ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

Sistema de Inyección de Soft-errors (SEU, SEFI) en un microcontrolador SAMV71

Versión B

Escrito por Gonzalo Nahuel Vaca

FIUBA

6 de julio de 2021

Índice

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
A	Creación del documento	27/06/2021
B	Se agrega encabezado en la plantilla del documento. Modificación de la tabla de registro de cambios. Nuevo formato de enumeración de requisitos. Se amplia la sección 2.1.	03/07/2021

1. Introducción

1.1. Propósito

Este documento representa una especificación de requerimientos de software para un sistema de inyección de *soft-errors* y un *firmware* de *self-testing*. Está dirigido a las personas que se ocupen de las siguientes tareas:

- análisis
- diseño
- implementación
- pruebas

1.2. Ámbito del sistema

El nombre del sistema será SISE y permitirá inyectar errores en todos los registros accesibles del microcontrolador *SAMV71*. Su función será evaluar las técnicas de mitigación de *soft-errors* en funciones a ser utilizadas en la misión *Sabiamar*. Adicionalmente, se proveerá un *firmware* de *self-testing* para determinar los presupuestos de *hardware*.

El beneficio que se espera obtener es utilizar componentes que no fueron sometidos a un proceso de calificación (alternativos). Además, se espera simular los 5 años de misión en un tiempo acelerado.

1.3. Definiciones, acrónimos y abreviaturas

1. Definiciones:
   - Sabiamar: constelación de dos satélites argentino-brasileños para la información del mar.
   - soft-errors: modificación no destructiva del valor de un registro o memoria.
2. Acrónimos:
   - CSV: comma separated value.
   - IEEE: Institute of Electrical and Electronics Engineers.
   - JTAG: Join Test Action Group.
   - RAM: random access memory.
   - TCP: transfer control protocol.
3. Abreviaturas:
   - Std: estándar.

1.4. Referencias

INVAP - Propuesta de tesis: sistema de inyección de soft-errors.

1.5. Visión general del documento

Este documento se realizó según lo especificado en el estándar IEEE Std. 830-1998.

2. Descripción general

2.1. Perspectiva del producto

El software aquí especificado es independiente de otros sistemas y no tiene relación con otros productos.

El principio de inyección de *soft-erros* se basará en conectarse al microcontrolador a través de su interfaz de *debug*; se procederá a suspender la ejecución del *firmware* y luego se realizarán las modificaciones necesarias.

En la figura 1 se puede observar un esquema general del proceso de inyección de *soft-errors*.

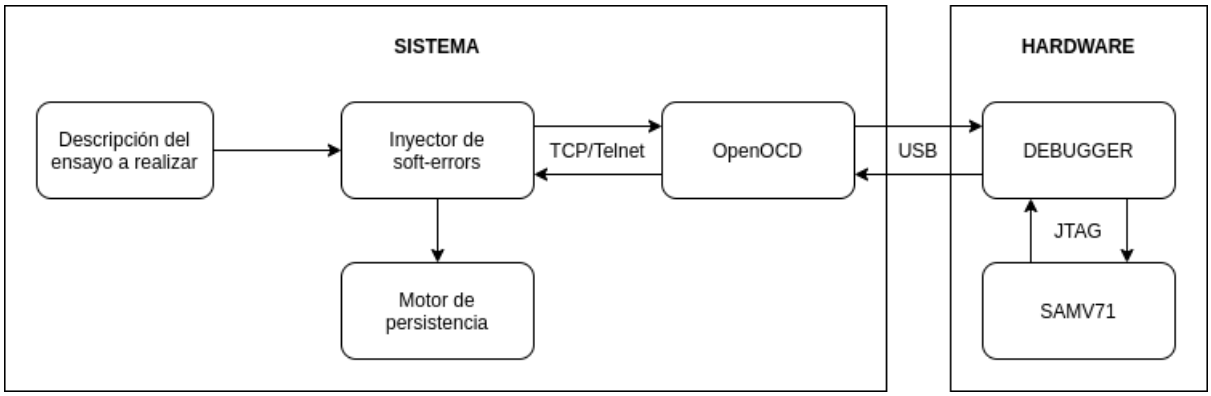

Figura 1. Esquema general de inyección de *soft-errors*.

