



# DMA e Introducción a la IEC61508

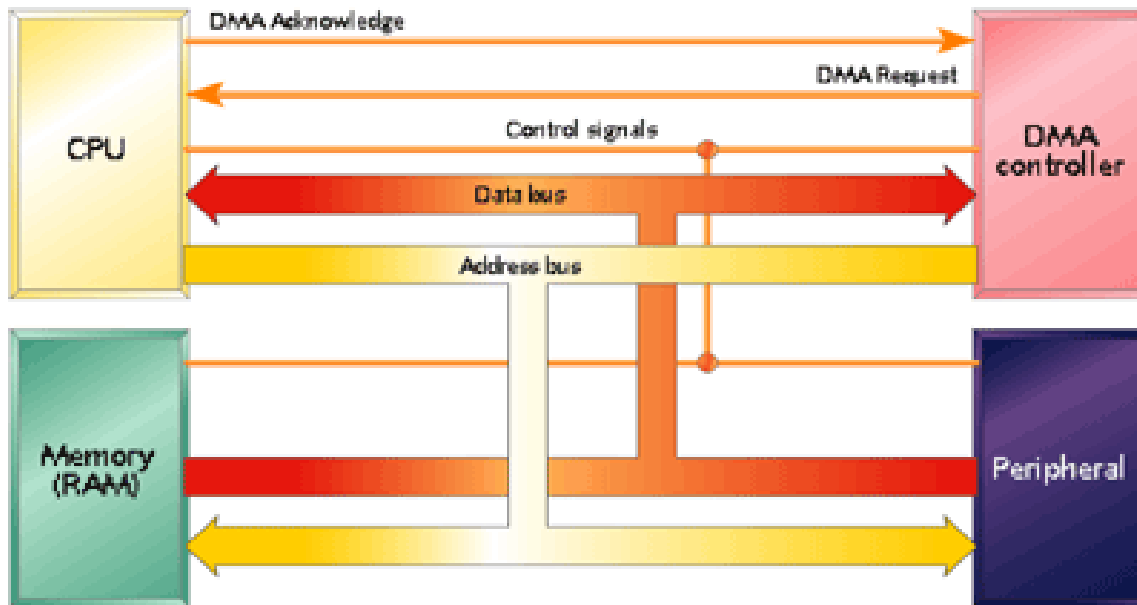
# Agenda.

- DMA
- IEC 61508.
  - ☐ Equipo bajo control
  - ☐ Riesgo.
  - ☐ Ciclo de vida.
  - ☐ Análisis de riesgo
  - ☐ Riesgo tolerable.
  - ☐ Niveles de Integridad de seguridad

# DMA (acceso directo a memoria)

- En una arquitectura sin DMA (acceso directo a memoria), independientemente de que sea Harvard o von Neumann el bus de direcciones **siempre** lo maneja el procesador y es el medio por el cuál selecciona con que periférico interactuar.
- ¿Qué pasa cuando se quiere utilizar un periférico que transmite datos a una velocidad más alta que la del procesador?
- O un sistema con varios procesadores que comparten zonas de memoria.
- En este momento ya deja, o puede dejar, de ser válida la idea de que el bus de direcciones lo maneje solamente un microprocesador.
- En general, en un sistema con un único procesador, vamos a considerar que un sistema tiene DMA cuando cuenta con un periférico que puede **copiar zonas de memoria** a “espaldas” del microprocesador. Es decir, un periférico que puede ser leído y escrito por el procesador, pero que además también puede leer y escribir otros periféricos.

