

DESARROLLO DE UNA HERRAMIENTA PARA LA ELABORACIÓN DE INTERFACES GRÁFICAS PARA SISTEMAS DE MONITOREO Y CONTROL CON LA STM32F429I-DISC0 Y .NET MICRO FRAMEWORK.

DEVELOPMENT OF A TOOL FOR THE ELABORATION OF GRAPHIC INTERFACES FOR MONITORING AND CONTROL SYSTEMS WITH THE STM32F429I-DISC0 AND .NET MICRO FRAMEWORK.

C. M. Correa Torres & J. F. Valencia Gómez

1. Centro de Tecnología de la Manufactura Avanzada, Avda. SENA, Medellín, Colombia, cmcorrea@unal.edu.co
2. Centro de Tecnología de la Manufactura Avanzada, SENA, Medellín, Colombia jhonfredy@outlook.co

Recibido: octubre 15 de 2017 Aceptado: enero 20 de 2018

RESUMEN

El propósito de este trabajo es presentar herramientas de hardware y software para elaborar aplicaciones con sistemas embebidos que permiten desarrollos electrónicos con alta funcionalidad y desempeño; Microcontroladores de 32 bits con arquitectura Advanced Risc Machine (ARM) y entornos de desarrollo integrado (IDE) como .Net Micro framework que utiliza programación orientada a objetos. Basado en estas tecnologías, este artículo presenta una herramienta de software desarrollada en Visual Studio que permite diseñar interfaces gráficas de forma rápida y simple para implementar sistemas de control y monitoreo que pueden ser utilizados en entornos académicos e industriales

Palabras claves: Arquitectura ARM, Entornos de Desarrollo Integrado, Framework, Sistemas Embebidos

ABSTRACT

The purpose of this paper is to show hardware and software tools to develop applications with embedded systems that allow electronics developments with high functionality and performance; 32-bit microcontrollers with Advanced Risc Machine (ARM) architecture and integrated development environments (IDE) such as .Net Micro framework, that use object-oriented programming. Based on these technologies, this article presents a software tool developed in Visual Studio that allows designing graphical interfaces quickly and simply, to implement control and monitoring systems to be used in academic and industrial environments.

Keywords: ARM Architecture, Integrated Development Environments, Framework, Embedded Systems

1. INTRODUCCIÓN

La Evolución de las arquitecturas de hardware para sistemas embebidos ha permitido que estos puedan ahora utilizar mayores recursos de memoria y aumentar el procesamiento de instrucciones por unidad de tiempo, siendo capaces de realizar funciones cada vez más diversas y complejas manteniendo la característica de un bajo consumo de potencia. Los procesadores con arquitectura Advanced Risc Machine (ARM) han conquistado el mercado global de los productos electrónicos y se ven

integrados en dispositivos con los que interactuamos diariamente como teléfonos inteligentes, tabletas, electrodomésticos, entre otros. Aplicaciones en la industria biomédica, automotriz, desarrollo de hardware entre otros pueden encontrarse en [1][2][3].

Es por la anterior que los dispositivos electrónicos modernos pueden además de procesar múltiples tareas, incluir interfaces de visualización y comunicación que permiten una interacción directa y clara con el usuario final. Esta versatilidad y multifuncionalidad demandará por parte del desarrollador electrónico un amplio conocimiento en diferentes tecnologías para así lograr que sus productos satisfagan los requerimientos y necesidades actuales. Resulta entonces pertinente hacer una actualización de las herramientas para conseguir de esta forma ser realmente competitivos en el desarrollo electrónico.

3. MATERIALES Y MÉTODOS

Utilizando microcontroladores con arquitectura ARM, con el módulo STM32f429I-DISCO y la plataforma para desarrollo de software .Net micro framework, este trabajo presenta una herramienta de software desarrollada en Visual Studio que permite construir interfaces gráficas de forma rápida y simple para diseñar e implementar sistemas de monitoreo y control, que puede aplicarse en entornos tanto académicos como industriales.

El software se compone principalmente de elementos comunes, indicadores y controles que a través de una selección y arrastre pueden ser ubicados en el área de la pantalla para el diseño de la interfaz, una vez se compila la solución se genera el código en C# de los objetos gráficos seleccionados y este es asociado a un proyecto donde el usuario podrá interactuar con cada uno de ellos para darle funcionalidad específica. El software incluye también pestañas para la configuración de Hardware

4. RESULTADOS Y DISCUSIÓN

En la Figura 1 se presenta el entorno de la herramienta y un ejemplo de su implementación en la tarjeta física.

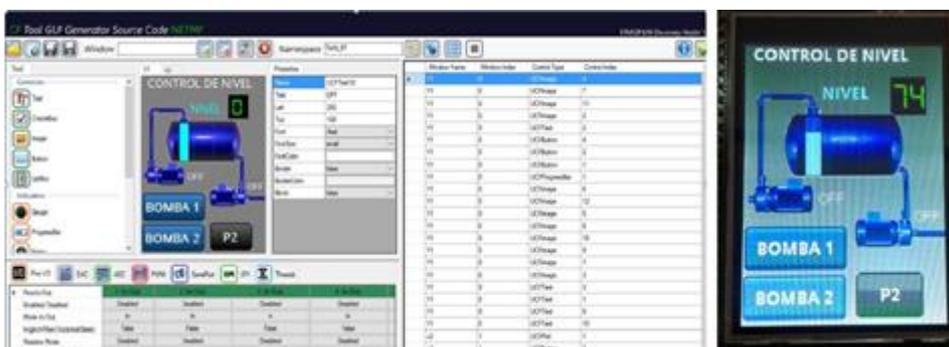


Figura 1. Interfaz de la herramienta desarrollada (Izquierda.), Implementación en la tarjeta STM32F429I Disc0 (Derecha.).