El inyector de *soft-errors* deberá modificar obtener la información de los registros del microcontrolador SAMV71 antes de realizar una modificación. Finalmente, debe persistir la operación realizada junto a los datos previos a la inyección.

Adicionalmente, se entregará un *firmware* de *self-testing* que generará reportes sobre el funcionamiento de los periféricos del microcontrolador *SAMV71*. En la figura 2 se puede ver un esquema general del proceso de *self-testing*.

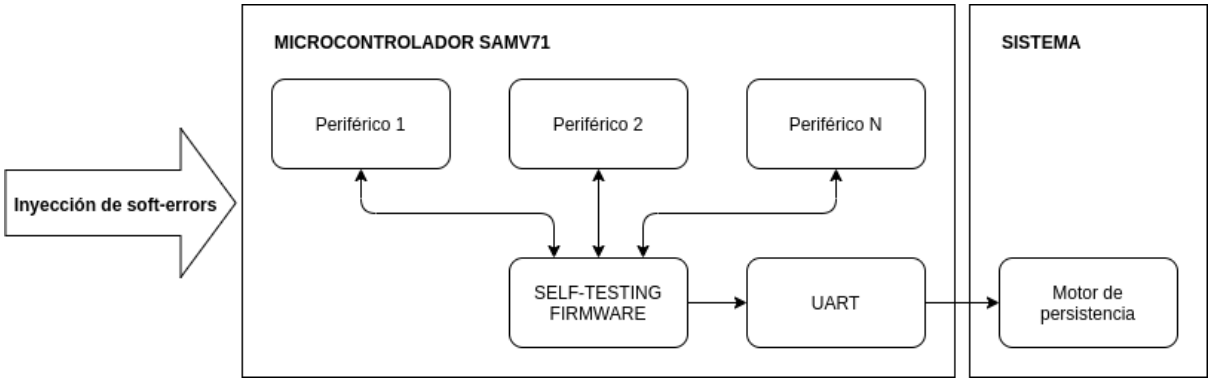

Figura 2. Esquema general del proceso de *self-testing*.

2.2. Funciones del producto

El software aquí especificado brindará las siguientes funcionalidades:

- Inyección de errores en todos los registros accesibles del microcontrolador SAMV71.
- Monitoreo del estado de funcionamiento del microcontrolador SAMV71.
- Persistencia de los soft-errors inyectados.
- Persistencia de los informes de estado de funcionamiento.
- Permitir escribir ensayos de evaluación.
- Presentación de resultados en histogramas que permitan un análisis estadístico.

2.3. Características de los usuarios

Los usuarios finales de este producto son ingenieros de desarrollo del INVAP.

2.4. Restricciones

Las restricciones del desarrollo del sistema son las siguientes:

- Utilización de repositorio con control de versiones *Gitlab*.
- Documentación del código con *Doxygen*.
- Utilización exclusiva del lenguaje de programación *Python 3*.

2.5. Suposiciones y dependencias

La suposición principal es que se tendrá acceso irrestricto al microcontrolador *SAMV71* antes del día 01/11/2021.

2.6. Requisitos futuros

N/A

3. Requisitos específicos

3.1. Interfaces externas

- **[SISE-RS-001]:** se comunicará de forma bidireccional con *OpenOCD* a través de *TCP* o *Telnet*.
- **[SISE-RS-002]:** se deberá recibir y capturar la información proveniente por puerto *Serial* del microcontrolador *SAMV71*.