# DMA (acceso directo a memoria)

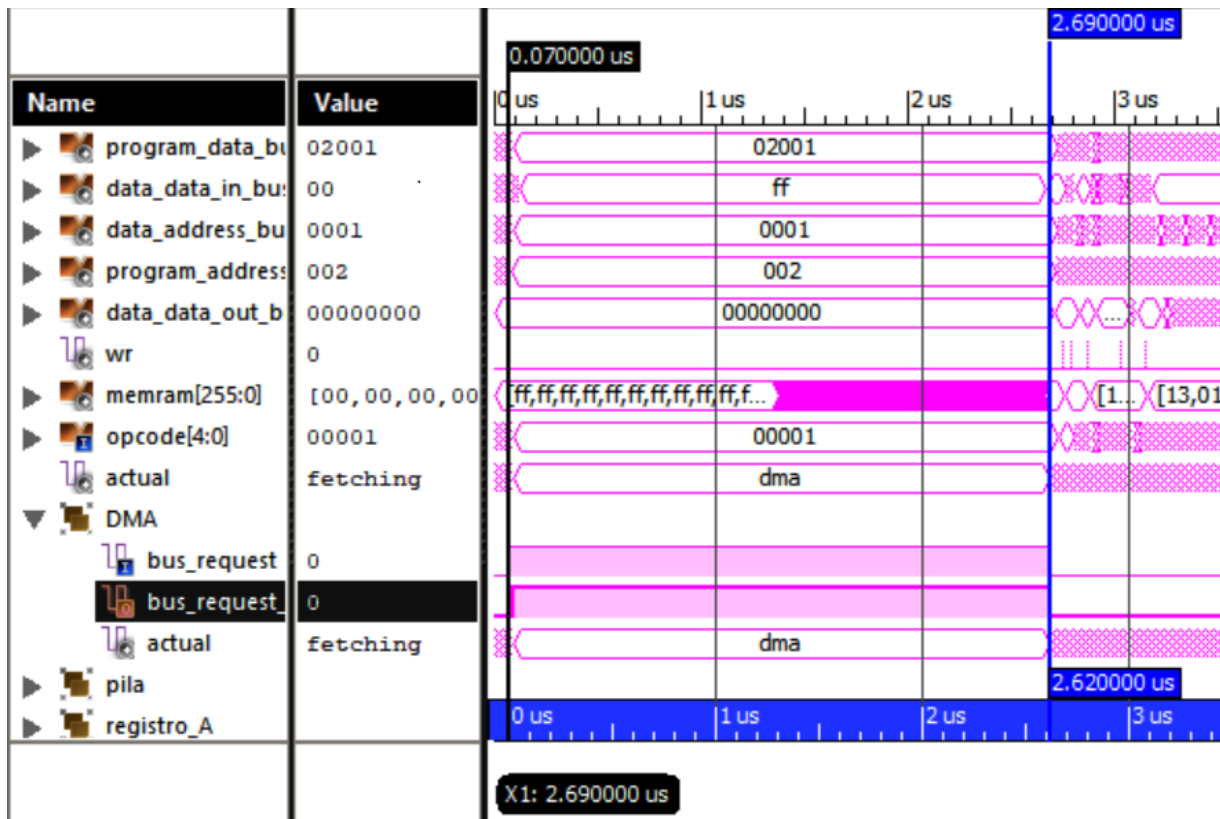


- El bus de direcciones puede ser escrito por el CPU y por el controlador de DMA.
- ¿Qué pasa con la CPU mientras está usando el bus de direcciones el controlador de DMA?

# Ejemplo DMA.

- Dado el código en VHDL de [https://gitlab.frba.utn.edu.ar/jalarcon/ejemplo\\_dma\\_vhdl.git](https://gitlab.frba.utn.edu.ar/jalarcon/ejemplo_dma_vhdl.git) que tiene el código de un muy pequeño procesador RISC con posibilidad de trabajar con DMA.
- En el archivo nano\_tb.vhdl se genera un sistema donde se borra por DMA toda la memoria de datos que se inicializa en 0xFF.
- Se pide:
  - Se adjunta también el archivo fuente led\_dma.asm. ¿En qué punto del programa se borra la memoria? ¿Cuánto se tarda por DMA? ¿Cuánto se tardaría por código?
  - ¿Qué está haciendo el procesador mientras se borra la memoria? ¿Qué señales lo indican?
  - ¿Qué recursos adicionales se tienen que agregar para poder usar el DMA? ¿Cuántas señales de WR tiene el DMA? ¿Por qué?

