán detallados a continuación.

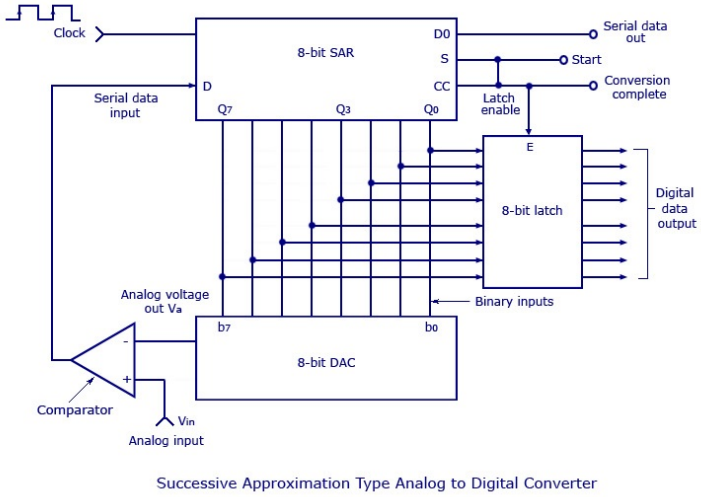


Figura 2.8: Circuito del conversor ADC SAR

2.3.5.1. Conversor analógico SAR

El conversor analógico digital SAR (figura 2.8) funciona de la siguiente manera:

1. Ingresa una tensión V_{in} por un recurso físico llamado **Sample & Hold**, que mantiene esta señal igual hasta que el conversor termine (en la figura 2.8 el bloque se considera implicito en la entrada). Puesto que si la tensión de entrada cambiase mientras se realiza la conversión, el valor de salida sería erróneo.

2. Una vez que se mantiene la señal de entrada, se comienza colocando un 1 en el bit más significativo y los demás bits puestos a 0.

3. Luego se utiliza un DAC para reconstruir la señal analógica, llamémosle V_a, a partir de la palabra digital formada. A su vez, V_a es comparado con V_{in}, del siguiente modo. Si este valor es más grande que la señal de entrada, el 1 se remueve y se coloca un 0, en caso contrario, se mantiene y se escribe un 1 en el siguiente bit menos significativo.

Este proceso se realiza N veces, donde N es la **resolución del conversor**. Cabe destacar que la cantidad de pulsos o ciclos necesarios para realizar una conversión son **N+2**, ya que 1 ciclo es utilizado para empezar la conversión, los N ciclos siguientes son para cada bit y el último ciclo es para alertar sobre el fin de la conversión.

Las características configurables [13] de este recurso en el PSoC incluye:

- Resolución de 8 bits hasta 12 bits.
- Conversiones por segundo.
- Clock interno y externo para realizar la conversión.
- Rango de voltaje que puede leer.
- Voltaje de referencia externa, interna o bypassed.

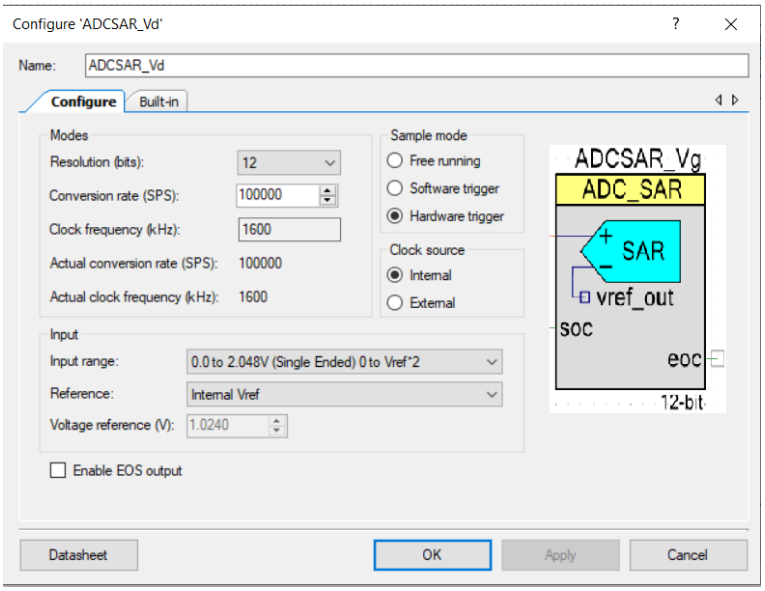


Figura 2.9: Bloque del ADC SAR en el PSoC junto a su configuración

- Comienzo de muestreo por hardware, software o conversión continua.

Para este trabajo, se utilizaron 2 ADC SAR, uno para la tensión de *gate* y otro para la tensión del *drain*. Ambos fueron configurados de la misma manera y puede verse en la figura 2.9

La resolución elegida fue de **12 bits ya que es la máxima permitida**. Cuan más alto sea el tiempo de muestreo, mayor será el consumo de corriente del conversor, pero al estar conectados a una PC por USB, este consumo no es de interés, por lo que siempre se utiliza la mayor velocidad posible, es decir, **100000 muestras por segundo** o 1 muestra cada 10 microsegundos. El rango de entrada es de 0 V **a** 2.048 V porque es lo que nos da una mejor precisión que las demás opciones. Se decidió usar activación por hardware ya que se desea **sincronizar la conversión de todos los ADCs.**

2.3.5.2. Conversor analógico digital Delta Sigma

En el modulador sigma-delta, en su estructura básica, se produce un tren de pulsos cuya proporción entre pulsos positivos (unos) y pulsos negativos (ceros) corresponde al valor analógico de la señal de entrada 2.10.

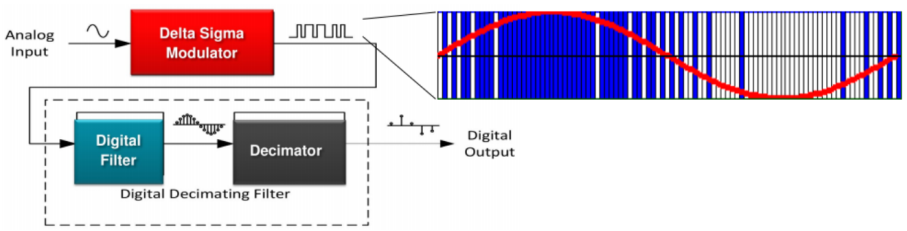


Figura 2.10: Funcionamiento del conversor Delta Sigma

Una mayor cantidad de 1's lógicos (azul) corresponden a una señal de entrada de mayor amplitud, por el contrario, si predominan los 0's (blanco), corresponde a una señal de entrada de amplitud baja (o nula). La trama de unos o ceros es entonces alimentada a un filtro pasabajos,

mismo que elimina las componentes de alta frecuencia, dejando como resultado una suave señal que es alimentada a un decimador[14] El decimador cuantizará la señal continua, es decir, la va a representar en una discreta, y entregará a la salida una señal binaria codificada, correspondiente al valor de la señal analógica de entrada.

El microcontrolador solo posee 1 conversor de este tipo ya implementado que se utiliza para convertir la corriente que circula por el circuito medida por el sensor de efecto Hall **HT300**, ya que necesitamos de una precisión más grande de 12 bits (que es la que tiene como máximo el SAR) para tener una mayor certeza del valor leído para así evitar posibles rupturas del transistor. Los parámetros configurables son:

- Tipo de conversión: Continua, muestreo simple, muestreo múltiple y muestreo múltiple turbo.
- Resolución de 8 bits hasta 20 bits, aunque mayor resolución implica una **cantidad de muestras por segundo menor**.
- Ratio de conversión.
- Rango de voltaje de la entrada.
- Ganancia de buffer de 1, 2, 4 u 8.
- Modo del buffer rail to rail, level shift y bypass buffer.
- Tipo de voltaje de referencia.
- Clock externo o interno.
- Modo de entrada single ended o diferencial, donde single ended tiene la conexión del pin negativo a la masa analógica (VSSA) y diferencial permite conectar una tensión negativa al conversor.
- Comienzo de conversión por hardware o software.

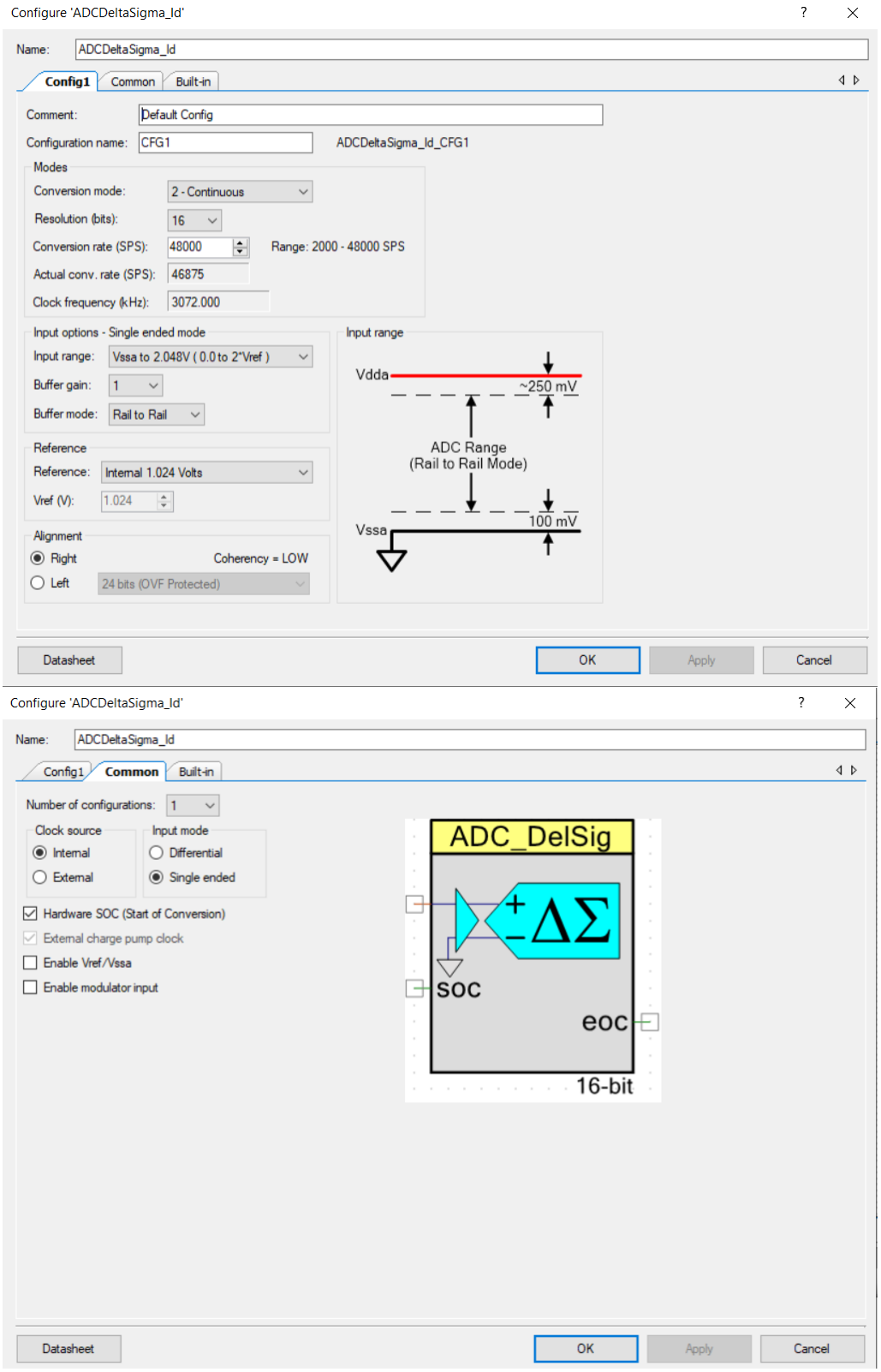


Figura 2.11: Bloque del ADC Delta Sigma en el PSoC junto a su configuración

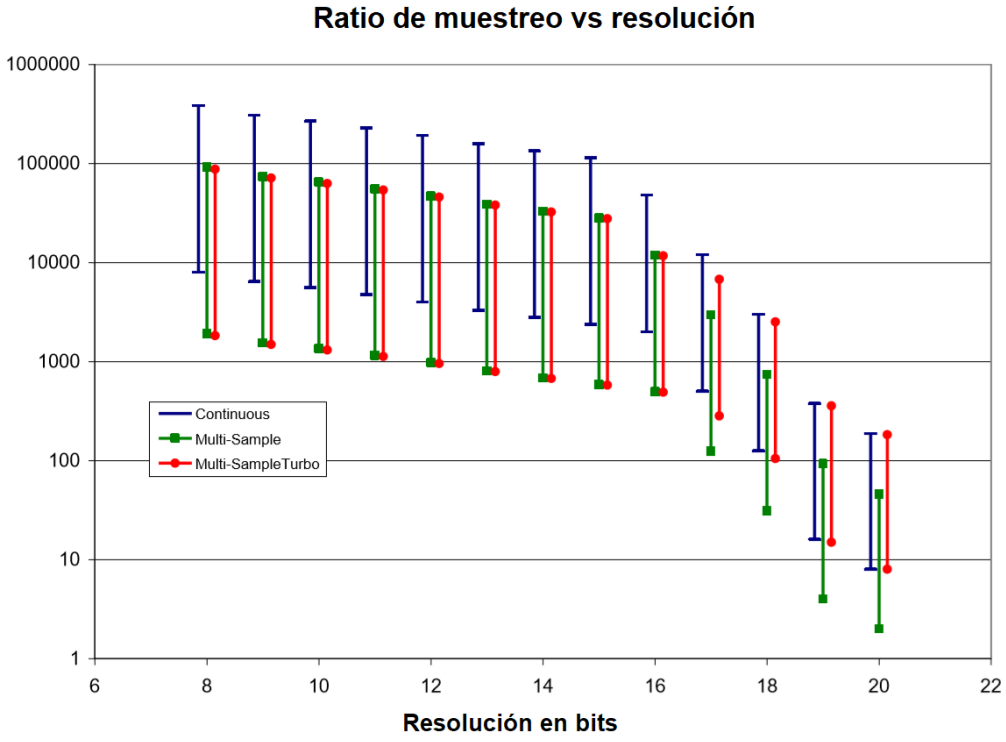


Figura 2.12: Ratio de muestreo del ADC Delta Sigma vs Resolución

La configuración puede verse en la figura 2.11. La figura 2.12 muestra la relación de tiempo de muestreo con respecto a la resolución y el modo de conversión. Se puede observar que el modo de conversión continuo es el más veloz, por lo que se utiliza dicho modo junto a 16 bits de resolución (es un poco más que el doble de lo que tarda un conversor SAR, $22\mu s$ o 48000 muestras por segundo). Como la tensión producida por el sensor hall es siempre positiva, se utilizó el modo de entrada **single ended**. La corriente máxima que puede circular por este transistor es de $150A$, utilizando la ecuación 2.1, se obtiene que la tensión máxima que puede obtenerse a la salida del sensor es de $1,66\,V$. Este valor, se encuentra dentro del rango de entrada del conversor, el cual es de $0\,V$ **a** $2.048\,V$.

2.3.6. VDAC

Un conversor digital analógico convierte un valor, en binario, a una determinada tensión o corriente. Para este experimento, se utilizaron 2 conversores digital analógico de tensión que provee el microcontrolador.

Las características de este en el PSoC son:

- Resolución de 8 bits.
- Rango de conversión de 0 a 1.020V (4 mV por bit) o de 0 a 4.080V (16 mV por bit).
- Conversión de datos por escritura en registro o bus de datos del propio VDAC.

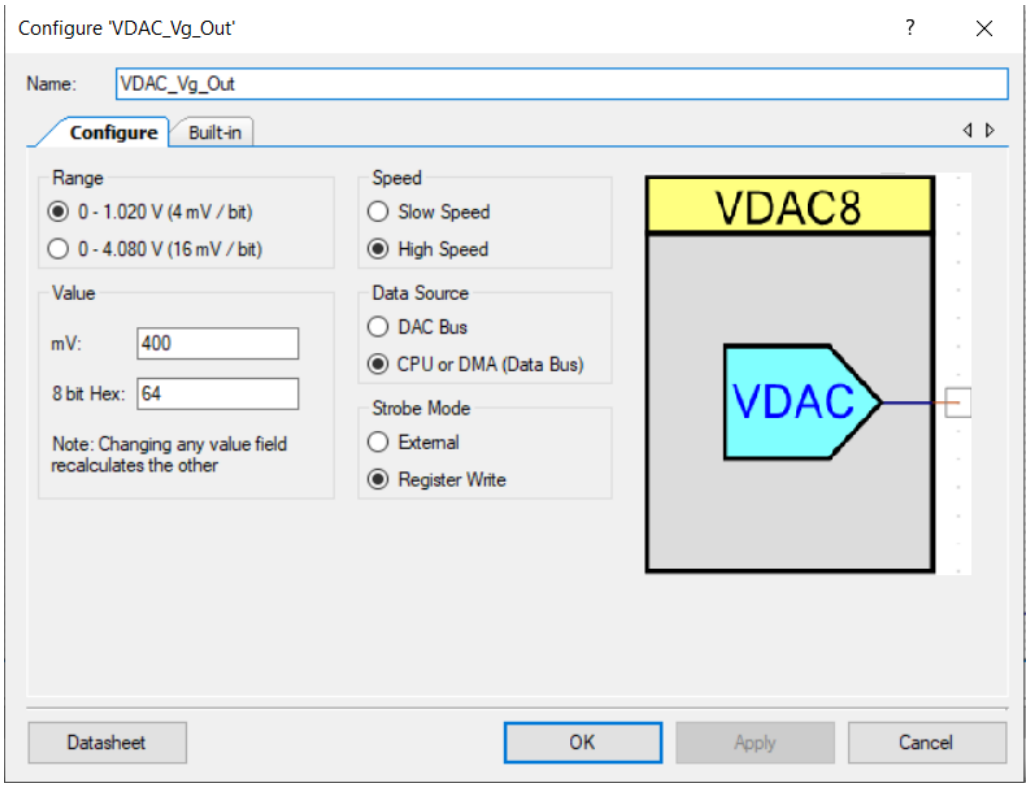


Figura 2.13: Bloque de VDAC en el PSoC junto a su configuración

Para el experimento se utilizaron dos VDAC para ampliar el rango de valores aplicados posibles en el *gate*. Para ello, se utilizó el rango de $0\,V$ a $1.020\,V$. La configuración de ambos es la misma y puede verse en la figura 2.13.

2.3.7. Programmable Gain Amplifier (PGA)

Un *PGA* es un amplificador tal que la ganancia puede ser controlada analógicamente o digitalmente. Éstas pueden ser configuradas para actuar como un sumador de señales analógicas, como se muestra en la 2.15.

Este recurso se utiliza debido a que el PSoC provee dos configuraciones para el rango de tensiones del VDAC. Uno entre $0\,V$ y $1,020\,V$ con $4\,mV$ por bit, mientras que la otra configuración brinda entre 0 y $4,040\,V$ con $16\,mV$ por bit. Si bien el segundo modo brinda la posibilidad de alcanzar tensiones mas altas, el PSoC está configurado para trabajar con alimentación de $3,3V$, por lo que, este modo no se puede utilizar. Es por ello que se utiliza un bloque PGA junto a 2 conversores digital-analógico entre el rango 0 a $1.020\,V$, donde cada VDAC tiene asignado la mitad del valor de *gate* configurado por el programa. Por lo tanto, en el puerto de salida se obtiene la suma de estos dos valores ($V_{g_out} = V_{ref} + V_g$). De esta forma, se logra mantener la precisión de $4\,mV$ por bit, pero en el rango de tensiones útiles.

