

|  |
| --- |
| Proyecto Hardware |

|  |
| --- |
| Alberto Martínez Menéndez  Pablo Piedrafita Castañeda  23-10-2016 |

|  |
| --- |
| Memoria práctica 1 |

**Contenido**

[1 Resumen 2](#_Toc465010940)

[2 Introducción 3](#_Toc465010941)

[3 Objetivos 3](#_Toc465010942)

[4 Metodología 3](#_Toc465010943)

[5 Resultados 4](#_Toc465010944)

[6 Conclusión 4](#_Toc465010945)

[7 Referencias 5](#_Toc465010946)

# Resumen

Para la primera práctica de Proyecto Hardware se ha planteado el problema de diseñar una herramienta de asistencia para la resolución de sudokus. Se trata de una utilidad que permite mantener actualizados los candidatos de las casillas vacías. El énfasis de este trabajo está en el estudio de la eficiencia de diferentes implementaciones del asistente. Por ello se trabaja con tres lenguajes, cada uno destacando en un aspecto distinto. El lenguaje de programación de “alto nivel” *C* nos permite abstraernos del hardware y simplificar enormemente la implementación. El lenguaje de ensamblador de *ARM* nos permite un mayor control del procesador, permitiendo esto una potencial mejora de la eficiencia. *Thumb* por otro lado está enfocado al ahorro de memoria gracias a sus instrucciones de 16 bits.

El apartado de programación de la práctica consiste en el desarrollo de dos funciones. Una primera función de inicialización servirá para definir los candidatos iniciales en las celdas que no contienen pistas, estos serán en un primer lugar todos los valores de 1 a 9. La segunda función permitirá a partir de las pistas dadas preliminarmente actualizar los valores que no pueden ir en la casilla. Como estipulan las reglas del sudoku, en una misma fila, columna o sección no puede aparecer dos veces el mismo número, por ello las actualizaciones de los valores se harán en primer lugar en la fila correspondiente, posteriormente pasará a actualizar las columnas, y por último la sección. La función de inicialización ha sido implementada únicamente en *C* y en *ARM*, mientras que la de propagación ha sido implementada en los tres lenguajes.

Estas dos herramientas se desarrollan con la intención de usarlas en el trabajo próximamente. Por consiguiente, es importante decidir qué lenguaje es más apropiado en nuestro caso. Esto se decide, principalmente, basándonos en el coste temporal y de memoria de cada una de las implementaciones. Para valorar estos factores, se ha realizado un estudio tras la implementación. Se han hecho mediciones de: tiempos de ejecución, tamaño del código, etc. Al final, se elegirá el lenguaje que mejor coordine estas dos características, velocidad y tamaño.

# Introducción

Se nos da como problema el diseño de dos funciones para un juego de Sudoku que se ejecutará en la placa de prototipado *Embest S3CEV40*. Se trata de unas funciones de ayuda a la resolución de Sudokus. La primera función consiste en una inicialización de los candidatos teniendo en cuenta el formato establecido para la representación de los mismos, presentado más adelante en este documento. A continuación estos valores deben actualizarse en función de las pistas que haya en el tablero. Para esto se pide la segunda función, de propagación. El aspecto importante de estas funciones es la eficiencia, por ello el trabajo de implementación se va a realizar en tres lenguajes de programación distintos, *C, ARM y Thumb*, teniendo cada uno sus ventajas e inconvenientes, que se estudiarán en este documento.

# Objetivos

El objetivo del trabajo es conseguir que estas dos funciones tengan un coste en tiempo y en memoria mínimo dado que se trata de las funciones más críticas del sistema. Se parte de la base de que *ARM* nos permite un mayor control sobre las acciones del procesador, lo que a priori va a permitir una mayor optimización de la función. *Thumb* además hace uso de instrucciones de 16 bits, lo que es un factor muy importante en cuanto al consumo de memoria. *C* por otro lado tiene opciones de optimización a la hora de compilar que pueden resultar interesantes de comparar con *ARM.* A partir de estas tres premisas se debe analizar y evaluar el coste tanto en tiempo como en memoria de cada una de las opciones, teniendo en cuenta sus posibles mejoras gracias a optimizaciones. Al final de esta evaluación se deberá tomar una decisión en cuanto al lenguaje que se usará para el desarrollo de las funciones pedidas y justificar esa decisión con los resultados de las pruebas.

# Metodología

-Explicación esquema proyecto (C ARM THUMB)

-Medición numero instrucciones

-Calcular tiempo de ejecución de las 6 opciones

-Compilación optimizada de C

-Optimizaciones ARM y THUMB

-Función de validación

-Horas dedicadas y código comentado

# Resultados

# Conclusión

# Referencias

|  |  |
| --- | --- |
| [1] | Idealista.com, «Idealista — Casas y pisos, alquiler y venta. Anuncios gratis,» 8 Octubre 2016. [En línea]. Available: https://www.idealista.com/. |
| [2] | Fotocasa.es, «Fotocasa.es: Alquiler de pisos, compra y venta,» 8 Octubre 2016. [En línea]. Available: www.fotocasa.es/es. |
| [3] | Idealista.com, «Idealista,» 8 Octubre 2016. [En línea]. Available: https://st1.idealista.com/comunicacion/files/informe-de-precios/precios-q3-16.pdf. |
| [4] | Idealista.com, «Idealista.com,» 8 Octubre 2016. [En línea]. Available: https://www.idealista.com/particulares. |
| [5] | Fotocasa.es, «Fotocasa.es,» 8 Octubre 2016. [En línea]. Available: http://www.fotocasa.es/catalogo-productos\_\_fotocasa.aspx. |
| [6] | Wikipedia.org, «Wikipedia,» 8 Octubre 2016. [En línea]. Available: https://es.wikipedia.org/wiki/Idealista.com. |