# Ejemplo DMA (2).



- Considerando que el período de reloj es de 10ns y que cada instrucción se completa en 4 ciclos de reloj.
- ¿Cuántas posiciones de memoria se pondrían a cero en el tiempo que el DMA borró toda la memoria?

# Manejo de E/S por DMA

- Los sistemas de acceso directo a memoria (DMA) son sistemas que pueden controlar la memoria del sistema sin el uso de la CPU.
- Dado un evento determinado el módulo de DMA puede mover datos entre diferentes zonas de memoria.
- Si bien es menos flexible que la CPU, es mucho más rápido en la copia de datos que la CPU

# E/S por DMA

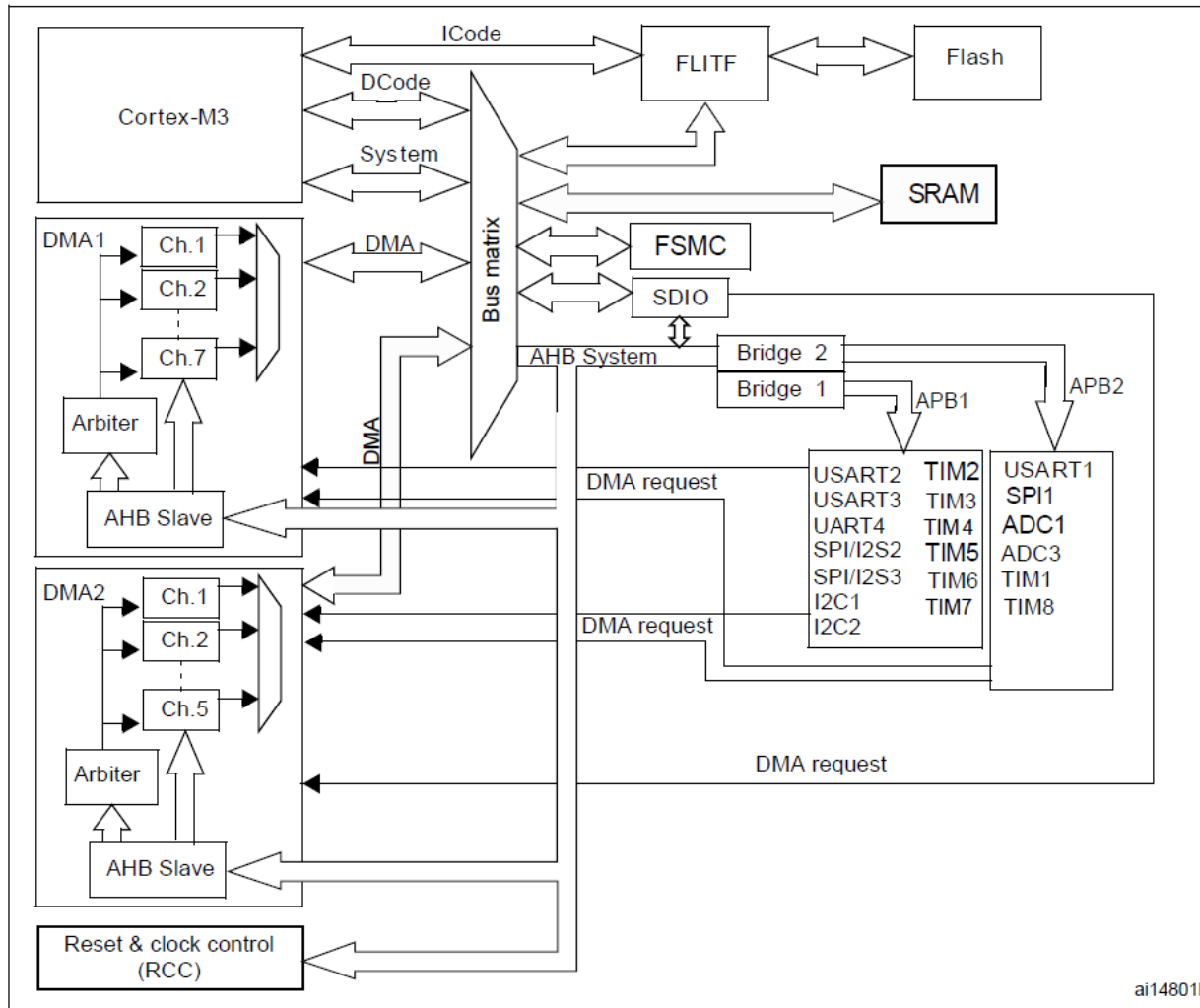
- Algunos periféricos pueden generar grandes cantidades de datos (ADC, DAC, Ethernet, etc.).
- Las transferencias por DMA son las más adecuadas para grandes volúmenes de datos o bien para altas tasas de datos.