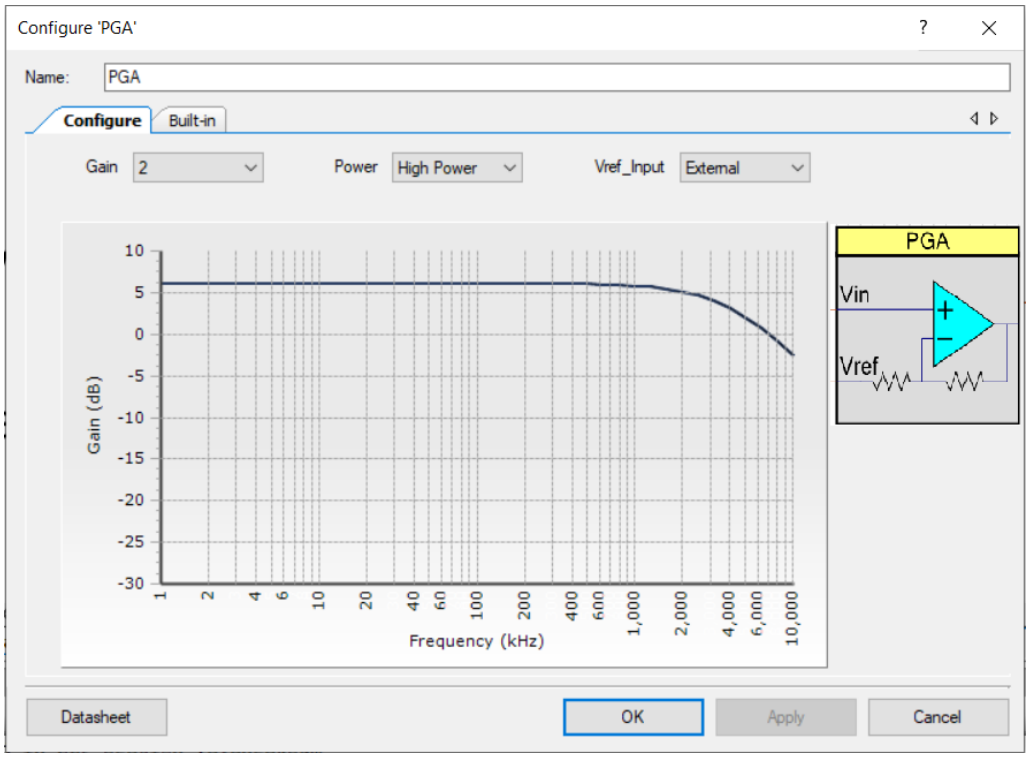


Figura 2.14: Bloque del PGA en el PSoC y su configuración

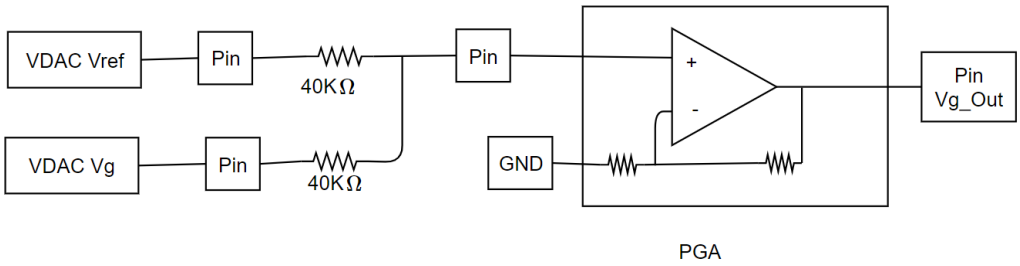


Figura 2.15: Circuito que se encuentra en la protobard del PSoC para la PGA

Ganancia	Rb	Ra
1	0	40 kΩ
2	40 kΩ	40 kΩ
4	120 kΩ	40 kΩ
8	280 kΩ	40 kΩ
16	600 kΩ	40 kΩ
24	460 kΩ	20 kΩ
32	620 kΩ	20 kΩ
48	470 kΩ	10 kΩ
50	490 kΩ	10 kΩ

Tabla 2.1: Resistencias asociadas a la PGA en base a la ganancia. Extraído del datasheet de Cypress

La ecuación 2.3 muestra la relación entre la entrada y la salida del circuito sumado mostrado en

la figura 2.15. Dado que se desea obtener ganancia $G = 1$, todas las resistencias del circuito deben ser iguales. Para esto, se configura el PGA con ganancia 2, de manera que las dos resistencias internas del módulo se establezcan en 40kOhm (tabla 2.1), y se conectan externamente otras dos resistencias del mismo valor entre las señales a sumar (VDAC asociado al *gate* y VDAC asociado a V_{ref} [15]) y la entrada no inversora del PGA.

$$V_{g_{out}} = G * (VDAC_{Vref} + VDAC_{Vg}) \tag{2.3}$$

$$V_{g_{out}} = VDAC_{Vref} + VDAC_{Vg} \tag{2.4}$$

De esta forma, el rango de tensiones que se obtienen a la salida del PGA es de $0\,V$ a $2,040\,V$. Como se explicó en la sección 2.1.1, este valor es multiplicado por $3,75$ en la etapa de adaptación de señales, por lo tanto, el rango final de tensiones de *gate* que pueden utilizarse se encuentran entre $0\,V$ y $7,65\,V$.

2.3.8. Contador

Un contador es un bloque digital (ver figura 2.16) que cuenta la cantidad de pulsos que recibe. Si a la entrada se le conecta la señal del reloj principal, se puede utilizar como un temporizador, ya que, dependiendo de la frecuencia de reloj y la cantidad de pulsos a contar se puede modificar el tiempo a medir.

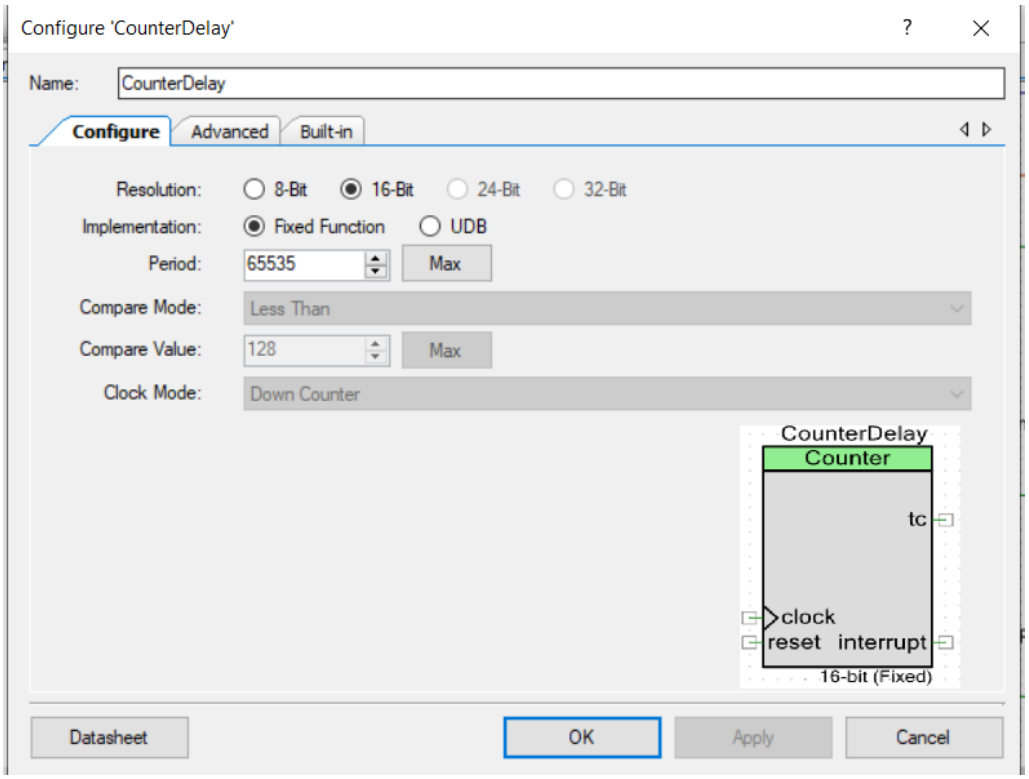


Figura 2.16: Bloque en el PSoC del contador y su configuración

Se observó que utilizando la función de **delay** implementada por Cypress, para esperar un tiempo determinado, retrasaba el envío de datos por USB, pero no era el caso al utilizar un con-

tador. La necesidad de utilizar este recurso entre el tiempo de envío de datos y la continuación de la degradación surge del requerimiento de **repetitividad** del ensayo. Dado que el tiempo que la PC tarda en realizar los cálculos necesarios entre cada envío de datos y su respuesta es variable (depende del poder de computo con el que se cuenta y el disponible en el momento), se decidió fijar un tiempo algo mayor al requerido por una PC estándar actual para realizar estos cálculos. De forma en que sea posible poder replicar estos ensayos aunque se modifique la PC que se está utilizando.

2.3.9. Registro de control

Un registro de control es una posición de memoria cuyo contenido se refleja en el estado lógico de un puerto de salida GPIO. Éste es controlado **únicamente mediante software.** Se utilizaron **3 registros de control de 1 bit**. Uno para comenzar la **conversión síncrona** (ver figura 2.17) de los conversores ADC asociado al *gate*, *drain* y la tensión asociada a la corriente sensada por el sensor de efecto Hall. Los otros 2 registros son para **encender el ventilador y apagar el relé** que está conectado a la fuente de alimentación del circuito.

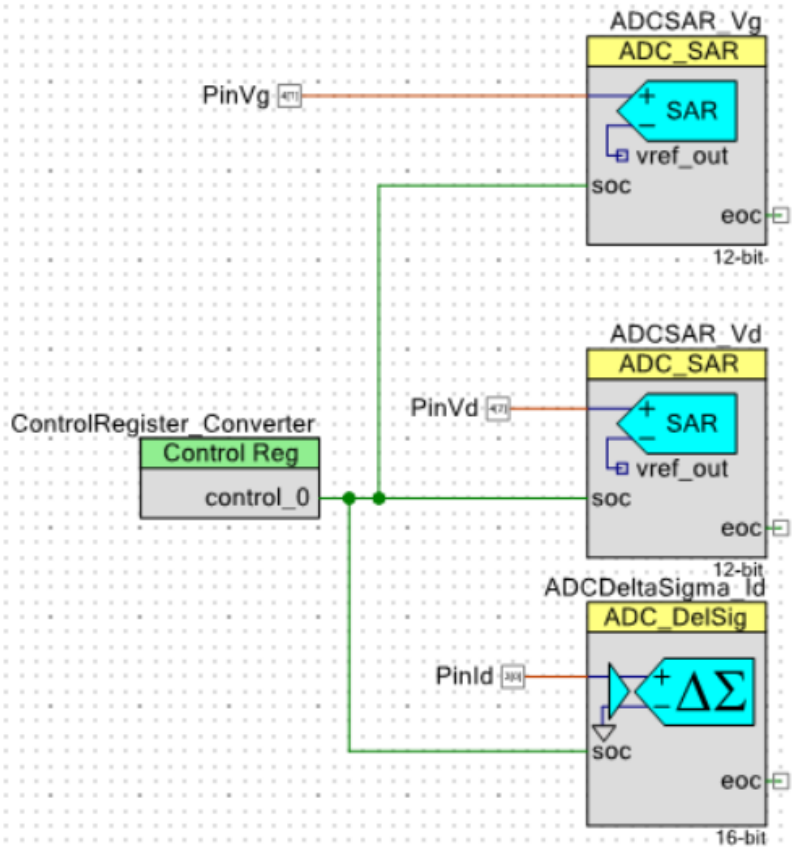


Figura 2.17: Registro de control para sincronizar la conversión de sensores

2.3.10. UART

Finalmente, para realizar la comunicación con el software en la PC (que se encuentra a cargo de analizar los datos obtenidos), fue utilizada el bloque de conexión USB que el microcontrolador ya posee. El protocolo utilizado para la comunicación es **UART**.

2.3.11. Sensor de temperatura D18S20

Para el sensado de la temperatura del imán y del encapsulado del transistor se utilizan dos sensores iguales denominado DS18S20, el cual provee mediciones de temperatura de 9 bits expresadas en grados **Celsius**.

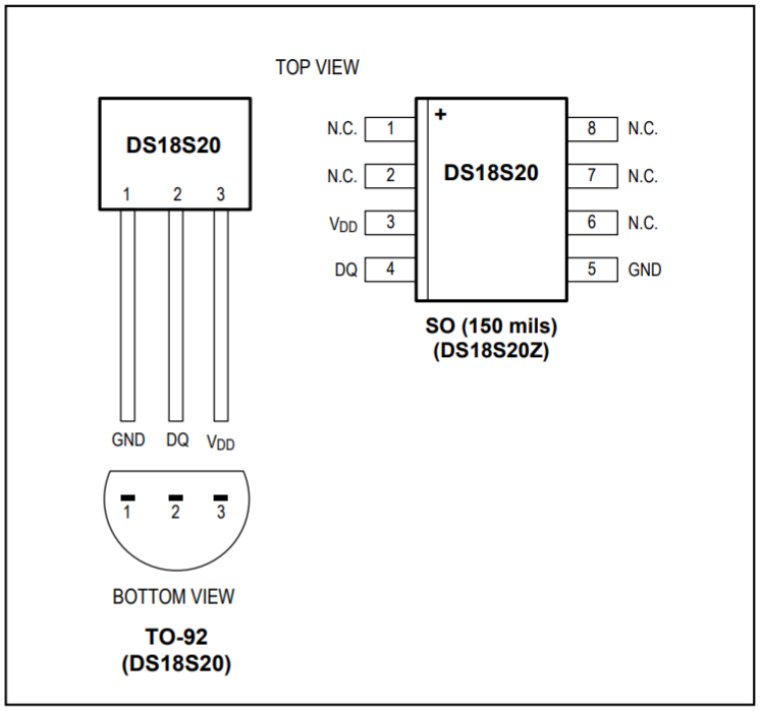


Figura 2.18: Sensor digital DS18S20

Éste utiliza el protocolo OneWire, diseñado por Dallas Semiconductor, para comunicarse con el microcontrolador. Dicho protocolo consiste en transmitir la información únicamente por un pin *DQ*, por lo que se deben enviar códigos especiales para poder iniciar la conversión de temperatura, leer la memoria, entre otras funciones. Todo esta información se encuentra en el **datasheet** [16] del sensor. A continuación se mencionan algunas características de este:

- Su memoria consiste de un SRAM Scratchpad y una EEPROM no volátil.
- Posee un solo pin de puerto para la comunicación llamado *DQ*.
- Mide temperaturas de -55 °C a 125 °C (-67 °F a 257 °F).
- ± 0.5 ° C de Precisión entre -10 ° C y 85 ° C.
- Tiempo de conversión fijo de 750 *ms*.

La función principal del DS18S20 es sensar temperatura y convertirla a digital con una resolución de hasta **9 bits**. Se enciende en un estado inactivo de baja potencia, y a través de un comando, se inicia la medición de temperatura y, posteriormente, se hace la conversión de analógico a digital. Después de la conversión, la temperatura sensada se almacena en los primeros 2 bytes de la ScratchPad, figura 2.20a, y vuelve a su estado inactivo. La estructura de la temperatura puede verse en la figura 2.20b.

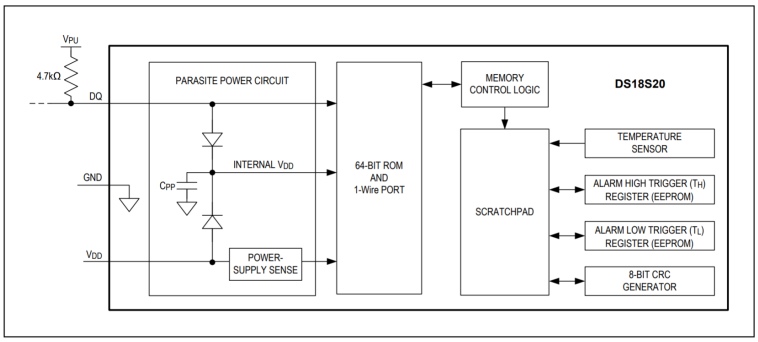
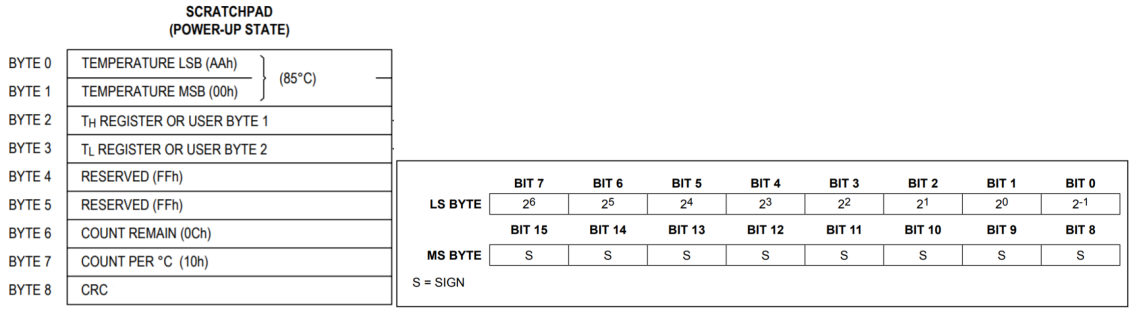


Figura 2.19: Circuito del sensor DS18S20



(a) Organización del ScratchPad del sensor

(b) Interpretación de los 2 primeros bytes que representan la temperatura

Figura 2.20: La memoria e interpretación de los registros de temperatura del sensor

Los bits de signo (S) indican si la temperatura es positiva o negativa: para números positivos $S = 0$ y para números negativos $S = 1$. Para calcular una resolución superior a 9 bits, se utilizan los datos de los registros de temperatura, **CountRemain** y **CountPer**°C del Scrathpad. Después de leer el Scratchpad, el valor de la temperatura se obtiene truncando el bit de 0.5 $^{\circ}C$(bit 0) de los datos de temperatura. Luego, la fórmula que se utiliza para obtener la temperatura puede verse en 2.5, donde **CountRemain** y **CountPerC** se encuentran en el byte 6 y 7 de la ScratchPad respectivamente.

$$Temperatura = TemperaturaLeida - 0.25 + (CounterPerC - CountRemain)/CountPerC \quad (2.5)$$

En el trabajo, se utilizan **2** de estos sensores de temperatura para medir **la temperatura de la cápsula** que cubre el MOSFET y otro para medir **la temperatura del imán**. Para poder hacer uso de este sensor, fue necesario crear la comunicación con el protocolo antes mencionado. Se modificó ligeramente un código funcional para el sensor DS18B20 hallado en los foros oficiales de Cypress[17] para utilizar los bytes adicionales de CountPerC y CountRemain que el sensor DS18S20 contiene.

