

Bienvenido al repo de desarrollo colaborativo - EMIC

Una iniciativa para compartir experiencias y mejorar la productividad de programadores de sistemas embebidos.

¿Que es EMIC?

EMIC es una plataforma que se utiliza, principalmente, para crear software embebido de calidad listo para integrar una solución determinada (entre otras cosas).

EMIC genera código a partir de conjunto de librerías previamente desarrollada por expertos, es decir que la calidad del programa generado depende directamente de la calidad de las funciones y drivers que se utilizaron como base para construir dicha aplicación.

Es por eso que nos referimos a EMIC como un gestor de código, nos ayuda de forma efectiva con las tareas de buscar ejemplos en Google, editarlos, adaptarlos a nuestra arquitectura, probarlos e integrarlos a nuestro programa.

El tiempo y la incertidumbre son factores críticos en todo desarrollo, y EMIC ayuda a reducir ambas de manera drástica (es posible crear una aplicación nueva, lista para ser usada en pocos minutos).

EMIC significa Electrónica Modular Inteligente Colaborativa. Está compuesto por: un conjunto de funciones y drivers, un editor de script simple e intuitivo que junto con un sistema de integración, puede crear código listo para compilar o ejecutar.

En un principio EMIC fué creado para crear programas en lenguaje C, pero en el último tiempo se puso a prueba en desarrollo de páginas web (HTML, CSS y JS). Con muy buenos resultados.

Además, EMIC es agnóstico respecto a la arquitectura, familia de microcontroladores y compiladores. Y se busca que las funciones y drivers aportados por los usuarios también lo sean.

Breve historia:

Todo comenzó cuándo un grupo de programadores intercambiaba código con el fin de mejorar la productividad y calidad de sus desarrollos.

Vieron que la eficiencia podría mejorar si todo el código desarrollado, cumplía con cierta normalización y buenas prácticas, y comenzaron a redactar las reglas para que las funciones puedan reutilizarse de forma automática, incluso sin la necesidad de comprender a fondo el funcionamiento del código desarrollado por otros programadores.

Con el correr del tiempo la cantidad de código generado fue creciendo y hubo que organizarlos en carpetas. Pudiendo separar, de esta manera, las funciones que utilizan recursos de bajo nivel y las que pertenecen a distintas capas de abstracción, la [lógica de negocio](#) y la descripción del hardware.

Más tarde se crea una aplicación de consola que siguiendo las indicaciones de un archivo de comandos secuenciales, automatiza la creación de código. Esto marcó un cambio fundamental, principalmente, porque cuando el código generado está basado en funciones que ya fueron testeadas y validadas en otras

aplicaciones, casi siempre el código generado funciona a la primera. Dependiendo solamente de la lógica de negocio. Los tiempos de desarrollo se redujeron de meses a días

Las reglas originales van evolucionando con el paso del tiempo, se agregan la definición de drivers con funciones no bloqueantes, máquinas de estado y bibliotecas genéricas.

Se crea un entorno de desarrollo en la nube, que incluye editor, generador de código y compilador, esta nueva herramienta permite generar la lógica de negocio de forma intuitiva y rápida, evitando errores de sintaxis. En esta etapa la plataforma permite utilizar hardware y firmware existente para crear equipos electrónicos en pocos minutos.

Posteriormente se diseña un protocolo para comunicar microcontroladores, esto permite modularizar los desarrollos, de forma que cada parte de un dispositivo electrónico puede recombinarse. Este sistema modular permite crear nuevos dispositivos en pocos minutos combinando módulos de hardware existentes. EMIC está pensado para que el desarrollador de sistemas embebidos viva una experiencia única. Logrando productos robustos y confiables en tiempo record.

Convocatoria:

El sistema sigue evolucionando, y para acelerar el crecimiento convocamos a todos los interesados a participar de esta gran experiencia de desarrollo colaborativo. La invitación va dirigida tanto a personas que tienen un camino recorrido en el mundo de los embebidos, como a quienes están dando sus primeros pasos, las contribuciones esperadas pueden ser: simples comentarios, validación de código, aporte de ideas, funciones, drivers, diseño de hardware etc.