# DMA. Generalidades.

- Los sistemas de DMA pueden funcionar de diferentes formas:
  - **Transferencia única.** Un byte, word, doble-word se transfiere por operación de DMA. Se suele utilizar en dispositivos de flujo de caracteres. Cada transferencia deshabilita la CPU.
  - **Transferencias por bloques.** El módulo transfiere un bloque completo de memoria, mientras que la CPU no se encuentra activa.
  - **Transferencia por ráfaga.** El módulo de DMA transfiere un bloque de memoria, pero no desactiva la CPU, sino que alterna los accesos a la memoria con ésta

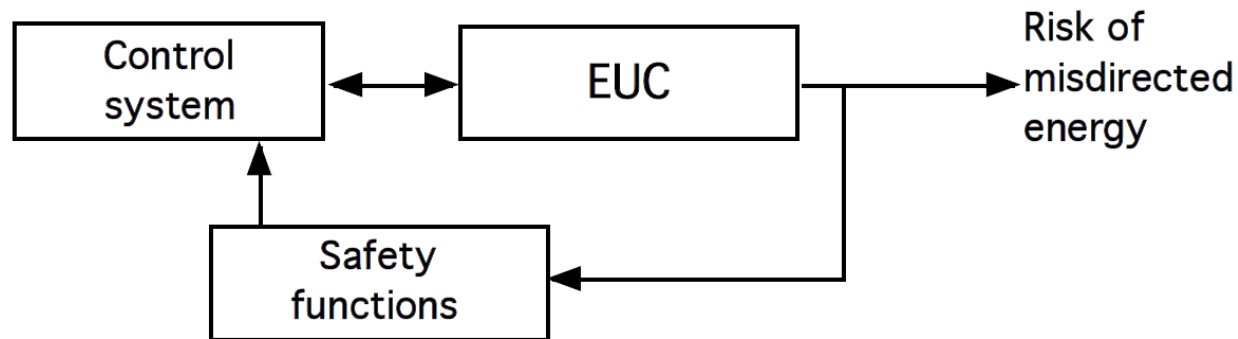
# DMA en STM32F103C8



# Manos a la obra

- Con el código [https://gitlab.frba.utn.edu.ar/jalarcon/ejemplo\\_dma.git](https://gitlab.frba.utn.edu.ar/jalarcon/ejemplo_dma.git) este código copia un bloque de memoria a otro bloque de memoria de 5Kbytes ambos. Tiene dos modos de funcionamiento, con DMA y sin DMA. Ejecútelo en ambos modos y determine:
  - Estimaciones de BCET y WCET para ambos modos.
  - ¿Encuentra diferencias? Determine la carga de la CPU para ambos modos y determine la planificabilidad de ambos.