2.4 Conexionado entre el PSOC y la etapa de potencia

Finalmente, la figura 2.21 muestra como los bloques que se nombraron en las secciones anteriores se comunican con la etapa de potencia, en caso de ser necesario, mediante la etapa de

adaptación de señales mostrada en la figura 2.1.

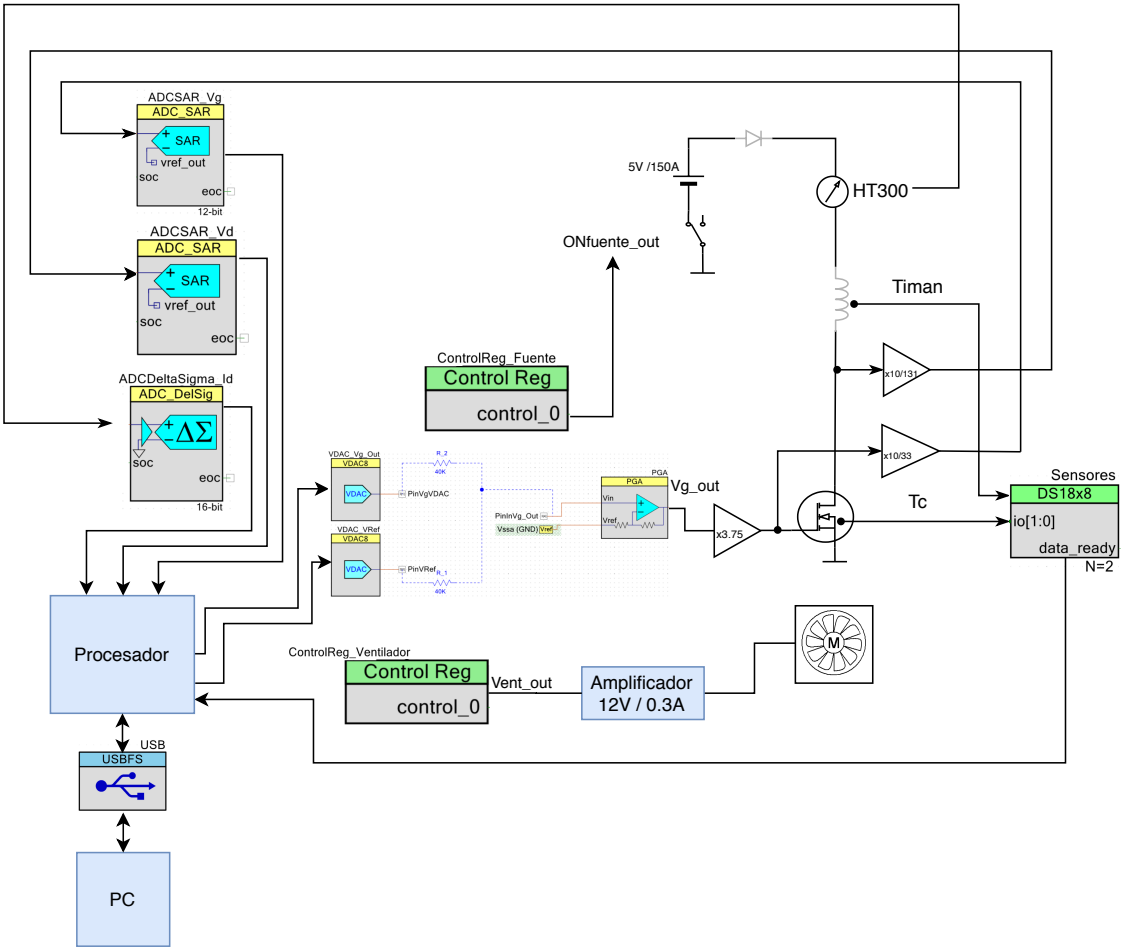


Figura 2.21: Conexionado entre el PSOC y la etapa de potencia

El código que ejecuta el microcontrolador se puede dividir en las siguientes etapas:

- Configuración inicial de bloques: Cuando se enciende el PSoC, inicializa todos los bloques detallados anteriormente en los valores iniciales.
- Espera de datos de degradación: El microcontrolador se queda en pausa hasta que los datos de una degradación lleguen por USB.
- Medición de datos por ciclo: Comienza la degradación, por lo que el PSoC comenzará a realizar las mediciones acordes.
- Verificación de límites extremos: El PSoC verifica 2 límites (corriente circulante menor a 150 A y violación por punto fuera de SOA) tal que no puede esperar a que la PC le confirme si hubo una violación o no, ya que podría romper el MOSFET antes de la respuesta.
- Envío de datos a la PC: El PSoC envía los datos obtenidos de la degradación a la PC.
- Toma de acción correctiva en base al error: Dependiendo de las circunstancias, el microcontrolador puede tomar la decisión de terminar la degradación o tomar alguna acción correctiva para continuar con el experimento.

La figura 2.22 muestra lo descripto anteriormente en el orden que fueron nombrados, de izquierda a derecha y de arriba hacia abajo.

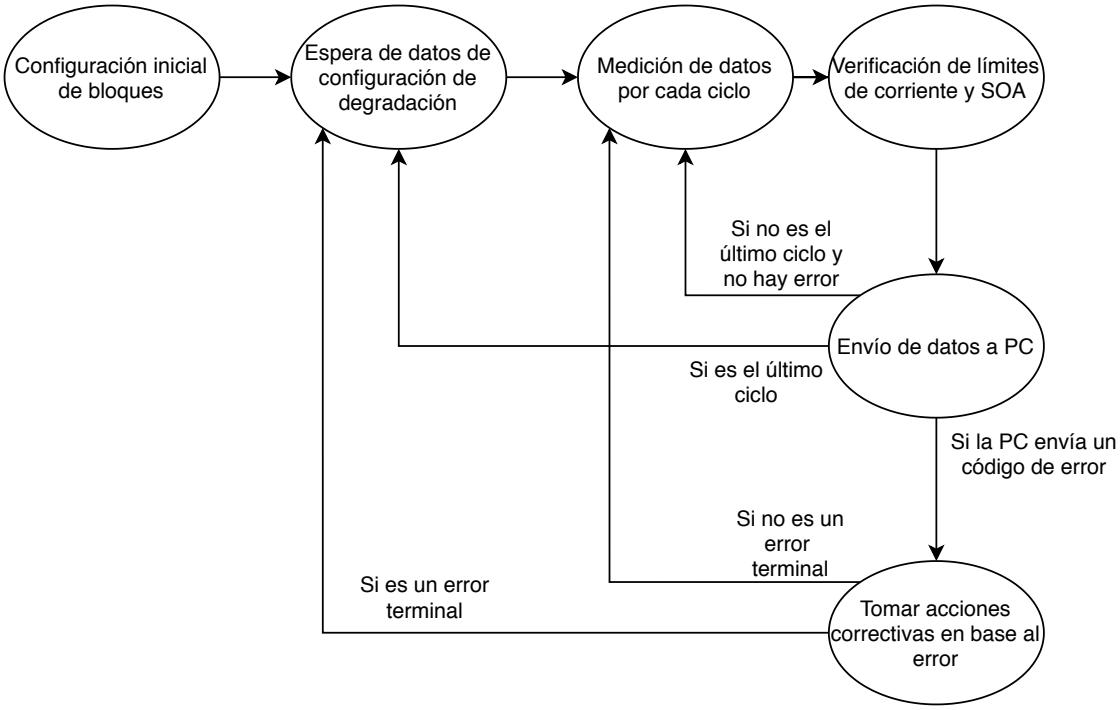


Figura 2.22: Programa que ejecuta el PSoC

3. Proceso de degradación del transistor

La degradación de un MOSFET de potencia, o cualquier componente electrónico en general, provoca variación en las características físicas de este con las que fueron fabricados. Algunas de ellas pueden ser:

- En caso de conectores eléctricos, el aumento de la resistencia de contacto del componente por el uso de este, influenciado por la temperatura y humedad del ambiente. También, ésta puede aumentar por el estrés mecánico de los pines conectores por un uso extendido del componente.
- Utilizar el componente en las condiciones extremas durante un tiempo prolongado acorta la vida de los materiales que componen el recurso físico.
- **Estrés eléctrico:** Utilizar niveles de voltaje y/o corriente que superan los recomendados por el fabricante del componente. Esto hace que la vida útil del mismo se reduzca [18].

Dos enfoques serán utilizados para realizar la degradación del MOSFET, denominados: **alta/baja corriente y conmutación.** El primero se basa en la aplicación de pulsos prolongados (baja frecuencia) al *gate* del transistor, con *duty cycle* bajo. En este modo el transistor trabaja con pulsos que se consideran de corriente continua (DC) y es posible generar un rápido aumento de la temperatura interna del mismo, usando alta corriente, o uno más lento y prolongado, usando baja corriente. Por su parte, en la degradación por conmutación se aplican pulsos cortos al *gate* del transistor (alta frecuencia) y se utiliza la carga inductiva para generar picos de tensión entre *drain* y *source* cada vez que se interrumpe el flujo de corriente (sección 1.3).

3.1 Degradación por alta/baja corriente

En la degradación por alta/baja corriente [19], por cada pulso que se aplica sobre el *gate* se harán 3 mediciones en distintos instantes. La primera 500 μs después de comenzar el pulso. La segunda 22 μs antes de finalizar el pulso y la última 100 μs después de que el pulso de *gate* sea 0, como puede verse en la figura 3.1. Notar que en esta figura solo se muestra la medición de Vg y Id, pero en ese mismo instante también se mide la tensión en *drain*. Además, como el sensor de temperatura tarda un tiempo alto (respecto a los pulsos de *gate*, ver sección 2.3.11) en realizar la conversión y la temperatura no sufre cambios abruptos en poco tiempo, se mide cada 1 segundo.

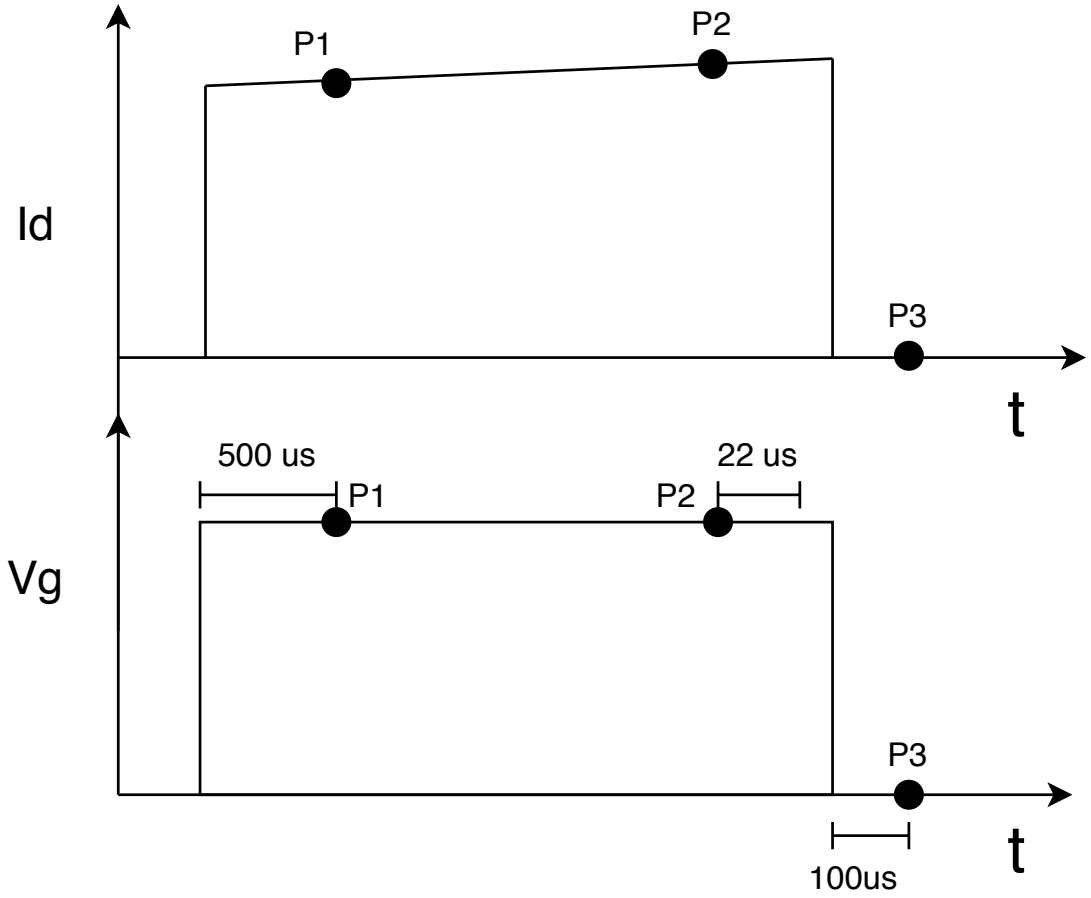


Figura 3.1: Un pulso en el *gate* del transistor.

Antes de realizar la primer medición (P1), es necesario esperar 500 μs para que la corriente circulante por el circuito se estabilice. Esto se debe a que en el momento en el que se brinda tensión en el *gate*, hay un pico de corriente que no representa la corriente de *drain* efectiva que circula el resto del tiempo del pulso. La segunda medición (P2) es para observar el estado actual de la corriente, que debería de ser mayor a la primera medición debido al embalamiento térmico. El tiempo esperado es el requerido por el conversor AD. La tercer medida (P3), se realiza 100 μs después de cortar el pulso de *gate* para esperar que la corriente de *drain* sea efectivamente 0.

Es relevante notar, por las aclaraciones previas, que la señal del *gate* en alto (T_{on}) debe permanecer por lo menos 522 μs y en bajo (T_{Off}) 100 μs, por lo que la relación entre frecuencia y *duty cycle* debe cumplir con esos parámetros, es decir, debe cumplir las ecuaciones 3.1 y 3.2 tal que $T_{On} >= 522\ \mu s$ y $T_{Off} >= 100\ \mu s$.

$$T_{On} = DutyCycle * T/100 \tag{3.1}$$

$$T_{Off} = T - T_{On} \tag{3.2}$$

Es de suma importancia respetar estos límites, ya que, tiempos más pequeños no permitirían que los sensores adquieran valores correctos debido a que las señales no se podrían haber establecido. Por lo tanto, la frecuencia máxima debe ser de 1608 Hz, ya que el período sería igual a 622 μ

s, con un *duty cycle* exacto de 78.852 % para que $T_{On} = 522\ \mu s$ y $T_{Off} = 100\ \mu s$.

3.2 Degradación por conmutación

En la degradación por conmutación se requieren frecuencias mas altas, por lo tanto, se realizan solo 2 mediciones por pulso de *gate*. La primera se realiza 40 μs antes que finalice el pulso y la segunda en el momento en el que el pulso de *gate* es 0 (ver figura 3.2). En este caso, se debe cumplir las ecuaciones 3.1 y 3.2 tal que $T_{On} >= 40\ \mu s$ y $T_{Off} >= 22\ \mu s$. Estos 22 μs se deben a que es el tiempo que tardan los conversores en finalizar la conversión.

En este caso, la frecuencia máxima permitida es 16130 Hz con un *duty cycle* exacto de 67.77 %.

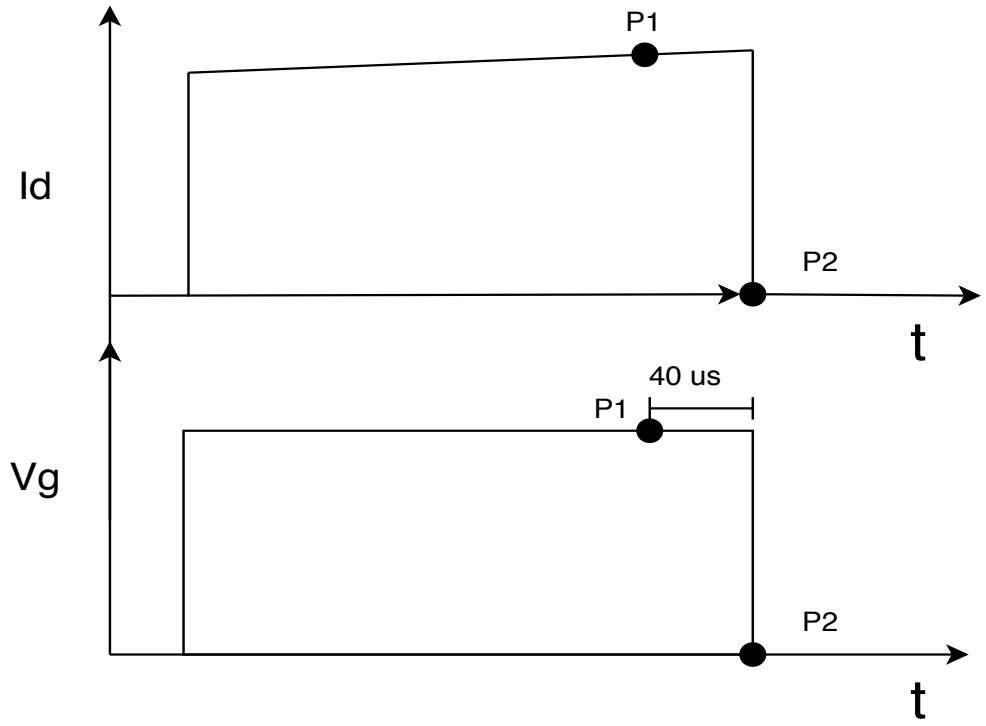


Figura 3.2: Medición de pulsos en 1 ciclo en degradación por conmutación

Un parámetro importante que hubo que analizar es el *slew rate* de la **PGA** del PSoC, ya que nos interesa analizar el comportamiento del MOSFET para el valor deseado de *gate*. Un osciloscopio fue utilizado para medir estos tiempos. Una función cuadrada se aplicó sobre el *gate* y se observó que en 5 μs la tensión ya se establecía en el circuito. Lo mismo ocurre cuando la señal llega a 0 V. Este ensayo se realizó para verificar que el tiempo de respuesta de este dispositivo no fuera una limitante de la frecuencia máxima del sistema.

Se utiliza el imán en este tipo de degradación, ya que, se quiere analizar el pico de avalancha que sucede al generar y cortar un pulso en muy poco tiempo causado por el campo magnético que este genera [6].

3.3 Comunicación entre la PC y el microcontrolador

El software que controla la degradación no opera sobre un sistema operativo de tiempo real (no se puede saber con exactitud el tiempo en el que una instrucción puede ejecutarse). Además, como el PSoC no puede realizar los cálculos del regresor lo suficientemente rápido, así como el cálculo

de la nueva curva de SOA, se debió definir, para no comprometer la degradación, un compromiso importante para este proceso.

El PSoC únicamente **enviará datos a la PC cada 10 ciclos**. En la figura 3.3 puede observarse un ejemplo del tren de pulsos que genera el PSoC (la duración es configurada con el software creado que es detallado en el capítulo 5) y una espera de 500 *ms* por cada 10 ciclos. Como no se utiliza un sistema operativo de tiempo real (ver sección 2.3.8), se midió el tiempo que tardaba el microcontrolador en enviar los datos y recibir una respuesta de la PC. Se usaron todos los núcleos del microprocesador, además de un uso intenso de la memoria RAM y se observó que se tardó 200 *ms* aproximadamente en responder al PSoC. Ya que este tiempo se ve alterado **no solo por el hardware** que la compone si no también de los **procesos actuales** que están ejecutándose, se optó por incrementar el tiempo medido en un 150 %, es decir 500 *ms*, para que el programa pueda ser ejecutado en diversas computadoras.