A continuación, se describe brevemente la funcionalidad de los elementos de los que se compone el software:

- Text: Escribe caracteres alfanuméricos ya sea de manera estática o dinámica durante la ejecución del programa.
- Chek Box: Permite realizar selección de un listado de items.
- Image: Carga y ubica imágenes en la pantalla.
- Button: Dibuja un control tipo botón para que al presionar realice una función determinada por el programador.
- List Box: Permite hacer listados para selección.

En la sección de indicadores se incluye:

- Gauge: Visualiza el valor de una variable en un instrumento indicador de aguja, permite escalizar en la función inicial.
- Progress Bar: Visualiza barras de progreso para indicar progreso o nivel a partir de una variable, puede configurarse color, tamaño, orientación y permite escalizar en la función inicial.
- Motor: Visualiza la activación y desactivación del eje de un motor y su dirección de giro puede configurarse a través de programación.
- Indicator: Grafica indicadores lumínicos tipo LED, permite configurar color encendido, apagado y tamaño
- Plot: Grafica estados de variables de forma continua en un plano X-Y, permite escalización en la función inicial.
- Keypad: Permite usar un teclado alfa-numérico.
- Timer: Presenta un temporizador que puede ser configurable por software o a través de la pantalla táctil una vez creado.
- Numeric: Genera un control numérico que puede configurarse a través de software o a través del táctil.

Cada control contiene campos para configurar sus características propias, y estos controles pueden ser usados múltiples veces en un proyecto. La Herramienta posee para configuración de Hardware:

- PINS I/O: Configura pines de entrada y salida.
- ADC: Configura una entrada analógica usando el conversor analógico digital.
- DAC: Configura una salida analógica usando el conversor digital analógico.

- PWM: Configura salidas moduladas por ancho de pulso (PWM).
- Serial Port: Configura comunicaciones seriales UART.

- SPI: Para Configura comunicaciones seriales tipo SPI.
- Thread: Configurar múltiples hilos de ejecución.

La Herramienta permite crear aplicaciones que incluyan múltiples ventanas en un mismo proyecto, presentando una lista de los elementos con sus respectivos índices y número de ventana donde fueron usados, permite además guardar los proyectos para ser posteriormente modificados.

Este trabajo tiene como propósito evidenciar que el trabajo con las nuevas herramientas disponibles de software y hardware permite que los desarrollos de sistemas electrónicos tengan una mayor versatilidad, funcionalidad además de disminuir los tiempos para su elaboración y costos. Es evidente que debemos apoyarnos en estas nuevas tecnologías ya que los dispositivos modernos demandan de mayores requerimientos. Cabe notar también que .Net micro framework facilita el manejo de las bibliotecas de funciones de algunas plataformas de Hardware modernas que incluyen arquitectura ARM, a través de lenguaje C# , este lenguaje común para desarrolladores de aplicaciones de escritorio acerca a estos al trabajo con hardware embebido y hace que expertos en hardware embebido aprovechen las ventajas de los lenguajes de más alto nivel. Debe considerarse que .Net Micro framework no tiene todas las bibliotecas de funciones para el número tan considerable de plataformas de hardware existentes, por esto no puede implementarse en cualquier sistema embebido, es por eso que en trabajos futuros, se buscará elaborar una herramienta que pueda ser usada para el desarrollo de interfaces gráficas que sea lo más transversal posible, esto se logrará desarrollándola con base en bibliotecas escritas en lenguajes más transversales como lo son por ejemplo el C ó C++.

5. REFERENCIAS BIBLIOGRAFICAS

- [1] Sandeep Raj, G.S.S. Praveen Chand, Kailash Chandra Ray, ARM-based arrhythmia beat monitoring system, Microprocessors and Microsystems Volume 39, Issue 7, October 2015, Pages 504-511.
- [2] Kristian Ismail, Aam Muharam, Mulia Pratama, Design of CAN Bus for Research Applications Purpose Hybrid Electric Vehicle Using ARM Microcontroller , Energy Procedia, Volume 68, April 2015, Pages 288-296.
- [3] Alejandro Castillo Atoche, Javier Vázquez Castillo, Jaime Ortegón Aguilar, Roberto Carrasco Alvarez, Jesús Sandoval Gío, Adrian Colli-Menchi, A high-accuracy photovoltaic emulator system using ARM processors, Solar Energy, Volume 120, October 2015, Pages 389-398.
- [4] JD Carlson, M Mittek, LC Pérez , Exploring the microsoft .NET micro framework for prototyping applied Wireless Sensor Networks, Electro/Information Technology (EIT), 2013, ieeexplore.ieee.org.
- [5] Marek Babiuch, .Net Micro Framework gadgeteer measurement applications development, Control Conference (ICCC), 2014 15th International Carpathian.