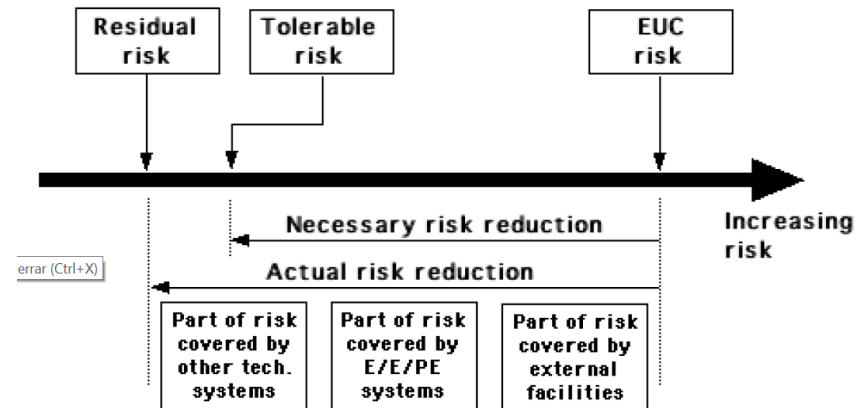
# IEC61508



- La norma IEC 61508 se publica en 1998.
- La principal característica de de la norma es que introduce el concepto de que el equipo bajo control (equipment under control) implementa una **función**.
- El estándar hace énfasis en **la reducción de riesgos en las funciones de seguridad**.
- Las funciones de seguridad son implementadas en **sistemas de protección**

# Riesgo

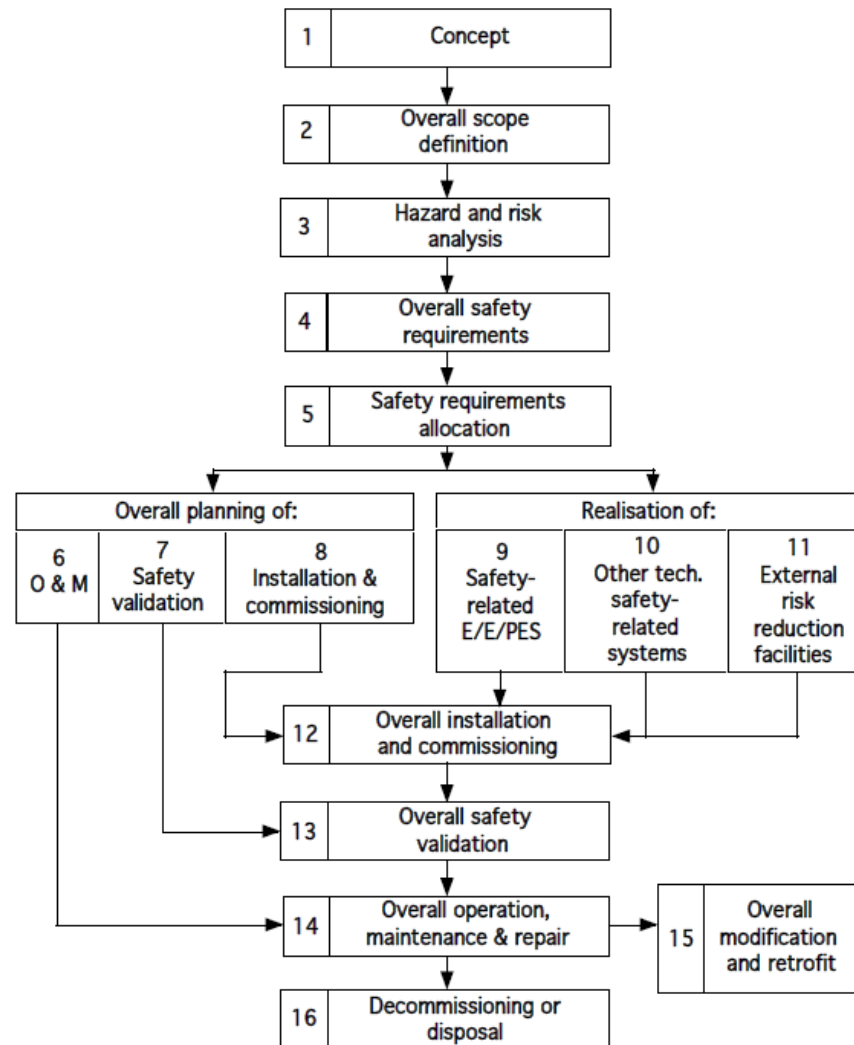
- La norma utiliza el concepto de riesgo y la definición de:
  - Riesgo actual.
  - Riesgo tolerable (El que se considera adecuado para la aplicación)
  - Riesgo residual ya que conceptualmente ***nunca se alcanza riesgo cero.***
- La reducción de riesgo se puede dar por la utilización de mecanismos distintos o ***diversos.***