Para el primer caso, **si no hubiera un límite**, los datos recibidos por la PC podrían llegar tarde y no prevenir la destrucción. También podría suceder que la lectura de datos retrase algún ciclo, por lo que impactaría en los resultados del experimento y la repetitividad ya no se podría lograr.

Para el segundo caso, **si hubieran menos ciclos**, el transistor tardaría demasiado tiempo en degradarse ya que el tiempo que se espera por una respuesta de la PC mitigaría la actividad realizada.

La figura 3.4 muestra la medición de 20 ciclos para mostrar el tiempo entre paquetes, además de la diferencia de corriente medida en cada ciclo. En caso que la PC no alcance a responder, la degradación finaliza porque no es seguro proseguir con el experimento.

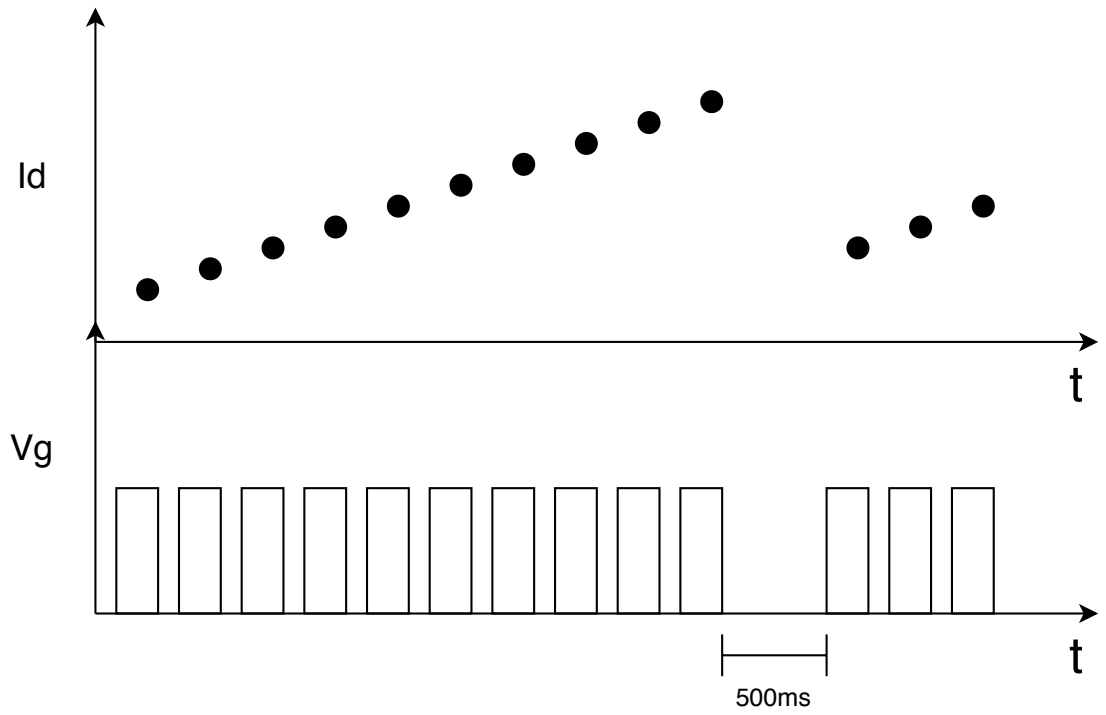


Figura 3.3: Tren de pulsos generados por el PSoC.

Ya que el microcontrolador no obtiene ningún tipo de respuesta de la PC durante esos 10 ciclos, se decidió chequear que la corriente no pase del límite de los 150A y que no se encuentre

fuera del área de operación segura. Para ello, cada vez que la PC analice los datos, enviará al PSoC el límite de corriente para los próximos 10 ciclos. Como la temperatura de juntura puede variar durante esos 10 ciclos, ya que depende de la corriente, tensión de *drain* y temperatura de cápsula, se agregó un *offset* de -5A al límite del *SOA*, para prevenir que el MOSFET se destruya.

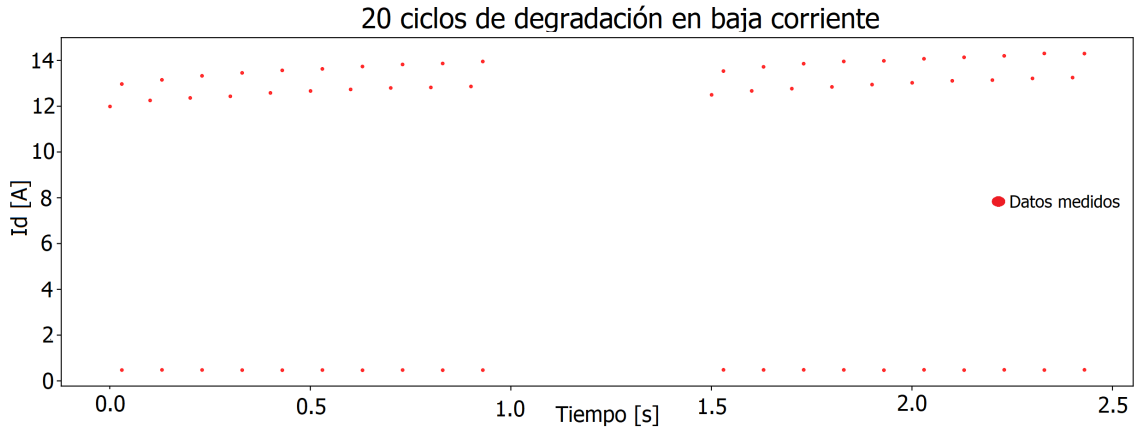


Figura 3.4: Tiempo entre paquetes en degradación por baja corriente

3.4 Condiciones que finalizarían o pausarían la degradación

Se desea llevar el MOSFET a condiciones extremas para acelerar el proceso de degradación, pero se debe cuidar que no se produzca una falla que derive en la rotura del componente. Para eso, a continuación se listan todas las situaciones en la cual se tomarán acciones sobre el flujo de la degradación.

1. Si la corriente sigue circulando por el MOSFET cuando la tensión en el *gate* del transistor sensada por el ADC SAR es $0\,V$:
   a) Desactivación del relé de la fuente.
   b) Encendido del ventilador para enfriar el encapsulado del MOSFET.
   c) Terminación del experimento señalando el error como “Pérdida de control de *gate*”
2. Si la corriente máxima establecida ($150\,A$) o superior es sensada por el sensor de efecto Hall:
   a) Suministro de la señal del *gate* cortada, es decir, se configuran ambos VDAC a $0\,V$.
   b) Desactivación del relé de la fuente.
   c) Encendido del ventilador.
   d) En caso de que la temperatura de la cápsula (T_c) sea mayor a $50\,°C$, se esperará que baje a una temperatura menor a $50\,°C$, en caso contrario, se esperará **5 minutos** para dar tiempo al MOSFET de enfriarse lo suficiente para continuar el experimento desde donde quedó.
3. Si la corriente sensada por el sensor Hall es $0\,A$ y la tensión de *gate* es distinta de $0\,V$:
   a) Suministro de la señal del *gate* cortada, es decir, se configuran ambos VDAC a $0\,V$.

b) Desactivación del relé de la fuente.

c) Encendido del ventilador.

d) El experimento finaliza indicando el error "No hay circulación de corriente".

4. El límite máximo de T_c dado en la hoja de datos del MOSFET es de 175 $°C$, sin embargo, durante la degradación se está disipando potencia y, en consecuencia, generando calor en el interior del componente. Como la T_j máxima también es de 175 $°C$, limitamos la máxima T_c a 100 $°C$:

   a) Suministro de la señal del *gate* cortada, es decir, se configuran ambos VDAC a 0.

   b) Desactivación del relé de la fuente.

   c) Encendido del ventilador.

   d) Esperar que la temperatura de la cápsula baje a 50 $°C$, conectar el relé de la fuente y continuar con el siguiente ciclo de degradación.

5. Si bien la temperatura de juntura máxima es de 175 $°C$ (1.1), en el momento que supere los 160 $°C$, se tomarán las siguientes acciones ya que se quiere evitar que el MOSFET se rompa:

   a) Suministro de la señal del *gate* cortada, es decir, se configuran ambos VDAC a 0 V.

   b) Desactivación del relé de la fuente.

   c) Encendido del ventilador.

   d) Si la temperatura de la cápsula (T_c) supera los 50 $°C$, se espera hasta que esta disminuya a menos de 50 $°C$, caso contrario se esperan 5 minutos y se continúa con el siguiente ciclo de degradación.

6. Se considera que la temperatura máxima de operación del imán es de 100 $°C$, esto es para que la resistencia del imán no varíe en forma considerable. Por lo que se tomaran las siguientes acciones cuando se alcance este límite:

   a) Suministro de la señal del *gate* cortada, es decir, se configuran ambos VDAC a 0 V.

   b) Desactivación del relé de la fuente.

   c) Encendido del ventilador.

   d) Esperar que la temperatura del imán disminuye a menos de 50 $°C$, activar el relé de la fuente y continuar con el siguiente ciclo de degradación.

7. La relación entre la tensión *drain-source* (V_{ds}) y la corriente que circula por el MOSFET (Id) está fuera del área de operación segura (ecuación 1.11):

   a) Suministro de la señal del gate cortada, es decir, se configuran ambos VDAC a 0 V.

   b) Desactivación del relé de la fuente.

   c) Encendido del ventilador.

   d) Espera de 5 minutos a partir de que la temperatura de la cápsula sea menor a 50 $°C$ y se apaga el ventilador.

e) Activación del relé de la fuente.

f) Continuación del siguiente ciclo de degradación.

g) Si en alguno de los siguientes **10 pulsos** el valor cae de nuevo fuera del área, la tensión del *gate* es cortada, se desactiva el relé de la fuente, se enciende el ventilador y se esperan 10 minutos a partir de que la temperatura de la cápsula sea menor a 50 $°C$.

h) Activación del relé de la fuente y apagado del ventilador. Se continúa desde el siguiente ciclo de degradación. En caso de que se repita nuevamente este problema, el error **"Parámetros de degradación fuera del SOA"** es mostrado y se pregunta al usuario si quiere continuar con el experimento, volviendo al paso 1 si el usuario decide continuar y el error vuelve a suceder.

8. La energía disipada en el pulso por pico de avalancha es mayor a 5 J [4] o la temperatura de juntura en el pico de avalancha es mayor a 160 $°C$:

   a) Suministro de la señal del *gate* cortada, es decir, se configuran ambos VDAC a 0 V.

   b) Desactivación del relé de la fuente.

   c) Encendido del ventilador.

   d) Si T_c es mayor a 50 $°C$, esperar que la temperatura de la cápsula baje a 50 $°C$. En caso contrario, esperar 5 minutos. Luego, conectar el relé de la fuente y continuar con el siguiente ciclo de degradación.

Por cada minuto que el experimento se encuentre pausado, se medirá la temperatura. En caso que de que no haya un decrecimiento de 0.5 $°C$ (el bit menos significativo del sensor) en **2 mediciones consecutivas**, se pregunta al usuario si quiere continuar la degradación a partir de ese punto.

La figura 3.5 muestra cómo es el flujo de las degradaciones que se van a realizar en el programa y las acciones descriptas anteriormente.

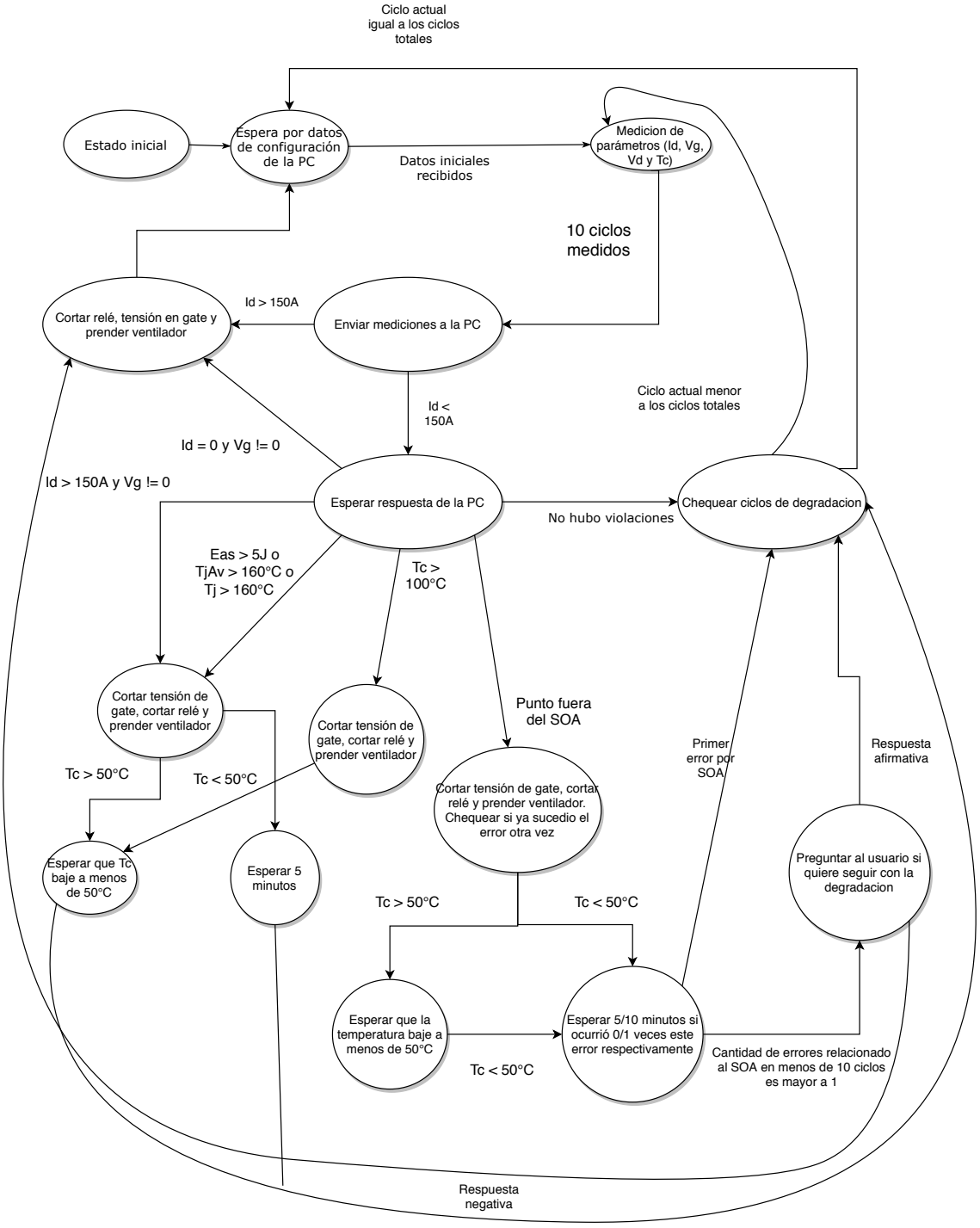


Figura 3.5: Diagrama de flujo del sistema

4. Pronóstico de falla a corto plazo

El microcontrolador de la plataforma de degradación acelerada es el encargado de controlar los parámetros de funcionamiento del MOSFET y finalizar el proceso si durante la operación se alcanza alguno de los límites establecidos. Sin embargo, los límites de corriente y temperatura de juntura varían rápidamente ante un embalamiento térmico, por lo que puede darse el caso en que el dispositivo falle antes que el microcontrolador analice los parámetros y ejecute la acción correctiva. Esto conduce a la necesidad de pronosticar con la mayor certeza posible y en un corto periodo de tiempo si un tren de pulsos de degradación el siguiente ciclo llevará al transistor a trabajar fuera de los límites permitidos, para frenar el proceso con antelación.

Para esto, se emplea un método simplificado de pronóstico a corto plazo mediante regresión local (*Local Regression Based Short Term Forecasting*) en el cual se utilizan los últimos datos históricos de una magnitud para estimar el próximo valor de la misma a partir de la tendencia de los datos. Este método se utiliza para predecir la carga en los grandes sistemas de gestión de energía [20] [21] [22], para predecir el trafico [23], la degradación de dispositivos semiconductores [24], entre otros usos.

A pesar de que existen una amplia variedad de algoritmos para el pronostico a corto plazo, en este trabajo se implemento una versión minimista dado los limitados recursos de computo y, fundamentalmente, el corto periodo de tiempo disponible para el computo. Dado que este proceso debe ser parte del software de la PC, se dispone de 500ms para recibir los datos, pronosticar el próximo punto de operación, verificar que se encuentre dentro de todos los limites del dispositivos y finalmente comunicarle al microcontrolador la decisión de continuar o no con la degradación.

En el caso puntual de este trabajo, es necesario poder pronosticar el próximo valor de la corriente I_d y la temperatura de juntura T_j para poder calcular si el dispositivo se encuentra dentro del área de operación segura. Estos valores se miden para cada pulso de que se aplica en el *gate* del transistor, de esta forma, ambos puede ser analizados como series temporales. A modo de ejemplo, en la figuras 4.1 y 4.2 se grafica en rojo, datos reales de corriente y temperatura, respectivamente, adquiridos durante una degradación.

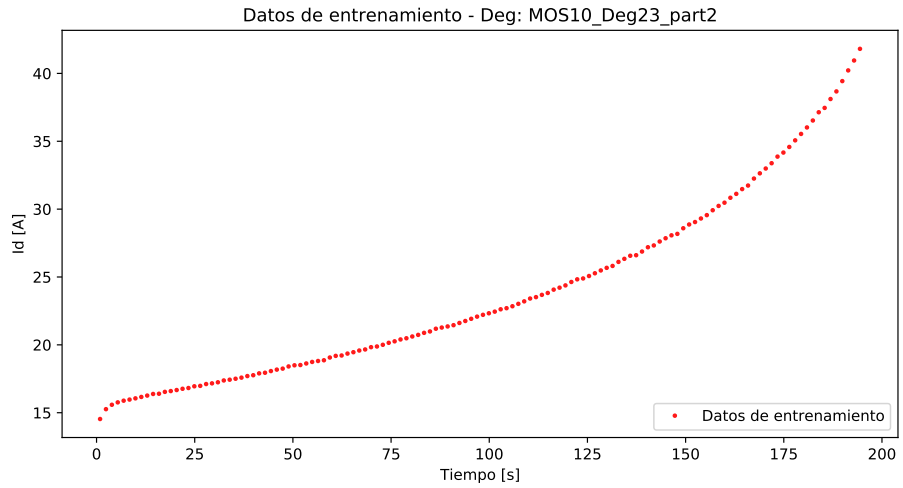


Figura 4.1: Gráfico de la serie temporal de corriente de *drain*

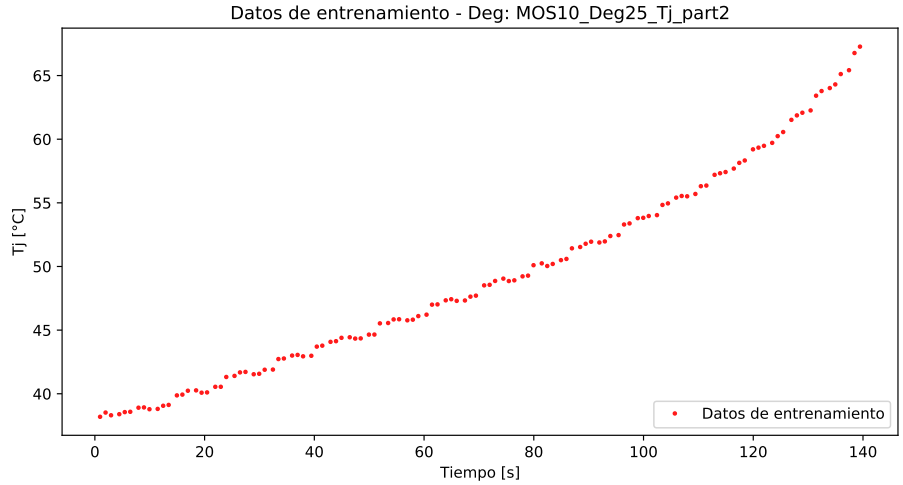


Figura 4.2: Gráfico de la serie temporal de temperatura de juntura

A partir de cada una de las series temporales, es necesario encontrar un modelo que represente individualmente a cada una de las tendencias. Para esto, se utiliza la regresión polinómica, dado que este es un método para encontrar la relación entre variables, más específicamente, la relación que hay entre una variable dependiente y una o más variables independientes. Este modelo se utiliza para pronosticar dichas variables en el próximo ciclo de degradación. Para que esta aproximación sea correcta, es necesario que el punto a pronosticar se encuentre en un periodo cercano de tiempo. En este caso, en el orden de algunos segundos dependiendo del tipo de degradación y la configuración.