3.2. Funciones

1. Inyección de *soft-errors*:
   - **[SISE-RS-003]:** sobrescribirá los valores en memoria *RAM*.
   - **[SISE-RS-004]:** invertirá el valor de un bit en la memoria *RAM*.
   - **[SISE-RS-005]:** sobrescribirá valores en los registros internos.
   - **[SISE-RS-006]:** invertirá el valor de un bit dentro de un registro interno.
   - **[SISE-RS-007]:** reportará el estado del microcontrolador antes de realizar una inyección de *soft-errors*.
   - **[SISE-RS-008]:** interpretará una descripción de ensayo escrita en *Python 3* y ejecutará las inyecciones según le indique.
   - **[SISE-RS-009]:** aceptará descripciones de ensayo que impongan una tasa de error para cada registro interno.
   - **[SISE-RS-010]:** procesará descripciones de ensayo que impongan una tasa de error para una posición o rango de memoria *RAM*.
   - **[SISE-RS-011]:** las descripciones de ensayo podrán especificar probabilidades de error con una resolución de 1 bit.
2. *Firmware* de *self-testing*:
   - **[SISE-RS-012]:** detectará el funcionamiento anormal de los periféricos del microcontrolador *SAMV71*.
   - **[SISE-RS-013]:** reportará periódicamente el estado de los periféricos utilizando el protocolo *Serial*.
   - **[SISE-RS-014]:** los reportes tendrán un formato *CSV* y un tamaño menor a 5 kB.
3. Almacenamiento de reportes:
   - **[SISE-RS-015]:** se incluirá un *timestamp* de recepción del reporte.
   - **[SISE-RS-016]:** se generarán histogramas para su análisis estadístico.

3.3. Requisitos de rendimiento

[SISE-RS-017]: el inyector de soft-errors deberá realizar la inserción solicitada en un tiempo menor a 10 ms.

3.4. Restricciones de diseño

[SISE-RS-018]: se utilizará el microcontrolador *SAMV71* como dispositivo principal.

3.5. Atributos del sistema

1. Mantenibilidad:
   - **[SISE-RS-019]:** el software deberá permitir su modificación para trabajar con otras arquitecturas.

3.6. Otros requisitos

N/A.

4. Apéndices

4.1. Restricciones acerca del lenguaje de programación

El lenguaje de programación será *Python 3* y el código deberá ser documentado según las recomendaciones del manual de usuario de *Doxygen*.

4.2. Casos de uso

Cuadro 1. Caso de uso número 1

Título	Descripción
1. Nombre	Simular misión Sabiamar
1.1 Breve descripción	Se simulan los 5 años de *soft-errors* en un lapso de 24 horas
1.2 Actor principal	Usuario del sistema
1.3 Disparadores	Comando de ejecución
2. Flujo de eventos	
2.1 Flujo básico	1. El software interpreta la descripción del ensayo 2. El software determina la tasa de falla del ensayo 3. El software determina la probabilidad de falla de cada registro 4. El software determina la probabilidad de falla en memoria 5. El software realiza inyecciones según los parámetros calculados 6. El software persiste todas las inyecciones realizadas 7. El software entrega un informe final 8. El software retorna un código de tarea finalizada
2.2 Fuljo alternativo	1. El software detecta una anormalidad en el ensayo 2. El software aborta el ensayo 3. El software genera un informe de fallo 4. El software retorna un código de error
3. Pre-condiciones	1. *OpenOCD* corriendo 2. *OpenOCD* conectado al microcontrolador 3. *OpenOCD* con puerto disponible
4. Pos-condiciones	*OpenOCD* con puerto liberado

Cuadro 2. Caso de uso número 2

Título	Descripción
1. Nombre	Validación de hardware
1.1 Breve descripción	Se prueba la capacidad de los periféricos del microcontrolador para reponerse de fallas
1.2 Actor principal	Banco de pruebas INVAP
1.3 Disparadores	Anormalidad en periférico
2. Flujo de eventos	
2.1 Flujo básico	1. El firmware detecta una anormalidad en un periférico 2. El firmware genera un informe en formato *CSV* 3. El firmware envía un reporte por *UART*
2.2 Fuljo alternativo	1. El microcontrolador se reinicia 2. El firmware genera un informe en formato *CSV* 3. El firmware envía un reporte por *UART*
3. Pre-condiciones	
4. Pos-condiciones	