Si bien, el aporte de más desarrolladores y usuarios que validen cada pieza de la plataforma es importante para la comunidad. No es necesario compartir todo el código creado, cada usuario puede trabajar en forma privada y luego decidir que quiere compartir con el resto.

Estado actual de EMIC

El proyecto EMIC es dinámico, semana tras semana se crean nuevas funciones, drivers completos y nuevo hardware, es por eso que las posibilidades de aplicaciones crecen.

Hoy EMIC cuenta con 50 módulos de hardware. Entre los que se encuentran los módulos de conectividad (WiFi, Bluetooth, LoRa WAN, radios con modulación FSK y LoRa, RS485, RS232); Módulos de display (siete segmentos y gráficos); sensores (temperatura, humedad, corriente, tensión, celdas de cargas, gases, etc); actuadores (relés, motores paso a paso, triacs, salidas PWM, etc); entradas y salidas digitales y analógicas; entre otros.

Respecto al firmware, podemos nombrar: comunicación SPI, I2C, Uart; Timers, salidas PWM, contadores, interrupciones, conversores AD, protocolos: MQTT, HTTP, JSON parser; controladores para ESP8266, sensores DHT22, DHT11, ADS1231, motores PXP, seven segment; MQ14

Funcionamiento

Como plataforma colaborativa, el propósito principal de EMIC es generar distintos tipos de documentos, como programas ejecutables por los módulos electrónicos, aplicaciones de teléfonos inteligentes o tablets y páginas webs.

Los ingredientes necesarios para generar estas aplicaciones y documentos se dividen en dos clases:

1. El material creado por los usuarios expertos en distintas áreas del conocimiento. Dentro de esta categoría están incluidos archivos en distintos formato estándar. Por ejemplo:

- Código C (*.c)
- Header (*.h)
- Web (*.html, *.css, *.js)
- Latex (*.tex)
- Markdown (*.md)

Estos archivos van acompañados de metatexto y comandos que el sistema interpreta para saber como tratarlos.

2. El *script* creado por cualquier usuario (incluyendo a los expertos). Es creado utilizando el [editor EMIC](#) edición se hace arrastrando y soltando los "recursos" disponible para cada módulo dentro de un entorno intuitivo. Donde además de código se puede editar páginas web de manera y pantallas gráficas de manera visual.

En otras palabras, en una aplicación EMIC se fusionan el conocimiento y la experiencia de desarrolladores e integradores y generan un resultado de alta calidad.