En la teoría de los regresores se indica que estos no deben ser utilizados con el objetivo de extrapolar datos, solo para interpolar. Sin embargo, la situación en la que se utiliza el regresor en este caso es particular, se lo emplea para pronosticar en el corto plazo a partir de una tendencia. Es por esto, que solo se utilizan los últimos datos obtenidos para entrenar el regresor y pronosticar el próximo punto de trabajo, esto es lo que se denomina **regresión local**. Al obtenerse un nuevo dato, este es incluido, el mas antiguo se descarta y el regresor es entrenado nuevamente, de esta forma se trabaja con una ventana deslizante de un tamaño determinado para la corriente y la temperatura

de juntura. Estos parámetros se decidieron a partir de pruebas experimentales con datos de degradaciones reales. Cuando un error interrumpe la degradación, se **reinicia el entrenamiento** de cada regresor.

4.1 Regresión lineal

Un modelo básico lineal puede verse en la ecuación 4.1, donde Y_i representa la variable dependiente, X_i representa la variable independiente y ε_i representa el residuo. a y b es lo que llamaremos, a partir de ahora, **coeficiente de regresión**.

$$Y_i = X_i a + b + \varepsilon_i \tag{4.1}$$

Existen una gran cantidad de modelos con distintas cantidades de coeficientes. La elección de uno u otro dependerá de la naturaleza de los datos, el poder de cómputo disponible, tiempo de ejecución, precisión necesaria, etc. Para calcular estos coeficientes, se utiliza el **método de los mínimos cuadrados** que asegura la minimización de la suma cuadrada de los residuos (la diferencia del valor observado con el predicho por el modelo), es decir, se debe encontrar los coeficientes a, b tal que minimicen la ecuación 4.2.

$$S = \sum_{i=1}^{n} {\varepsilon_i}^2 \tag{4.2}$$

La métrica que frecuentemente se utiliza para comprobar el comportamiento del regresor, es mediante el uso del **coeficiente de determinación** o *R-Squared* 4.3

$$R^2 = 1 - \frac{S}{S_{obs}} \tag{4.3}$$

$$S_{obs} = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{4.4}$$

donde y_i representa el punto medido e $\bar{y}$ es el promedio de los puntos medidos.

Este coeficiente indica que cuanto más cerca de 1 se encuentre, habrá una mayor relación entre los datos, mientras que cuanto sea menor a 1, no mostrarán ninguna relación entre ellos. Para poder predecir correctamente un valor, se esperaría que el *R-Squared* esté próximo a 1. Esta métrica se utiliza para entrenar el regresor.

4.2 Regresión polinomial

En la regresión polinomial se modela la relación de las variables por el uso de polinomios del grado deseado. Siguiendo la idea de la regresión lineal, el sistema de ecuaciones para obtener cada parámetro del modelo mediante el uso del método de mínimos cuadrados puede verse en la ecuación 4.5.

$$\begin{bmatrix} m & \sum x & \sum x^2 & \sum x^3 & \dots \\ \sum x & \sum x^2 & \sum x^3 & \sum x^4 & \dots \\ \sum x^2 & \sum x^3 & \sum x^4 & \sum x^5 & \dots \\ \sum x^3 & \sum x^4 & \sum x^5 & \sum x^6 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} * \begin{bmatrix} a \\ b \\ c \\ d \\ \dots \end{bmatrix} = \begin{bmatrix} \sum y \\ \sum xy \\ \sum x^2 y \\ \sum x^3 y \\ \dots \end{bmatrix} \tag{4.5}$$

donde m es la cantidad de datos, x la variable independiente e y la variable dependiente. Notar que si bien la cantidad de parámetros es $n+1$ (donde n es el grado del polinomio), la regresión sigue siendo lineal ya que la variable dependiente y sigue estando en función de la variable independiente x.

4.3 Implementación del regresor polinómico

La regresión polinómica es ampliamente utilizada y ya muchos paquetes científicos incluyen implementaciones optimizados y versátiles. Dos de ellos son el de la biblioteca **GSL**[25] y el de **Numpy** [26]. Para determinar que paquetes utilizar, se compara la precisión y el tiempo de computo.

4.3.1. Comparación de tiempo de computo

El objeto de estos experimentos es comparar el tiempo de computo de ambos algoritmos, sin analizarse la precisión. Para esto, se generaron series temporales con polinomios de distintos grados y se evalúan en una gran cantidad de puntos.

Como primer experimento, se eligió un polinomio de segundo grado: $y = 3x^2 + x - 5$, y se lo evaluó en 500 puntos aleatorios entre el rango $[-5, 5]$. Estos datos generados son considerados una serie temporal y se aproximaron dos polinomios de grado 2, uno cada paquete. El resultado puede verse en la figura 4.3.

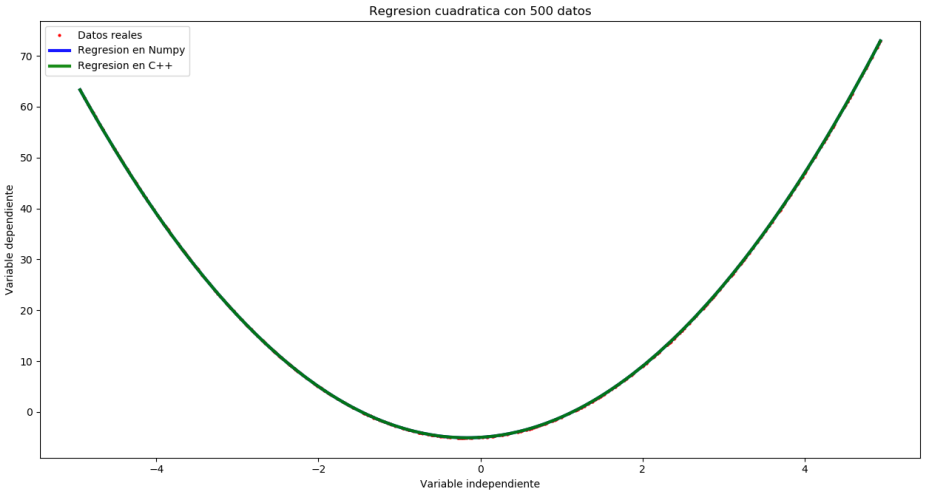


Figura 4.3: 500 datos en el rango [-5, 5] con polinomio de grado 2 para aproximar $y = 3x^2 + x - 5$

En la figura 4.4 se puede ver un experimento similar pero en este caso con un polinomio de tercer grado ($y = x^3 + 2x^2 - 1x + 3$). Este fue evaluado en 800 puntos y los cuales se utilizaron para entrenar un regresor de tercer orden.

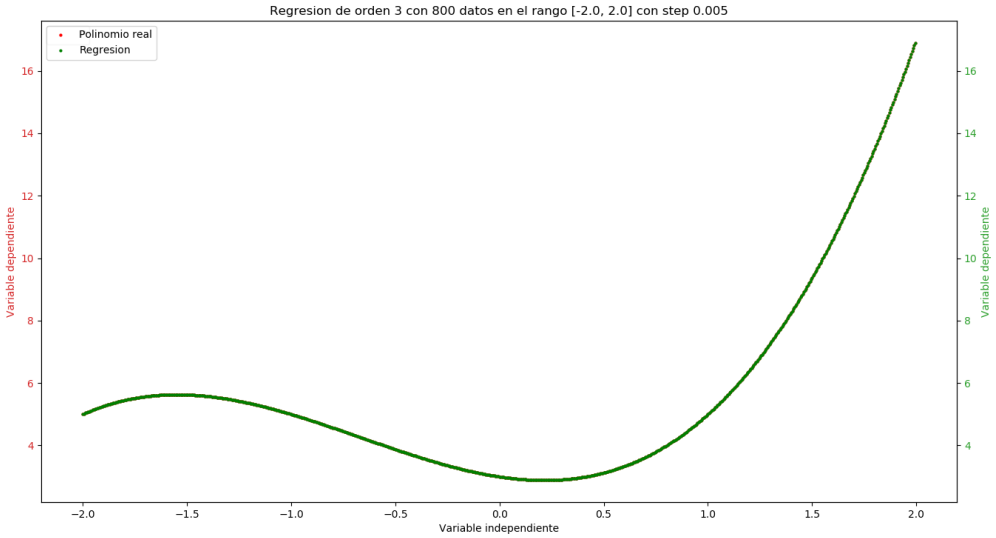


Figura 4.4: Regresor de grado 3

Finalmente, en la figura 4.5 se le agregó un ruido del 10 % a los datos del polinomio anterior, con el objeto de analizar el cambio de tiempo en situaciones no ideales.

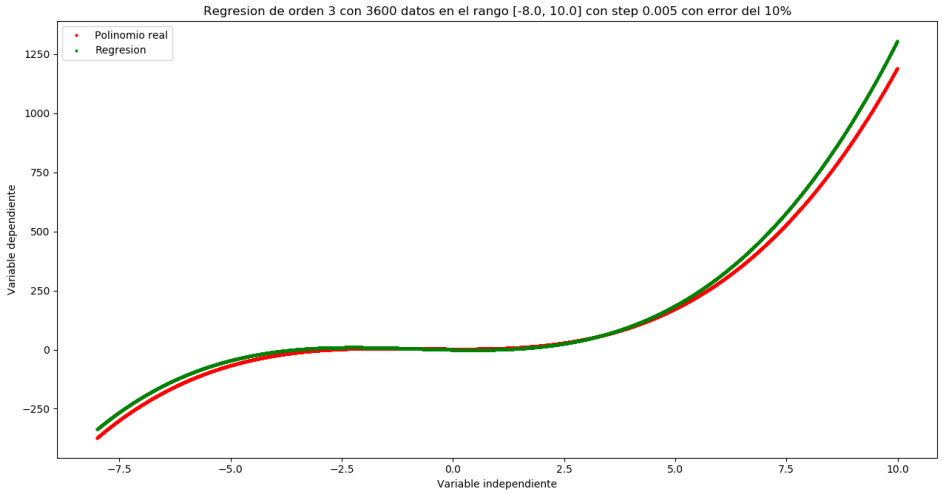


Figura 4.5: Regresor de grado 3 con error del 10 % en los datos de entrenamiento

Como puede verse en la tabla 4.1, los tiempos de computo del paquete GSL son muy inferiores a los tiempos del regresor del paquete Numpy. A esto se le suma que el proyecto se esta realizando en lenguaje C++ lo cual facilita la utilización del paquete GSL. Por estas dos razones, se decide utilizar el regresor del paquete GSL.

figura	Tiempo de ejecución en C++	Tiempo de ejecución en Python
4.3	0.2529 ms	20.53 ms
4.4	0.0395 ms	3.21 ms
4.5	0.0467 ms	4 ms

Tabla 4.1: Tiempos asociados a la ejecución del algoritmo para cada figura

4.4 Filtrado de datos para el entrenamiento

Como se mencionó al comienzo de este capitulo, se necesita una regresión local para analizar la tendencia de la corriente de *drain* y de la temperatura de juntura, para a partir de esta, pronosticar si durante el próximo ciclo de degradación el transistor se encontrará dentro del área de operación segura.

Sin embargo, como se explica en la sección 3.3, por cada pulso de gate se realizan dos o tres mediciones (dependiendo el tipo de degradación) y luego se envían a la PC los datos correspondientes a 10 pulsos de *gate*, es decir, 20 o 30 datos. Esto debe realizarse así debido a las limitaciones tecnológicas del dispositivo utilizado. Al finalizar la transmisión de los datos, se analiza si existe alguna violación de los límites de operación del dispositivo. En caso en que no se registre ningún problema, la PC le da la orden al microcontrolador de continuar, dado que la PC **no puede detener la degradación en medio de estos 10 ciclos** de degradación, es necesario pronosticar el peor caso.

En el gráfico de la figura 4.6, en rojo, pueden verse algunos datos recogidos en una degradación real. Pueden verse claramente los grupos de datos que corresponden al mismo paquete de transmisión. Debido a la forma, es evidente que un regresor polinómico no podría ajustar en forma adecuada para todos los puntos medidos. A esto se le suma que no es útil una curva que se ajuste por completo a los datos de cada ciclo, solo es necesario poder pronosticar el **máximo** de dicho conjunto de datos. La figura 4.7 es similar a la anterior pero en ella se resalta el máximo de cada grupo (en negro). Estos puntos son los que se utilizan para **entrenar el regresor**, si se aíslan estos datos en un único gráfico, se obtiene uno similar al mostrado en la figura 4.1 al comienzo del capítulo.

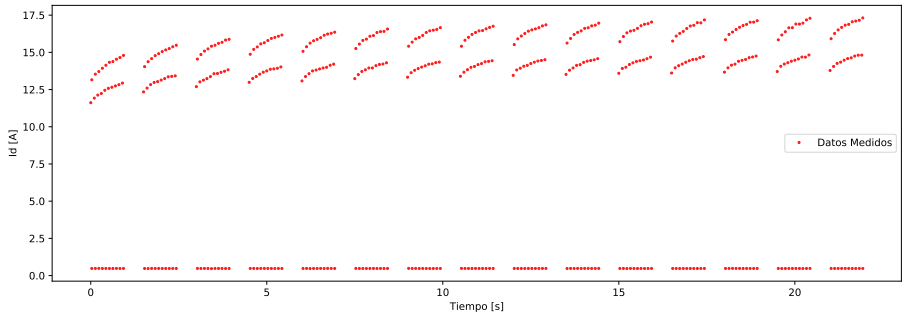


Figura 4.6: Datos adquiridos durante una degradación

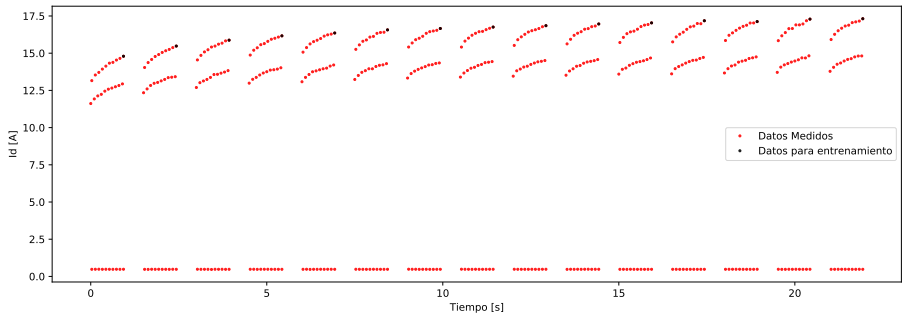


Figura 4.7: Datos adquiridos durante una degradación con el máximo resaltado

4.5 Análisis del tamaño de ventana deslizante

Ya definido el procedimiento para el cual se van a seleccionar los puntos de entrenamiento para el regresor, es necesario encontrar el tamaño de la ventana deslizante, mediante la cual, se eliminan puntos antiguos y se vuelve a entrenar el regresor. Para ello, se eligieron 2 métodos de medición del error de pronóstico que se detallan en las secciones 4.5.1 y 4.5.2.

4.5.1. Error porcentual absoluto medio

El error porcentual absoluto medio (MAPE, por las siglas en ingles) que mide el tamaño del error absoluto en términos de porcentaje. Este se obtiene con la ecuación 4.6

$$MAPE = 100 * \frac{\sum_{t=1}^{n} \mid \frac{A_t - F_t}{A_t} \mid}{n} \tag{4.6}$$

donde n es la cantidad de puntos obtenidos, A_t el valor real y F_t el valor pronosticado por el regresor en nuestro caso

4.5.2. Error cuadrático medio

El error cuadrático medio (ECM), se obtiene con la ecuación 4.7

$$ECM = \frac{1}{n} \sum_{t=1}^{n} \mid Y_t - \hat{Y}_t \mid \tag{4.7}$$

donde n es la cantidad de puntos obtenidos, Y_t el valor real y $\hat{Y}_t$ el valor pronosticado por el regresor.

4.5.3. Selección de tamaño de ventana deslizante

Ya definidos estas 2 métricas, se obtuvieron los datos de varias degradaciones en diversas condiciones y se promediaron los errores para cada tamaño de ventana. Las figuras 4.8 y 4.9 muestran los errores asociados para corriente y temperatura de juntura correspondiente de una degradación basado en el tamaño de la ventana deslizante.