# Ciclo de vida

- La norma define un conjunto de **buenas prácticas** y recomendaciones pero no asume la responsabilidad de la seguridad por el diseño.
- La IEC61508 no es una norma para el desarrollo, pero si es una norma para la gestión de la seguridad a lo largo de la **vida de un proyecto**.
- La norma define el ciclo de vida como un conjunto de etapas que deben cumplirse para pasar a la etapa siguiente en el desarrollo del proyecto.

# Ciclo de vida



# Análisis de riesgo

- Una parte importante de la norma, son los requerimientos de seguridad que se obtienen a partir del análisis de riesgo que posee el equipo bajo control y su sistema de control.
- El análisis de riesgo define tres etapas:
  - Identificación de peligros (hazard)
  - Análisis de peligros probables.
  - Asignación de riesgos
- Se define como peligro a una ***potencial fuente de daño.***
- Un equipo bajo control posee muchos potenciales peligros con su correspondiente riesgo



# Análisis de riesgo (2)

Table 1: Defining categories of likelihood of occurrence

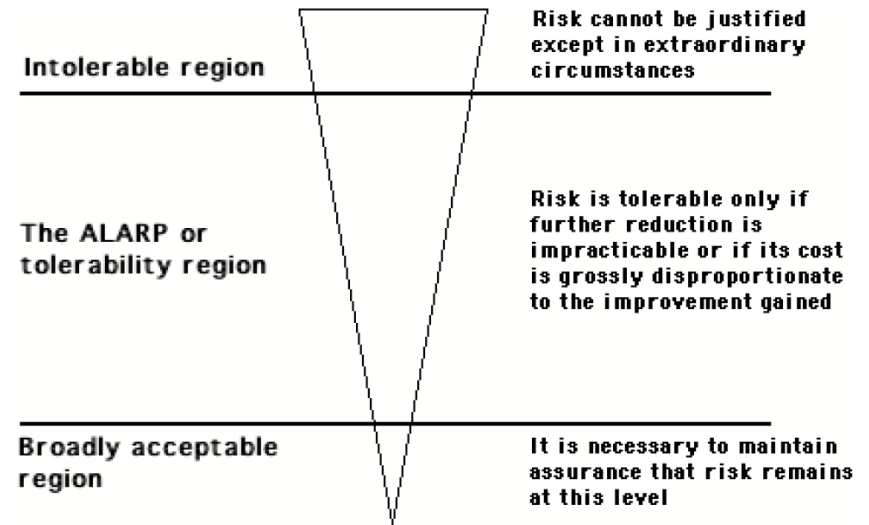
Category	Definition	Range (failures per year)
Frequent	Many times in system lifetime	$> 10^{-3}$
Probable	Several times in system lifetime	$10^{-3}$ to $10^{-4}$
Occasional	Once in system lifetime	$10^{-4}$ to $10^{-5}$
Remote	Unlikely in system lifetime	$10^{-5}$ to $10^{-6}$
Improbable	Very unlikely to occur	$10^{-6}$ to $10^{-7}$
Incredible	Cannot believe that it could occur	$< 10^{-7}$

Table 2: Defining consequence categories

Category	Definition
Catastrophic	Multiple loss of life
Critical	Loss of a single life
Marginal	Major injuries to one or more persons
Negligible	Minor injuries at worst

# Riesgo tolerable.

- Para cuestiones de software el análisis de riesgo suele ser cualitativo ya que el software suele fallar de manera sistemática.
- Por lo que la pregunta es ¿Cuál es el riesgo tolerable?
- Lo que es tolerable para una aplicación puede no serlo para otra.
- Se define lo que se conoce como principio ALARP (as low as reasonably practicable – tan bajo como sea razonablemente posible).