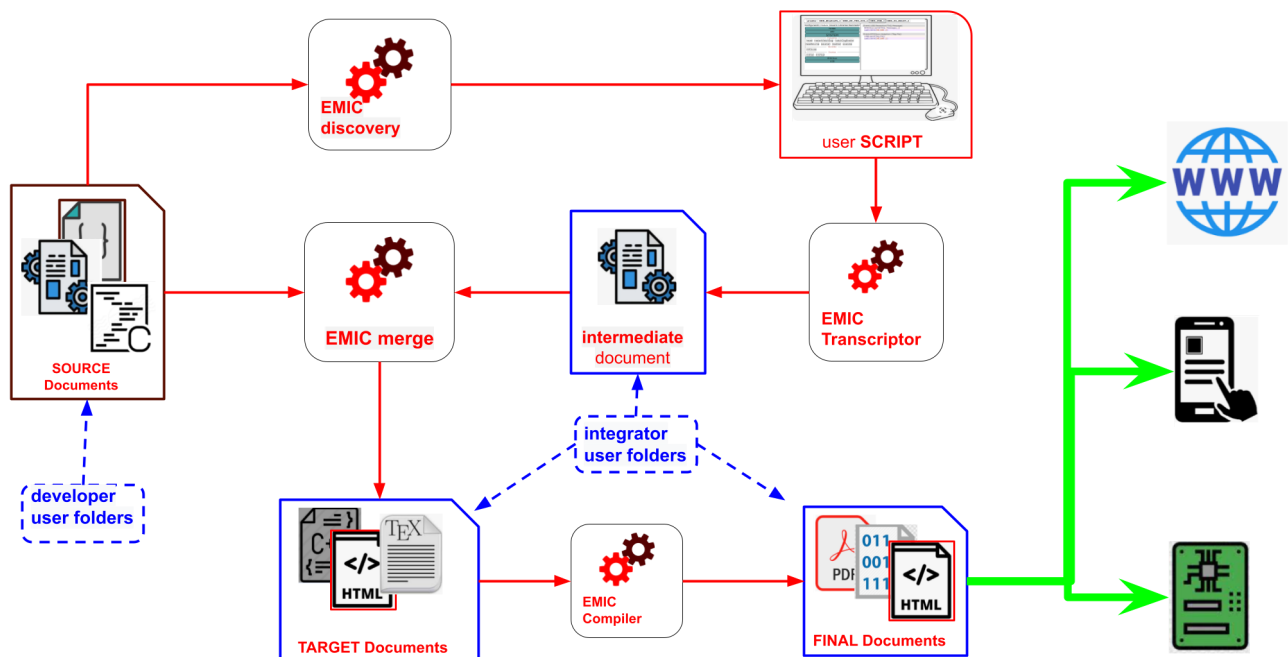


Diagrama simplificado del sistema EMIC

Vemos en el diagrama de funcionamiento, los distintos pasos para crear una aplicación web o un dispositivo electrónico. El sistema está formado por cuatro procesos que transforman distintos documentos de entrada en otros de salida (EMIC Discovery, EMIC Transcriptor, EMIC Merge, EMIC Compiler), tres almacenes de documentos (SOURCE Documents, Intermediate Documents, y TARGET Documents) y un editor de Script.

EMIC Discovery: este proceso es el encargado de transformar los documentos alojados en *Source Documents* en información utilizada por el editor, es decir, que genera una lista de recursos que luego pueden ser utilizados por quien edita el script.

EMIC Transcriptor: este proceso parte del script editado por el usuario y genera un documento que contiene la misma información pero en un formato diferente.

EMIC Merge: partiendo de los documentos originales y del script transcriptor, este proceso genera documentos que pertenecen a distintos formatos estándares como código C, latex, HTML, javascript, etc.

EMIC Compiler: en caso que los documentos generados por EMIC merge necesiten ser compilados, este proceso se encarga de ejecutar al compilador que corresponde dependiendo del tipo de documento generado.

Intermediate Documents: almacén donde residen los documentos generados por el transcriptor.











TARGET Documents: acá se almacena el primer resultado útil,

SOURCE Documents: es donde se almacenan los documentos creados por los desarrolladores del código fuente, este código va acompañado por información adicional, que describe el comportamiento dentro del sistema. Este es el único lugar donde los usuarios ingresan documentos en forma directa y donde los desarrolladores compartimos código con el resto. Dada la importancia de esta carpeta se dedicará una sección para explicarla en detalle su estructura.

Organización de archivos y carpetas en **SOURCE Documents:**

Esta carpeta contiene la información de la que el sistema parte para crear los documentos finales, y es además la que

EMIC

- |—  [_api](#)
- |—  [_drivers](#)
- |—  [_hal](#)
- |—  [_hard](#)
- |—  [_main](#)
- |—  [_modulos](#)
- |—  [_pcb](#)
- |—  [_templates](#)
- |—  [_util](#)
- |—  [_WEB](#)

En la carpeta [_API](#) están contenidos las funciones que tienen conexión con la lógica de negocio, es decir, puede tener funciones o variables que pueden ser accedidas desde el *script*.

Los recursos almacenados (funciones y variables, en el caso de la creación de código) en [_API](#), se describen en archivos de texto destinados a tal fin.

[drivers](#), [hal](#), [hard](#), [util](#), [pcb](#), [main](#)

En estas carpetas hay código que si bien cumplen distintas funciones (en correspondencia con su nombre) no recibe ningún tratamiento especial por parte del sistema. Para que un archivo almacenado en esta carpeta forme parte de una aplicación, deberá ser invocada por un recurso de nivel superior.