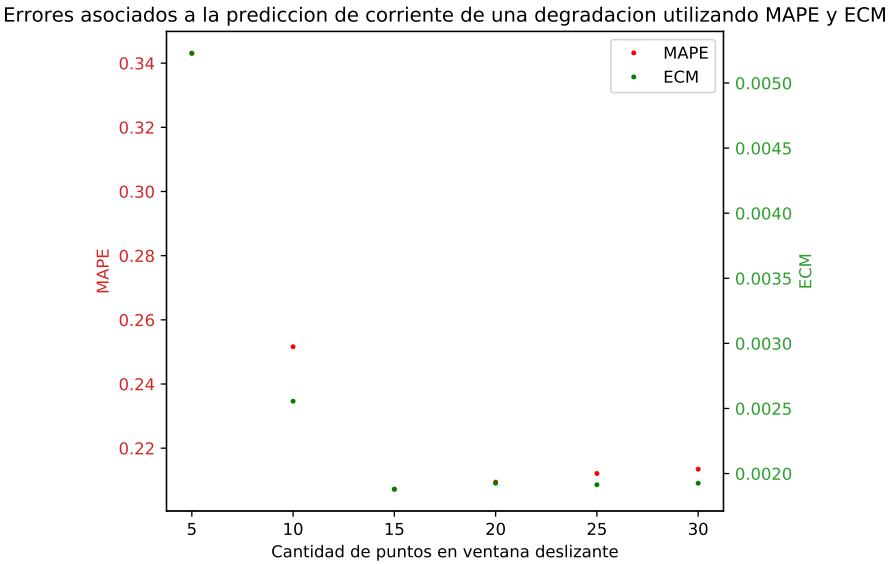


Figura 4.8: MAPE y ECM del regresor de corriente sobre una degradación

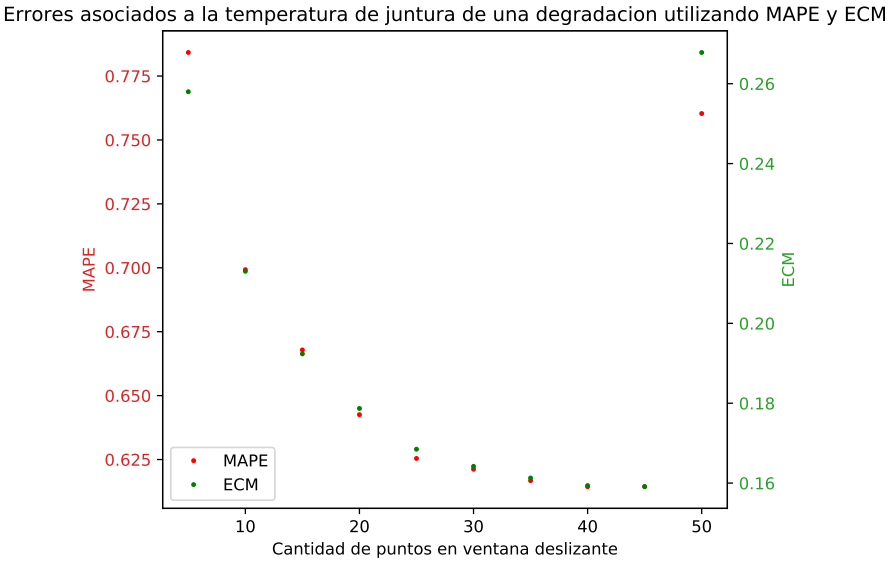


Figura 4.9: MAPE y ECM del regresor de temperatura de juntura sobre una degradación

A partir de estos graficos (figuras 4.8 y 4.9), se determino que para el regresor de corriente de *drain* el tamaño de la ventana óptimo es de **15 puntos**, mientras que, en el de temperatura de juntura se utilizará una ventana de **40 puntos**. Cabe destacar que, utilizando esta configuración, en ambos casos el error de **MAPE** es menor al 1 %, por lo que el regresor se encuentra muy cerca del valor real para el caso promedio.

5. Software de degradación

Para facilitar la configuración del proceso de degradación y visualizar los datos adquiridos por el PSoC, se creó un programa con interfaz gráfica. Dicho programa permite la visualización de todos los parámetros medidos, detección de condiciones que degraden al MOSFET, almacenamiento en base de datos local, etc. En las siguientes secciones se detallará el uso de las herramientas de software así como el programa creado.

5.1 C++

C++ es un lenguaje de programación compilado creado por Bjarne Stroustrup en el año 1979, con la idea de proveer las facilidades de **Simula 67** junto a la flexibilidad y performance de C. La gran diferencia que tiene con este, es la manipulación de objetos y, aunque, agregado en años posteriores, la posibilidad de realizar metaprogramación.

Actualmente, C++ es considerado uno de los lenguajes más óptimos en cuanto a performance, ya sea, por los compiladores que cada vez son más potentes y utilizan mejores algoritmos y por dar libertad total al usuario. Esta flexibilidad es la que lo hace diferente al resto. C++ sigue expandiéndose con nuevas características, librerías propias y mejoras en el rendimiento. Los compiladores más utilizados son:

- GCC
- MSVC (Windows)
- Clang

Cabe destacar que no todas las características definidas por el estándar son implementadas por los compiladores, e inclusive algunas de ellas son implementadas por uno pero no por otro.

5.1.1. Aplicaciones actuales

Como se mencionó anteriormente, C++ es uno de los lenguajes con mayor flexibilidad y rendimiento. Se encuentran grandes proyectos desarrollados principalmente con este lenguaje, porque requieren tener un tiempo de respuesta rápido ya que involucran mucha interacción con el usuario. Algunos de ellos son:

- Unreal Engine, un motor gráfico 3D para videojuegos y renderizados escrito puramente en C++
- Adobe, la compañía desarrolladora de herramientas para diseño digital.
- Autodesk, en particular, herramientas del tipo AutoCAD.
- Doxygen, herramienta para generar documentación a partir del código fuente.
- Microsoft, no solo en su sistema operativo Windows, si no también sus propias aplicaciones, como el pack Office.

- Mozilla Firefox, el navegador de internet open source.

5.1.2. Motivos de uso para el presente trabajo

Se eligió este lenguaje ya que se necesita de tiempos de respuesta rápidos debido a que los componentes del experimento son costosos y se necesita una respuesta rápida a cualquier inconveniente.

5.2 Qt

Qt es un framework open source escrito en C++ para desarrollar aplicaciones de escritorio multiplataforma. Fue creado por Haavard Nord y Eirik Chambe-Eng, en Noruega, lanzada aproximadamente en 1995. Sigue en constante desarrollo, por lo que hay actualizaciones seguidas para expandir las características de este. Se puede notar como una gran cantidad de aplicaciones GUI utiliza Qt. Esto es fácil de ver por las librerías dinámicas que estos utilizan. Alguno de los programas que usan Qt son:

- Maya: Programa de modelado 3D, animación, renderizado, entre otras características, hecho por Autodesk.
- Dolphin: Emulador de las consolas Nintendo Gamecube y Nintendo Wii.
- Wolfram Mathematica: Una herramienta enfocada al área científica, matemática y de computación.

5.2.1. Motivos de uso para el presente trabajo

Se decidió por usar este framework sobre C++ por sus grandes librerías para facilitar, no solo la creación de la interfaz y comunicación con base de datos, si no, por ofrecer una simple y potente librería de comunicación con puertos serie, necesario para comunicarse con el microcontrolador que describiremos más adelante.

Para poder obtener los datos que el PSoC obtiene de la etapa de potencia, se diseñó un software capaz de configurar los parámetros iniciales para la degradación; medir y almacenar los parámetros de interés; reaccionar a futuras violaciones mediante el uso del **regresor** y la verificación de cualquier posible error durante los ciclos medidos; visualizar hasta 16 curvas de los parámetros medidos; exportar curvas en formato **PDF** y **PNG**.

En las siguientes secciones, se muestra y detalla el funcionamiento del programa.

5.3 Ventana principal

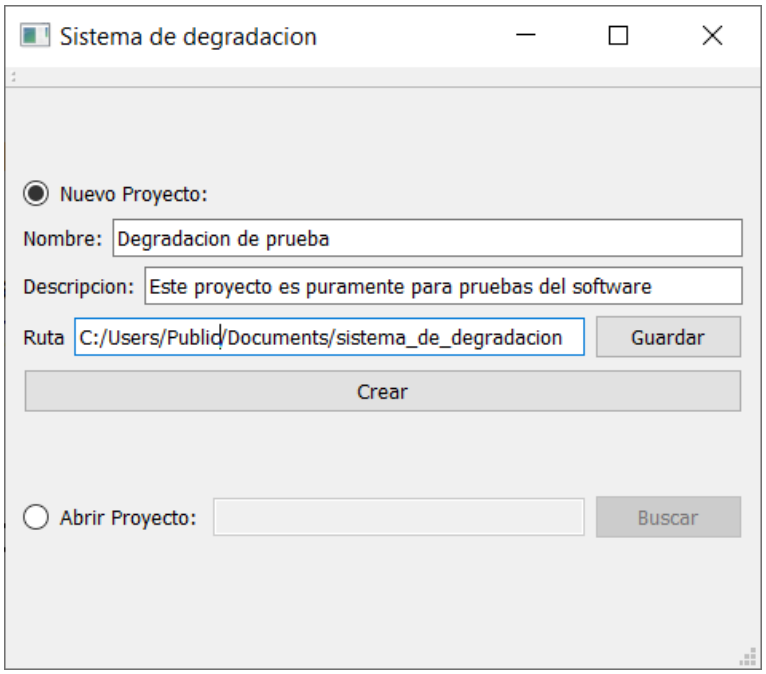


Figura 5.1: Ventana principal del programa.

Al ejecutar el programa, se abrirá la ventana para crear un nuevo proyecto o cargar uno anterior (ver Figura 5.1). Un proyecto es simplemente un archivo referenciando la ruta de la base de datos creada para ese únicamente.

5.4 Análisis

En esta pestaña (ver Figura 5.2), se podrá observar en forma de árbol todas las degradaciones realizadas a todos los MOSFETS de este proyecto. Cabe destacar que el software automáticamente agregará el sufijo **"_Nuevo"** cuando un nuevo MOSFET es degradado. Para continuas degradaciones del mismo MOSFET, se agregará el sufijo **"_Deg#"**, donde # será reemplazado por el número de la degradación actual. El software **no deja la posibilidad de editar este número manualmente** debido a que se asume que las degradaciones se harán con éste únicamente. Se pueden seleccionar las degradaciones deseadas para cargarlas y graficar la respuesta temporal de los parámetros medidos durante dichas degradaciones en el visualizador.

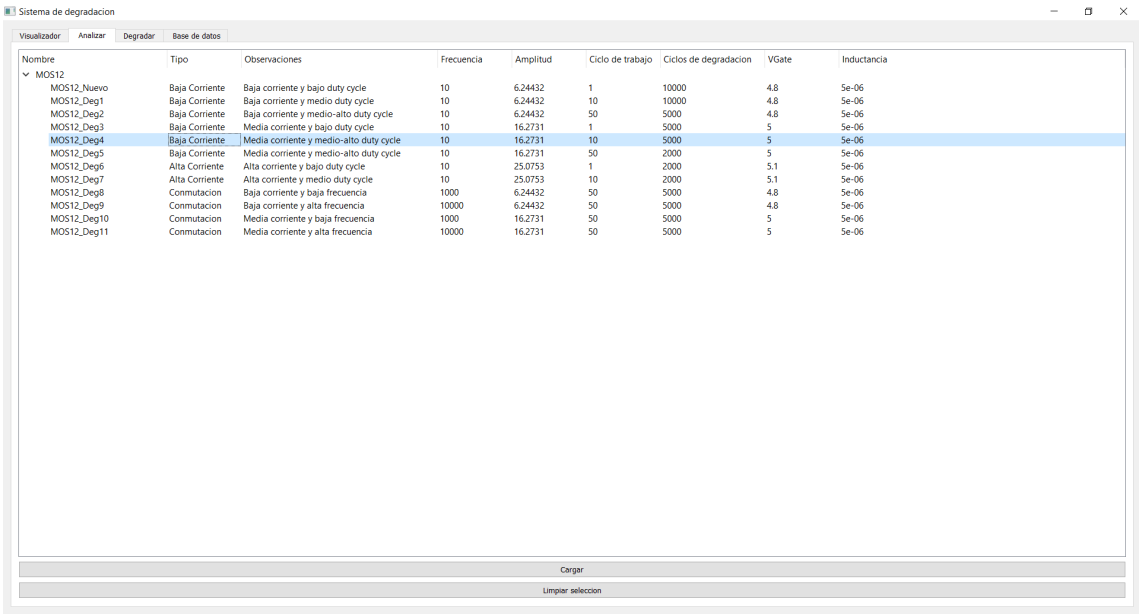


Figura 5.2: Ventana de análisis del software.

5.5 Visualizador de curvas

El visualizador de curvas permite graficar todos los parámetros deseados de las degradaciones cargadas previamente en la pestaña de Análisis, mostrar los errores que hubo en ellas, exportar el gráfico en formato **PDF** o **PNG** y exportar los datos en **CSV** para poder ser importado en otro programa si es deseado.

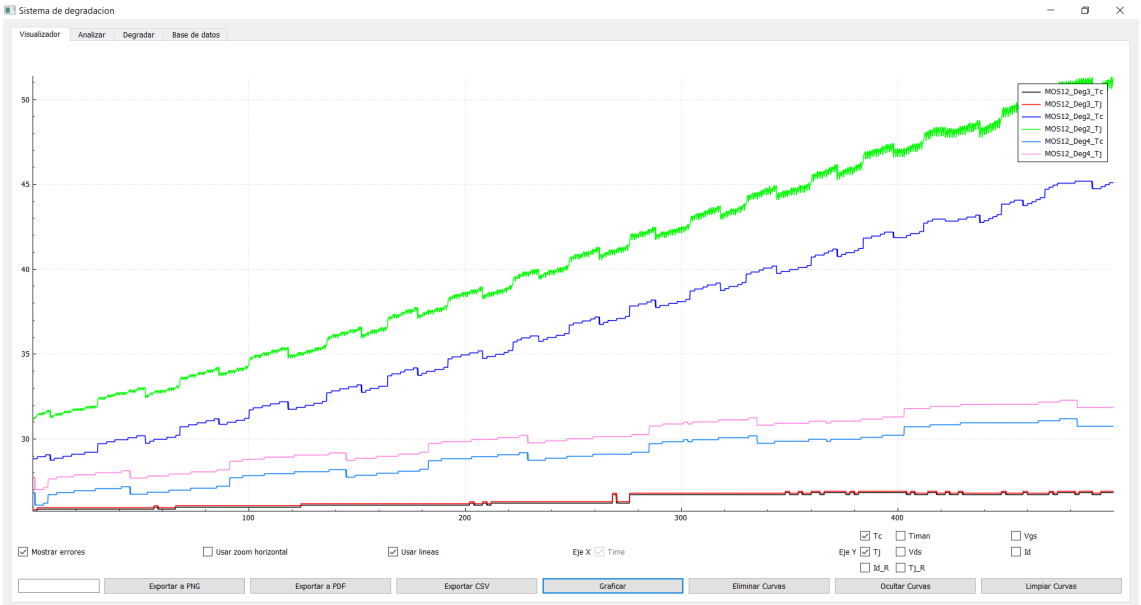


Figura 5.3: Ventana de visualización de datos.

5.6 Degradación

En la pestaña de degradación (ver Figura 5.4), se pueden configurar el tipo de degradación y los parámetros necesarios para empezar el experimento. Ellos son (por orden de aparición):

1. Frecuencia del pulso de *gate* en Hz (ver sección 3.2 y 3.1 para los valores máximos y míni-

mos permitidos).

2. Ciclo de trabajo, con un mínimo de 0 % y un máximo de 100 %, aunque el valor depende de la frecuencia especificada (ver sección 3.2 y 3.1 para entender la relación).

3. Inductancia del imán que es utilizado para la degradación de conmutación. Esto permite cambiar el imán si se desea y que el software realice correctamente los cálculos necesarios.

4. Cantidad de ciclos de degradación.

5. Tensión de *gate* que la PSoC debe brindar al circuito. Al modificar este valor, el software estima la corriente que circulará por el *drain* del transistor.

6. Las observaciones, si se desea aclarar algo antes de comenzar la degradación.

7. La resistencia térmica entre la juntura y la cápsula (R_{th}), por si otro tipo de MOSFET es utilizado en el experimento se deja este parámetro configurable.

Cabe resaltar que el software calcula automáticamente la corriente que va a haber en el circuito basada en la tensión de *gate*.

Se puede elegir por degradar un MOSFET nuevo o elegir uno de los previamente degradados. Nuevamente, cabe destacar que el software automáticamente asignará un sufijo al nombre para cada degradación del MOSFET elegido.

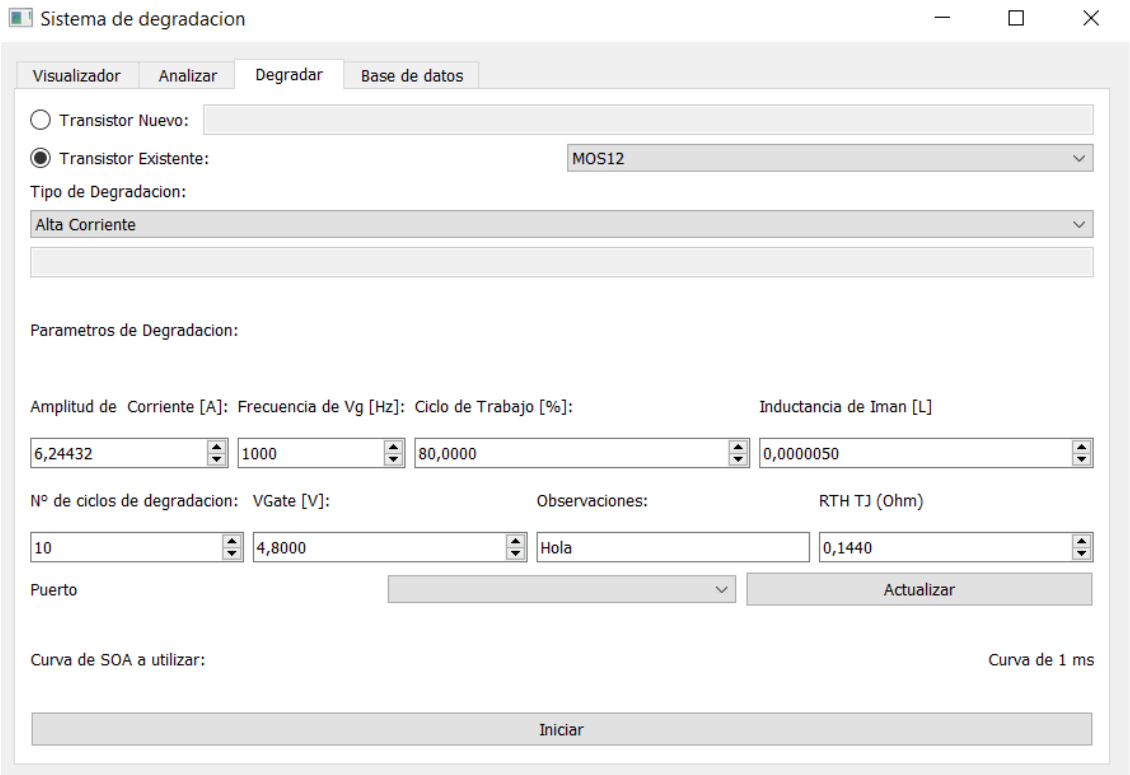


Figura 5.4: Ventana de degradación de MOSFETs.

Lo más importante a destacar es la selección del puerto para comunicarse y que los ciclos de degradación deben ser múltiplo de 10, esto último explicado en la sección 3.3. Con respecto a los

puertos para comunicarse, el programa detectará inicialmente si hay algún puerto serie conectado. En caso de que no haya, es dejado al usuario que se asegure de conectar el dispositivo y actualice la lista con el botón “Actualizar”.

5.6.1. Selección de funciones para calcular puntos de SOA

Antes de iniciar una degradación, se brinda al usuario la posibilidad de cambiar todas las funciones que se utilizarán para calcular los puntos que se encuentren fuera del área de operación segura. Se asume que las funciones son en escala logarítmica. Esto es útil por si se desea utilizar otro MOSFET donde las curvas sean completamente diferentes a las del transistor que se utilizó en este experimento.