# Clases de riesgo.

LIKELIHOOD	CONSEQUENCE			
	Catastrophic	Critical	Marginal	Negligible
Frequent	I	I	I	II
Probable	I	I	II	III
Occasional	I	II	III	III
Remote	II	III	III	IV
Improbable	III	III	IV	IV
Incredible	IV	IV	IV	IV

- Una forma de aplicar el análisis de riesgo es a través de las **clases de riesgo**.
  - Clase 1. Inaceptable bajo cualquier circunstancia.
  - Clase 2. Indeseable, solamente tolerable si la reducción de riesgo es impracticable o excesivamente costosa
  - Clase 3. Tolerable si el costo de reducción del riesgo excede a la mejora.
  - Clase 4. Tolerable, pero debe ser monitoreado.
- Estas consideraciones nos generan la matriz de riesgo de la derecha.

# Niveles de integridad de seguridad

- La norma define a la integridad de seguridad como ***la probabilidad de un sistema relacionado con la seguridad de realizar su función satisfactoriamente en cualquier condición en un período de tiempo dado.***
- Los niveles de integridad de seguridad (SIL) no son una medida de riesgo, sino una medida de la confiabilidad del sistema.
- Se definen cuatro niveles de integridad de seguridad.

# Niveles de integridad de seguridad

Table 5: Safety integrity levels

<b>Safety Integrity Level</b>	<b>Low Demand Mode of Operation (Pr. of failure to perform its safety functions on demand)</b>	<b>Continuous/High-demand Mode of Operation (Pr. of dangerous failure per hour)</b>
4	$\geq 10^{-5}$ to $10^{-4}$	$\geq 10^{-9}$ to $10^{-8}$
3	$\geq 10^{-4}$ to $10^{-3}$	$\geq 10^{-8}$ to $10^{-7}$
2	$\geq 10^{-3}$ to $10^{-2}$	$\geq 10^{-7}$ to $10^{-6}$
1	$\geq 10^{-2}$ to $10^{-1}$	$\geq 10^{-6}$ to $10^{-5}$

# IEC61508. Resumen

- La norma trata la gestión de la seguridad de un sistema programable a lo largo de toda su vida, desde su diseño hasta su desmantelamiento.
- Al igual que la mayoría de las normas no pone condiciones en el desarrollo en sí, sino principalmente en la gestión del sistema.
- Es clave para la norma el concepto de riesgo, pero también la definición de requerimientos en función de la aplicación.

# IEC61508 y otras normas

<b>Generic</b> (IEC 61508)	<b>(SIL 0)</b>	<b>SIL 1</b>	<b>SIL 2</b>	<b>SIL 3</b>	<b>SIL 4</b>
<b>Civil Aerospace</b> (DO-178C)	Level E	Level D	Level C	Level B	Level A
<b>Medical</b> (IEC 62304)	Class A	Class B		Class C	
<b>Automotive</b> (ISO 26262)	QM	ASIL A	ASIL B / ASIL C	ASIL D	--
<b>Machinery</b> (ISO 13849)	PL a	PL b / PL c	PL d	PL e	--
<b>Household</b> (IEC 60730)	Class A	Class B		Class C	--

# Bibliografía.

- Sistemas Operativos. Diseño e Implementación. Andrew S. Tanenbaum.
- Sistemas de Tiempo Real y Lenguajes de Programación. Alan Burns, Andy Wellings. Tercera Edición. Addison Wesley.
- [IEC 1998] Draft Standard IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems. International Electrotechnical Commission, Geneva, 1998