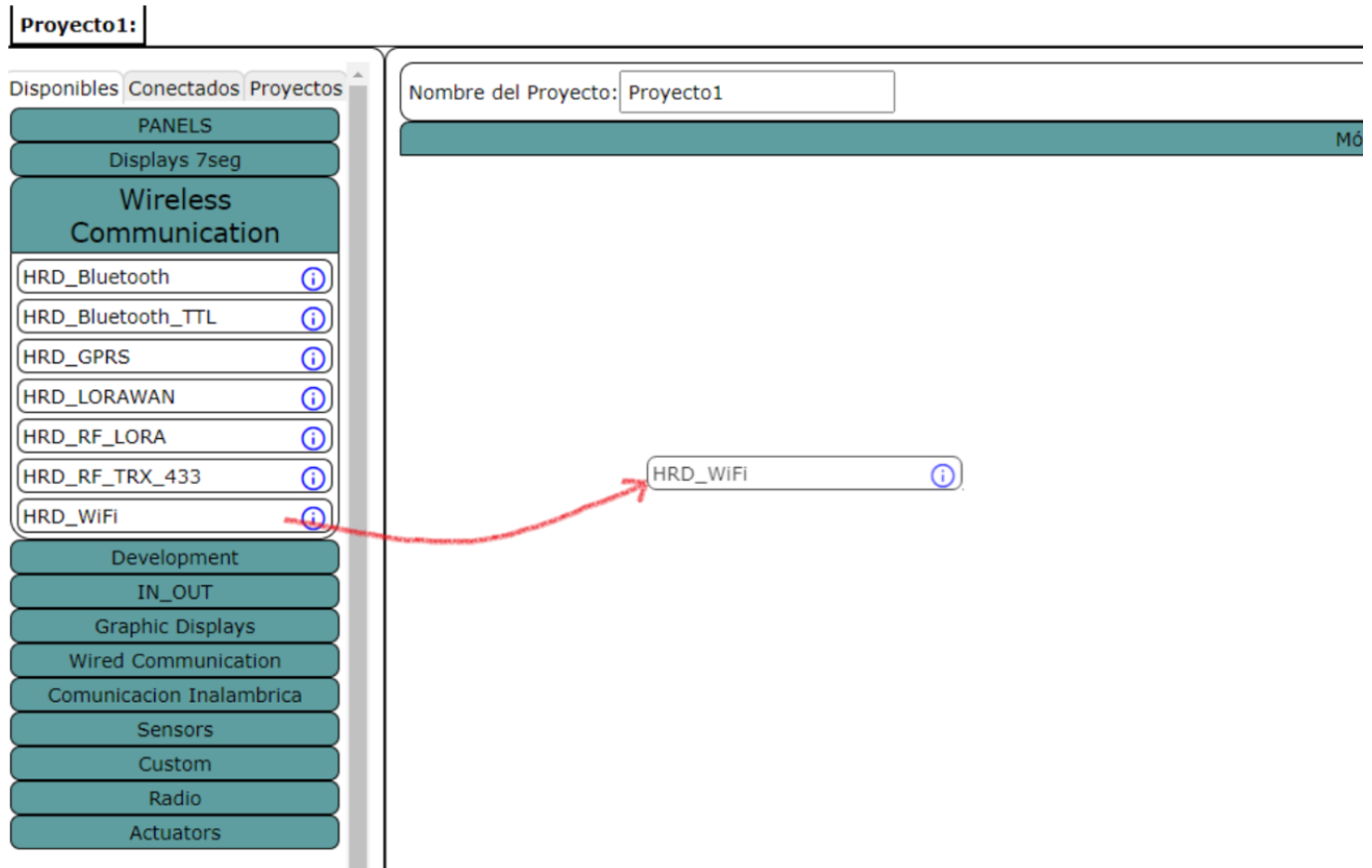
[_modulos](#)

En esta carpeta se almacena la descripción de los recursos de mayor nivel llamados módulos. Que a su vez están compuestos por otros recursos definidos en la carpeta _API

_WEB

Generación de Aplicaciones y documentos

En la etapa de integración, se crea el *script* mediante un proceso de edición utilizando la herramienta disponible en la web <https://community.emic.io>. Es decir el sistema toma el documento creado en el proceso de edición del *script* y los transforma en parámetros y código en un lenguaje de programación específico, como por ejemplo C, LaTeX, Javascript o HTML.



Esto es posible, gracias a la existencia previa de código fuente que fué creado utilizando herramientas para edición de código habituales, como IDEs y editores de texto (Visuals Studio, Notepad++, etc.) y luego suben los archivos generados a la plataforma.

[#1]: conjunto de algoritmos que realizan el trabajo que el usuario desea ejecutar