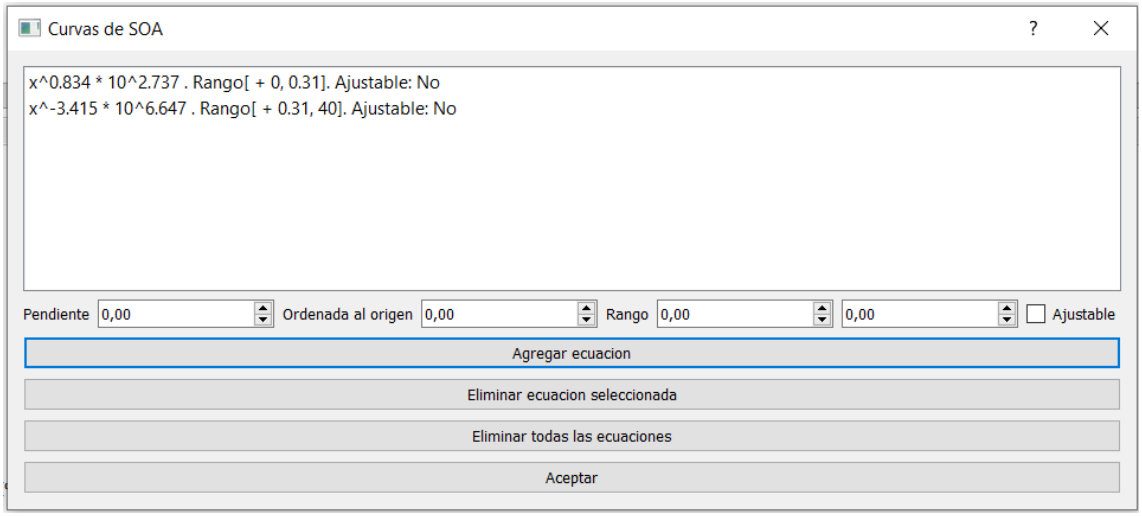


Figura 5.5: Especificación de curvas para calcular puntos de SOA

5.6.2. Visualización de parámetros mientras se realiza la degradación.

Al empezar la degradación, se observarán 2 gráficos que están asociados a las curvas de corriente y temperatura de juntura de los datos medidos y los calculados por el regresor (ver figura 5.6). En la parte inferior de la interfaz, se muestra el estado actual de la degradación, como por ejemplo, cuales son los siguientes pulsos, si hubo algún error y qué tipo de error con el valor asociado si corresponde, que acciones se tomarán al respecto, entre otros mensajes. Se puede elegir el grado del regresor a utilizar, que por defecto es 2, y también existe la opción de graficar o no los puntos de la degradación para consumir menos recursos de la PC si no es necesario. Además, se muestra el regresor para cada una de las variables en forma de polinomio. En cualquier momento se puede parar la degradación.

5.7 Base de datos

En caso de ser necesario ver los valores de las degradaciones, esta pestaña permite hacerlo mediante una tabla (ver Figura 5.7). Basta seleccionar la tabla deseada y los datos acordes son mostrados. También pueden borrarse tablas individualmente, pero no es recomendable ya que puede alterar el orden de las degradaciones ya que las tablas creadas no pueden ser reemplazadas.

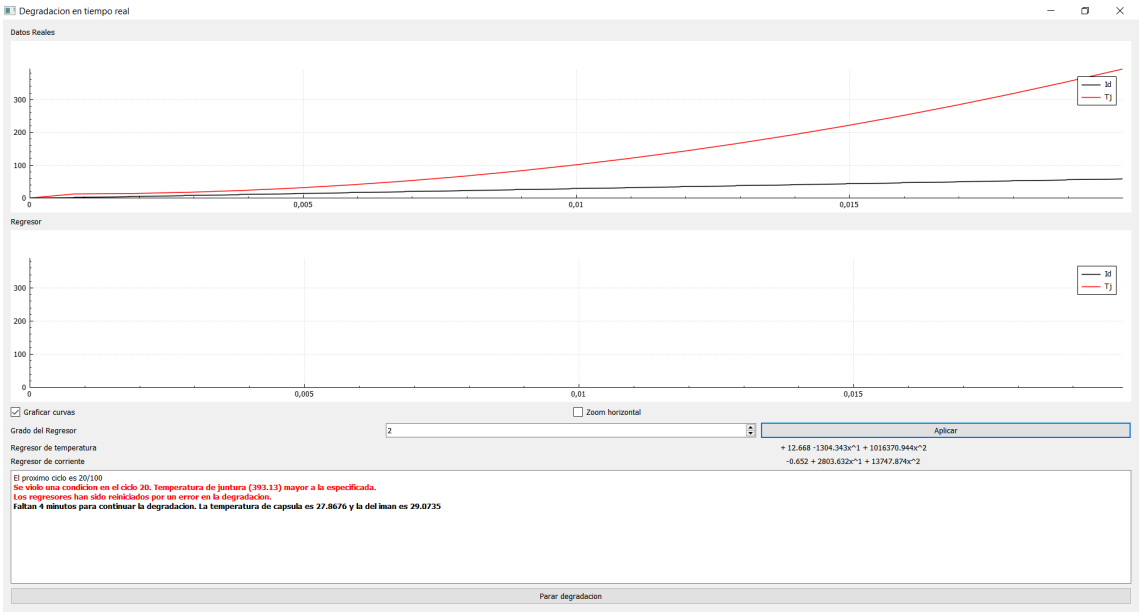


Figura 5.6: Ventana de degradación en tiempo real de un MOSFET.

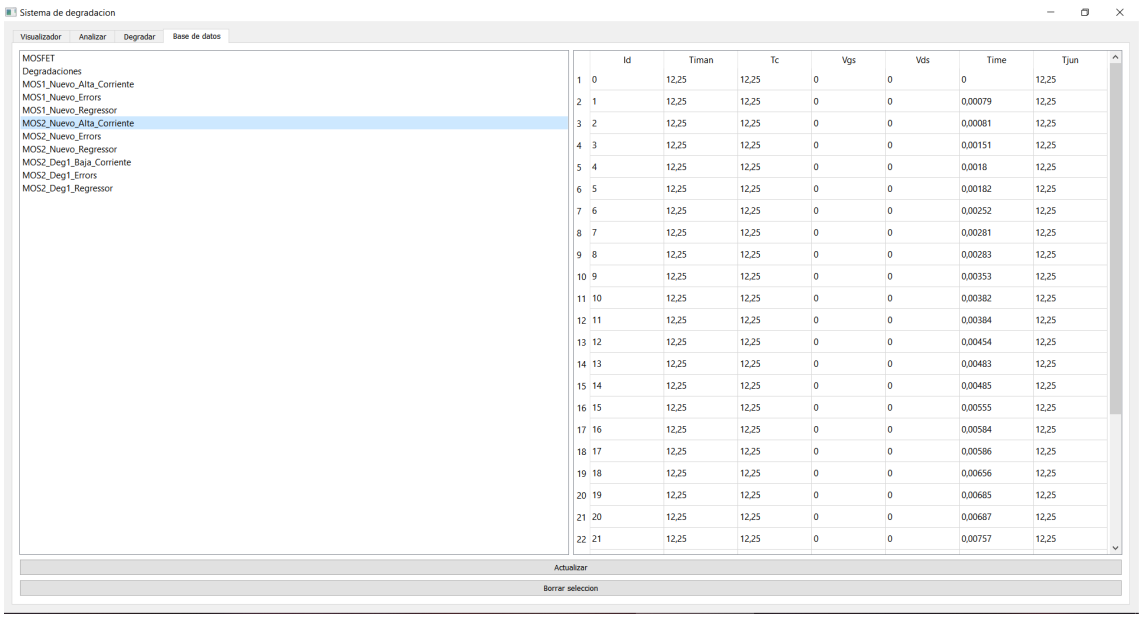


Figura 5.7: Ventana de base de datos.

5.8 Flujo de la degradación

Una vez que se han seleccionado los datos para la degradación, la PC los envía mediante USB al PSoC y éste comienza el experimento especificado por los parámetros elegidos por el usuario.

5.8.1. Degradación en continua por alta/baja corriente

A continuación se explicarán los pasos que se realizan en la comunicación del software de la PC y el PSoC con la degradación de alta/baja corriente.

1. Recibido los datos en el PSoC, éste envía la temperatura inicial del sensor que mide la temperatura de la cápsula. Una vez enviado, espera que la PC le envíe la confirmación de

que está lista para empezar a recibir datos.

2. Una vez recibida la confirmación, se inicia un bucle de hasta la cantidad de ciclos de degradación especificados dividido 10, donde por cada iteración envía 10 paquetes de mediciones por lo mencionado anteriormente en la sección 3.3.

3. Se procede a configurar ambos VDAC que se utilizan en el PGA con el valor enviado desde la PC. Se esperan $500\mu s$ para que la corriente circulante se estabilice.

4. Se escribe un 1 en el registro de control de los conversores, para que la conversión síncrona comience. Se espera que el conversor *Delta Sigma* finalice ya que es el conversor más lento (10 microsegundos más que los conversores SAR, es decir, 22 microsegundos en total) y una vez finalizada la medición, se escribe un 0 en el registro de control.

5. El PSoC espera hasta que queden $100\mu s$ antes de que el pulso finalice, se repite el paso 4 y se corta la tensión de *gate* una vez obtenido los datos.

6. Después de haber cortado el pulso de *gate*, se esperan 100 μs y se repite el paso 4.

7. Se mide en el microcontrolador el valor de la corriente y si ésta viola el límite establecido (150A), la llave del relé se abre y se prende el ventilador. Si quedaban ciclos restantes de los 10 que se envían a la PC, se clona el último ciclo para rellenar los restantes, se envían a la PC y se finaliza la degradación desde el PSoC. Como la PC recibirá los datos del PSoC, ésta identifica el mismo problema y notificará al usuario que la degradación finalizó por la violación de corriente.

8. En caso de que no se haya violado el límite de la corriente, el PSoC espera el tiempo restante del periodo y comienza el próximo ciclo del paquete para volver al paso 3.

9. Una vez medidos los 10 ciclos, estos se enviarán a la PC mediante USB; Si era el último paquete de ciclos se termina la degradación, en caso contrario, se iniciará el contador de espera a 500 milisegundos que la PC tendrá para hacer todo el análisis necesario.

10. La PC al recibir los datos de la degradación, los almacena en memoria y procede a chequear si alguno de los límites fue violado. En caso afirmativo, envía el código correspondiente al PSoC, reinicia el regresor y los gráficos correspondientes si el usuario habilitó la opción. Lo mismo sucede si el regresor detecta una falla en los próximos 10 ciclos. En caso negativo, la PC envía el código apropiado a la ausencia de error, además de actualizar el límite de corriente para que no viole el área de operación segura. Cuando el error ocurra en los últimos 10 ciclos, se almacena en memoria dicho error pero no se informa al PSoC porque éste ya finalizó.

11. Al terminar el contador de espera, si el PSoC no recibe respuesta de la PC, termina la degradación. En caso de recibir datos, verifica el código recibido. Si no hay error, espera 90 milisegundos adicionales para comenzar con el próximo paquete y volver al paso 3. Si hay error, toma la acción correspondiente y continúa volviendo al paso 3 en caso que no sea un error terminal.

5.8.2. Degradación por conmutación

La degradación por conmutación es muy similar a la anterior. Las principales diferencias, además del uso del imán, son la cantidad de mediciones por pulso (2 en lugar de 3), los tiempos de medición (40 μs antes de finalizar el pulso e inmediatamente cuando la tensión de los VDAC es 0) y el análisis de los límites del efecto avalancha y la temperatura de juntura en avalancha.

6. Resultados

Se realizaron pruebas sobre los límites mencionados anteriormente para verificar el funcionamiento correcto del software.

Para probar los límites de temperatura de cápsula e imán, se utilizó una pistola de calor para elevarlos hasta la temperatura correspondiente.

Para el caso del sensor de temperatura de la cápsula (figura 6.1), puede verse como evolucionó la temperatura respecto al tiempo al calentar el sensor con una pistola de calor. Puede verse que pasados los 40 segundos, se superó el umbral de 100 grados centígrados, provocando que el sistema detenga la degradación. A partir de este momento, el software verifica la temperatura cada 1 minuto para determinar si puede continuar con la degradación. Con el objeto de analizar si el software funciona correctamente, se dejó de calentar el sensor y luego de 60 segundos, el sistema continuó con la degradación, ya que la temperatura es menor de 50°C como puede verse en la parte inferior derecha del gráfico. Algo similar ocurrió con el sensor de temperatura del imán (figura 6.2).

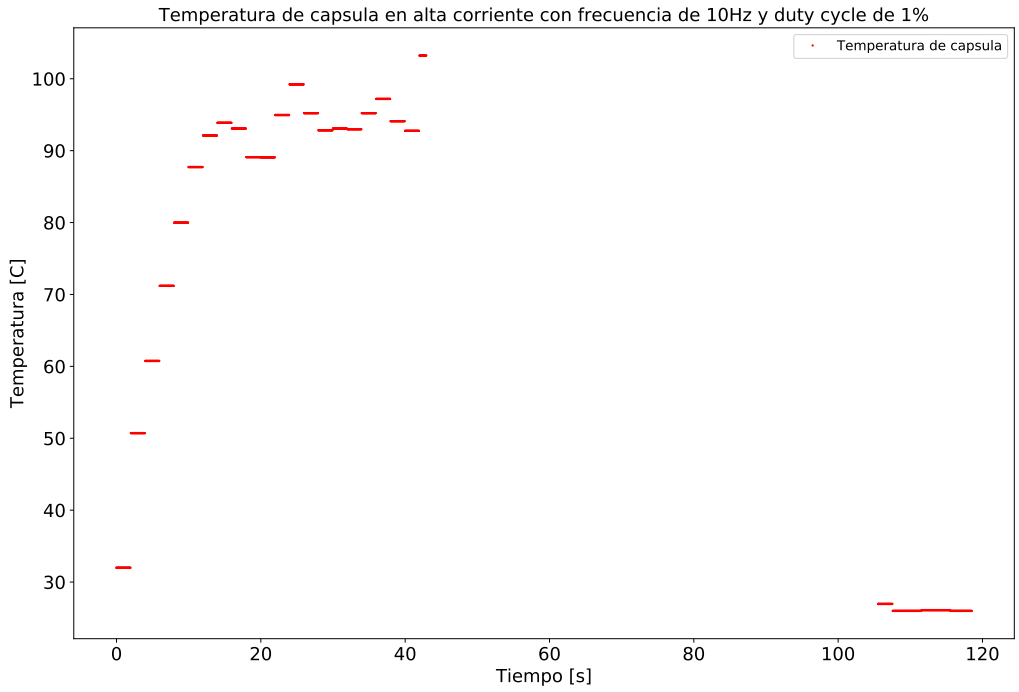


Figura 6.1: Temperatura de cápsula de un experimento a lo largo del tiempo.

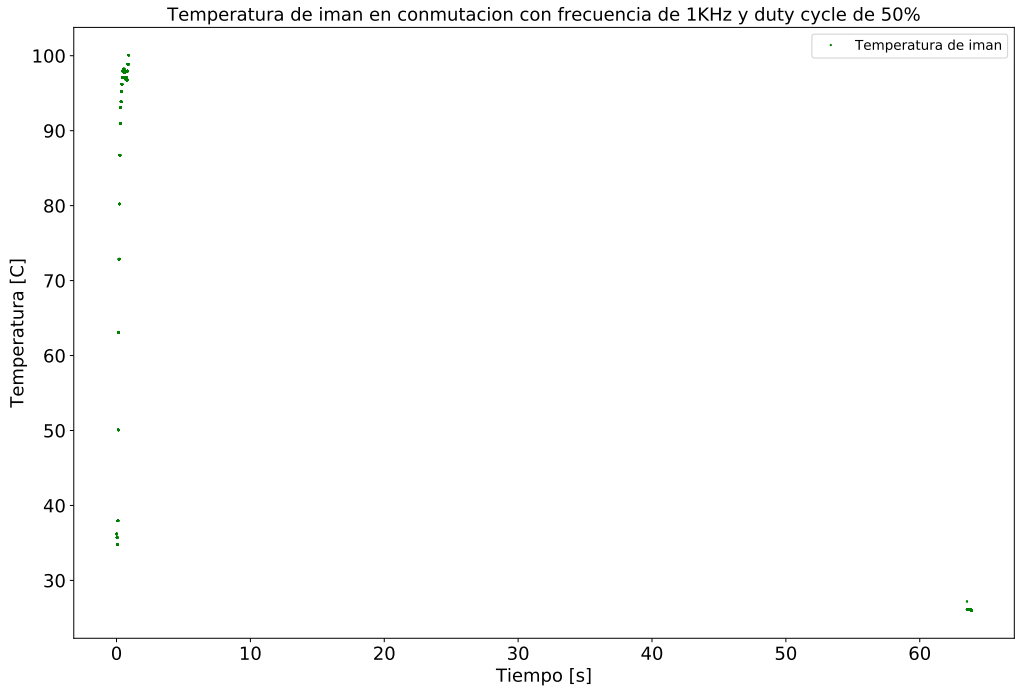


Figura 6.2: Temperatura de imán de un experimento a lo largo del tiempo.

Para analizar el límite de corriente de 150 A, se utilizó un generador de funciones para simular la tensión que ingresa por el sensor de efecto Hall, incrementándola hasta superar el límite establecido. La figura 6.3 muestra esta prueba, donde puede observarse que al superar los 150 A, el PSoC, junto a la PC, terminan el experimento.

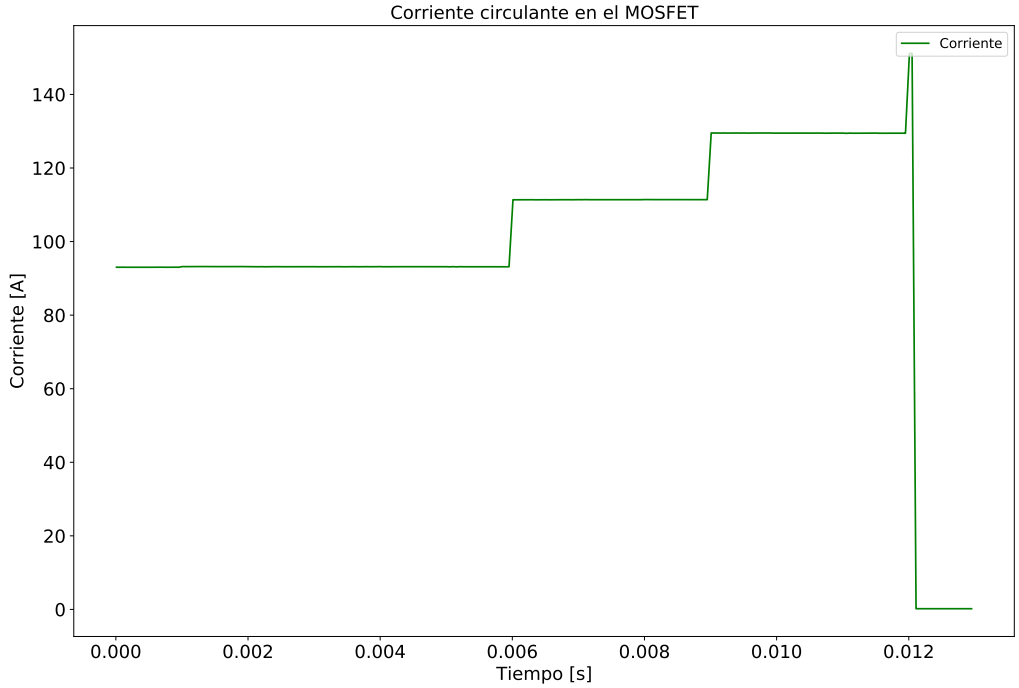
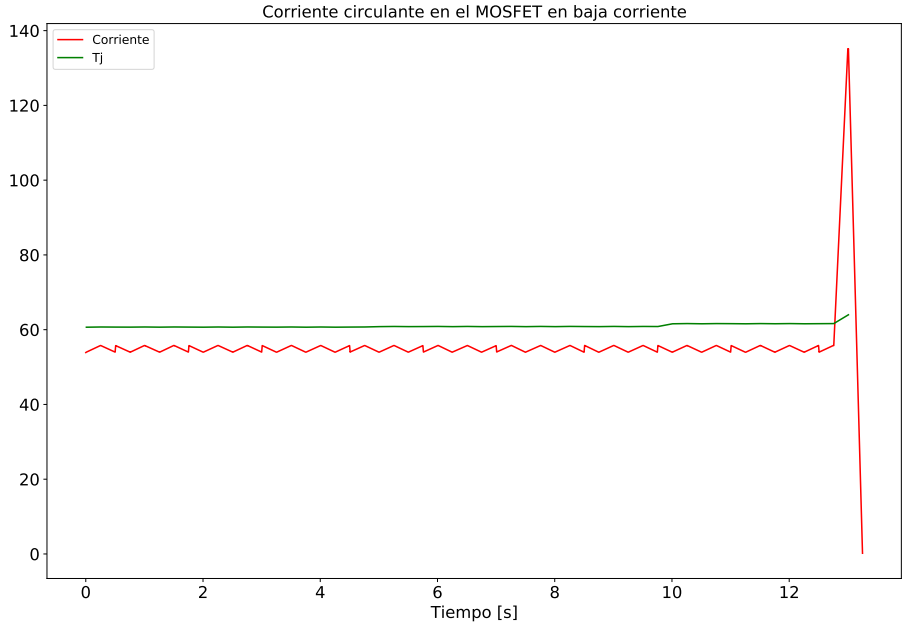


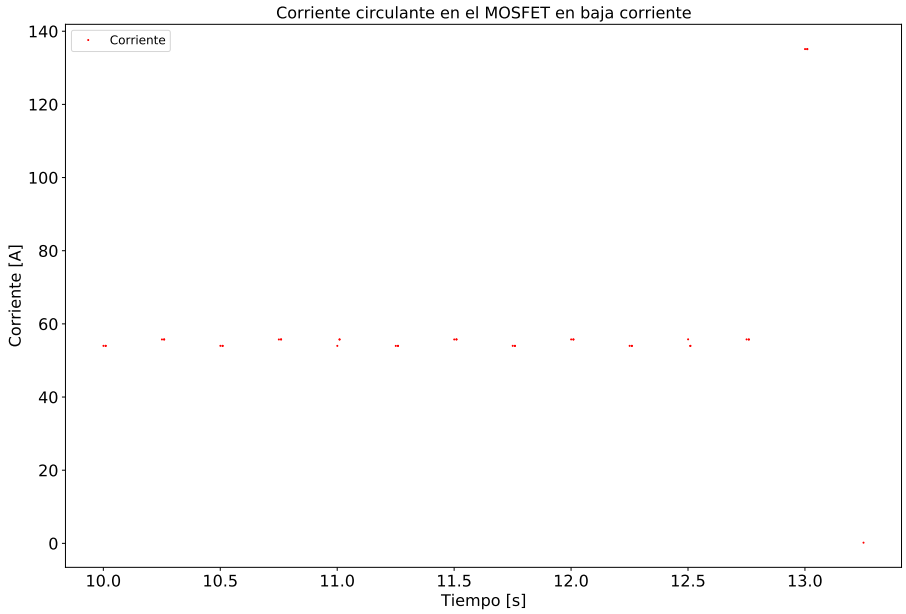
Figura 6.3: Límite de corriente (150 A)

En la figura 6.4 y figura 6.5 puede apreciarse que al momento de ocurrir un pico de corriente durante la experimentación por baja corriente (y conmutación respectivamente), el PSoC actúa de

inmediato cortando la señal del *gate* y la corriente sobre el circuito, evitando así que el transistor pueda romperse. Cabe destacar para la figura 6.4 que la temperatura de juntura es baja (menor a 60 $°C$) pero la corriente es el doble de alta.



(a) Limite por punto fuera de SOA



(b) Comportamiento de los datos previos al pico de corriente

Figura 6.4: Límite de SOA en baja corriente

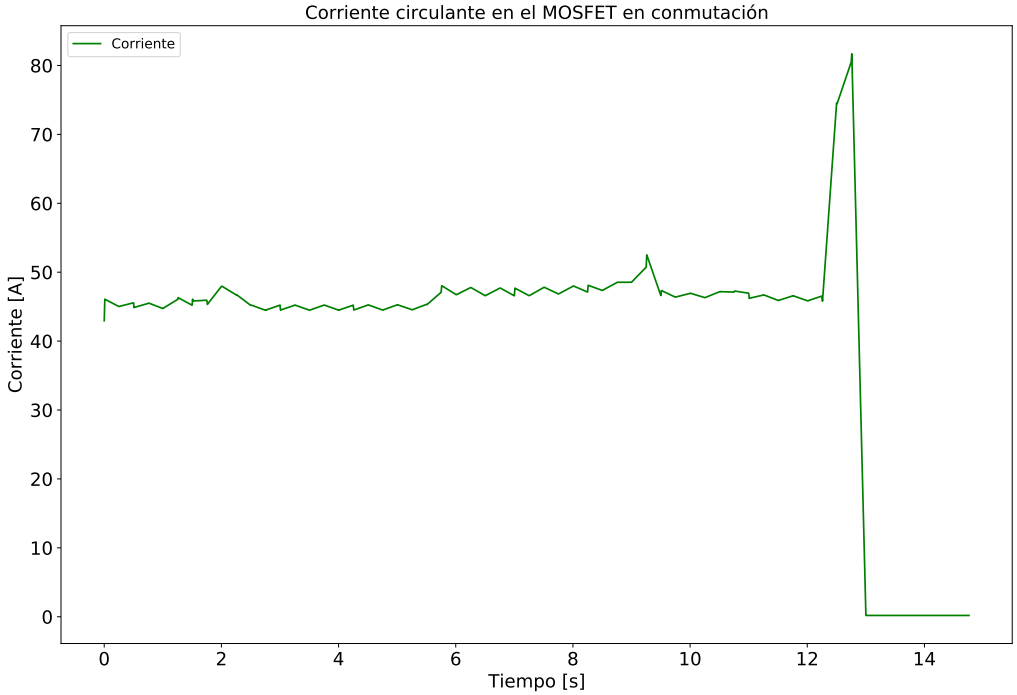


Figura 6.5: Límite de SOA en conmutación

Para ambos casos, se utilizó una frecuencia de 4 Hz y un *duty cycle* de 4 %. Eso termina brindando un tiempo de encendido de 1 *ms*, por lo que se utliza la curva de SOA correspondiente a tensión continua. También hubo un valor fijo en el terminal del *drain* de 5 V. Para el caso de la degradación por baja corriente, la temperatura de juntura se encontró alrededor de los 60 $°C$, una temperatura de cápsula de $59°C$, por lo que la corriente que circula por el transistor no debe superar los 85 A.

6.1 Regresor

En las figuras 6.6 y 6.7 se muestra el pronostico de corriente en el *drain* y de temperatura de juntura, respectivamente, para distintos momentos de un proceso de degradación. En esta figuras, puede verse en rojo el punto medido y en negro el valor pronosticado. Para este caso en particular, el error MAPE fue de 0.26 % para el pronósitico de corriente y 0.61 % para la temperatura de juntura. Podemos concluir que el regresor se comportó dentro de los parámetros necesarios por el sistema y que es posible confiar en la predicción de este para poder tomar las medidas necesarias sobre el transistor.

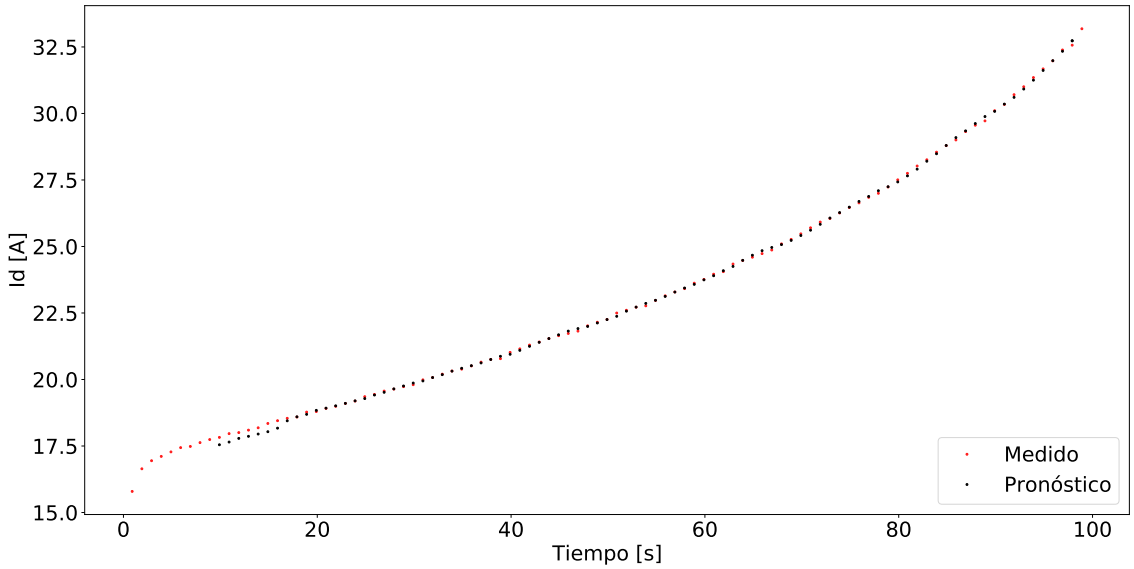


Figura 6.6: Regresor de corriente de una degradación

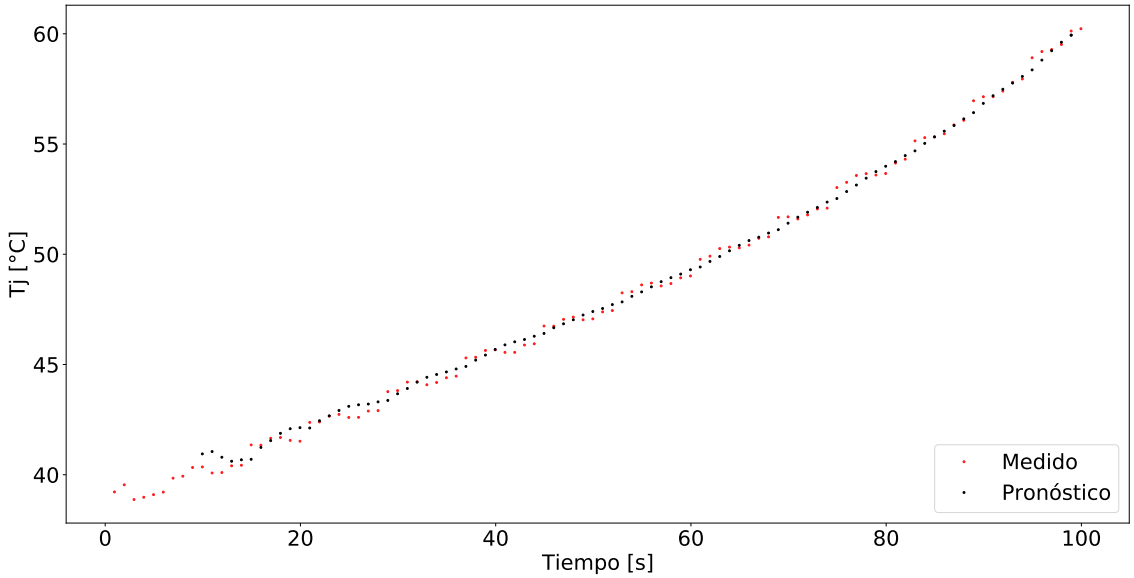


Figura 6.7: Regresor de temperatura de juntura de una degradación

7. Conclusión

Se presentó una plataforma configurable capaz de controlar el ensayo de degradación acelerada, medir los parámetros críticos y de funcionamiento del MOSFET que pueden causar la degradación y/o rompimiento de éste. Además, permite el almacenamiento en una base de datos local que puede ser consultada cuando el usuario lo desee.

Para la realización y verificación del programa se realizaron múltiples experimentos sobre MOSFETs, utilizando diferentes niveles de corriente y tensión, midiendo a la par con un osciloscopio para asegurar que las mediciones de la PC eran correctas. Cabe destacar que esto ayudó a detectar también problemas en la placa de adaptación de señales, que se fueron arreglando a medida que surgían. El proceso reiterativo e incremental de ir midiendo con ambos recursos fue clave para encontrar el criterio de selección de puntos para el regresor, ya que inicialmente se pensó que todos los puntos iban a tener un comportamiento lineal.

Algunas mejoras podrían realizarse en el software de la PC. Se podría compilar para diferentes plataformas, ya que el sistema operativo utilizado fue Windows y no se probó para otros. Realizar *tests* automáticos para verificar el correcto funcionamiento de los regresores para cambios futuros o actualización de la librería de GSL. Una simulación automatizada de comunicación ayudaría a probar cambios nuevos, tanto en el hardware como en el software, de manera más sencilla y rápida.

En caso de contarse con un computador con más recursos y múltiples placas de adaptación de señales y microcontroladores, se podrían realizar las modificaciones pertinentes en el software para poder ejecutarse degradaciones en paralelo para acelerar aún más las experimentaciones.

Con un microcontrolador más potente se podría ejecutar el algoritmo de regresión allí, lo que incrementaría la cantidad de pulsos o incluso podría eliminar la necesidad de interacción con la PC durante la degradación, lo que aceleraría aún más el proceso.

A pesar de que podrían realizarse estas mejoras, la plataforma cumplió con todas las especificaciones y requerimientos iniciales. El programa cumplió correctamente con los límites de control de los experimentos y el regresor fue lo suficientemente preciso, rápido y confiable para poder evitar que el transistor trabaje en condiciones extremas.

Apéndice

- ADC: Analog Digital Converter
- BJT: Bipolar Junction Transistor
- JFET: Junction Field Effect Transistor
- DAC: Digital Analog Converter
- DC: Direct Current
- ECM: Error cuadrático medio
- FET: Field Effect Transistor
- FFC: Fast Field Cycling
- MAPE: Mean Absolute Percentage Error
- MOS: Metal Oxide Semiconductor
- MOSFET: Metal Oxide Semiconductor Field Effect Transistor
- PC: Personal Computer
- PGA: Programmable Gain Amplifier
- PSoC: Programmable System on Chip
- PWM: Pulse Width Modulation
- RAM: Random Access Memory
- RMN: Resonancia Magnética Nuclear
- SAR: Successive-Approximation Register
- SOA: Safe Operation Area
- UART: Universal Asynchronous Receiver Transmitter
- UDB: Universal Digital Block
- USB: Universal Serial Bus
- VDAC: Voltage Digital Analog Converter

Bibliografía

[1] M. H. Rashid, *Electrónica de potencia: circuitos, dispositivos y aplicaciones*. Pearson Educación, 2004.

[2] B. J. Baliga, *Fundamentals of power semiconductor devices*. Springer Science & Business Media, 2010.

[3] N. Iwamuro, "Wide bandgap semiconductor power devices," 2019.

[4] IXYS, "Mosfet ixtn660n04t4," available at https://www.littelfuse.com/~/media/electronics/datasheets/discrete_mosfets/littelfuse_discrete_mosfets_n-channel_trench_gate_ixtn660n04t4_datasheet.pdf.pdf.

[5] B. J. Baliga, *Fundamentals of Power Semiconductor Devices*. 1010 Main Campus Drive, USA: Springer Science, 2008.

[6] R. Electronics, "Unclamped inductive switching (uis) test and rating methodology," 2015.

[7] J. Schoiswohl, "Linear mode operation and safe operating diagram of power-mosfets," *Infineon Application Note*, 2010.

[8] B. Yarborough, "Components and methods for current measurement," *Power Electronics*, 2012.

[9] G. A. Dominguez, "Relaxometría magnética nuclear con campo magnético ciclado de baja homogeneidad. conmutación ultra rápida." 2016.

[10] C. S. Corporation, "Psoc 5lp cy8c5868axi-lp035," available at https://www.cypress.com/part/cy8c5868axi-lp035.

[11] F. Edition, M. M. Mano, C. R. Kime, and T. Martin, "Logic and computer design fundamentals," 2015.

[12] C. S. Corporation, "Universal digital block (udb) editor guide," available at https://www.cypress.com/file/139386.

[13] ——, "Adc successive approximation register," available at https://www.cypress.com/file/179601.

[14] B. Baker, "Delta-sigma adcs in a nutshell," *EDN*, 2007.

[15] B. Carter and T. R. Brown, *Handbook of operational amplifier applications*. Texas Instruments Dallas, Tex, USA, 2001.

[16] M. Integrated, "1-wire parasite-power digital thermometer," available at https://www.maximintegrated.com/en/products/sensors/DS18S20.html.

[17] "Component to read ds18b20 digital temperature sensors," available at https://community.cypress.com/thread/28125.

[18] A. R. D. M. S. E. M. Marco Lima, Bruno Pereira, "Comparing two power supplies for fast field cycling nuclear magnetic resonance relaxometers: Power losses and performance." 2016.

[19] L. Dupont, S. Lefebvre, M. Bouaroudj, Z. Khatir, and J.-C. Faugières, "Failure modes on low voltage power mosfets under high temperature application," *Microelectronics reliability*, vol. 47, no. 9-11, pp. 1767–1772, 2007.

[20] R. Zivanovic, "Local regression-based short-term load forecasting," *Journal of Intelligent and Robotic Systems*, vol. 31, no. 1-3, pp. 115–127, 2001.

[21] W. Charytoniuk, M. Chen, and P. Van Olinda, "Nonparametric regression based short-term load forecasting," *IEEE transactions on Power Systems*, vol. 13, no. 3, pp. 725–730, 1998.

[22] G. Dudek, "Pattern-based local linear regression models for short-term load forecasting," *Electric Power Systems Research*, vol. 130, pp. 139–147, 2016.

[23] H. Sun, H. X. Liu, H. Xiao, R. R. He, and B. Ran, "Use of local linear regression model for short-term traffic forecasting," *Transportation Research Record*, vol. 1836, no. 1, pp. 143–150, 2003.

[24] M. Gorlov, A. Strogonov, and D. Y. Smirnov, "Transistor-degradation prediction by time-series analysis," *Russian Microelectronics*, vol. 35, no. 5, pp. 337–344, 2006.

[25] F. S. Foundation, "Gnu," available at https://www.gnu.org.

[26] T. Oliphant, "Numpy," available at https://numpy.org.