

Características técnicas del firmware de la Computadora Industrial Abierta Argentina

1. Introducción

En el presente documento se describen las características del firmware que poseerá la Computadora Industrial Abierta Argentina (CIAA). El firmware es el programa, software, que se ejecuta en la CPU del microcontrolador; comprende los módulos de código de programas, para realizar una aplicación determinada, e interactúa directamente con los periféricos internos y otros componentes físicos de la computadora industrial, dispositivos electrónicos, e interfaces de comunicación.

El contenido de este documento está sujeto a cambios que puedan producirse en el diseño de la arquitectura de hardware de la CIAA y en los requisitos finales para la implementación del PLC (ver 3. Documentos aplicables y de referencia) que surjan de la discusión pública y con las consultas a especialistas.

2. Equipo técnico a cargo del desarrollo de firmware de la CIAA

- **Responsable:** Ing. Gustavo Alessandrini (INTI, ORT)
- **Subresponsable:** Ing. Pablo Ridolfi (FIUBA, UTN-FRBA, UTN-FRH, Unitec Blue)

Colaboradores:

- Ing. Alejandro Celery (LSE-FIUBA)
- Ing. Leandro Francucci (Vortex)
- Ing. Gustavo Muro (FCEIyA-UNR)
- Ing. Javier Goglino

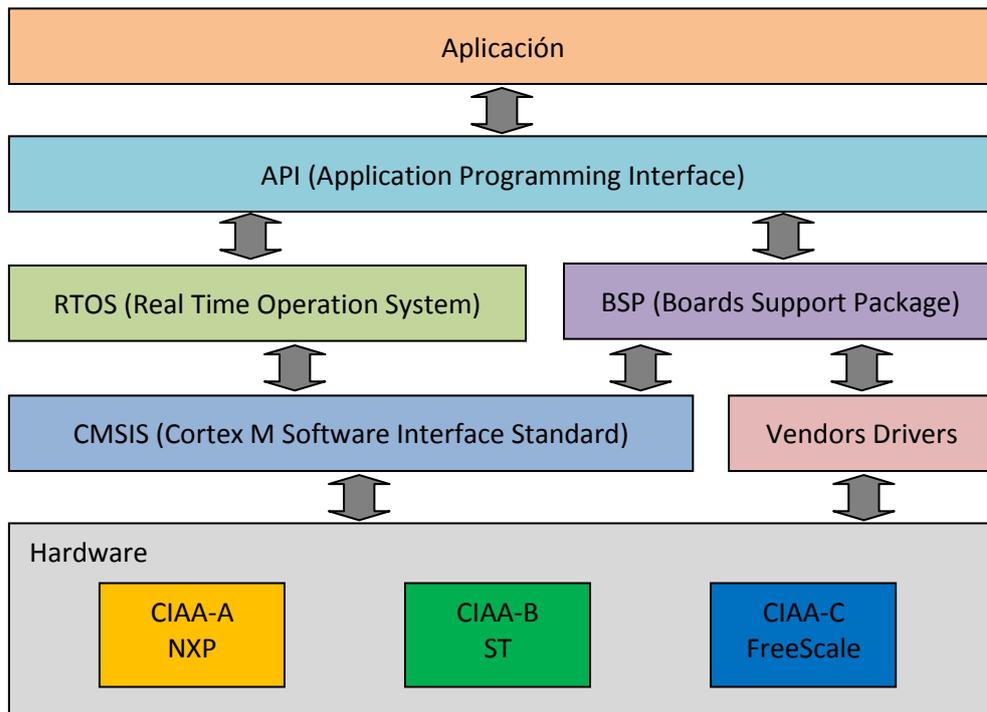
3. Documentos aplicables y de referencia

- Propuesta para el desarrollo de una Computadora Industrial Abierta Argentina:
<http://www.sase.com.ar/asociacion-civil-sistemas-embebidos/files/2013/11/Propuesta-Computadora-Industrial-Abierta-Argentina-ACSE-CADIEEL-v2.5.pdf>
- Características técnicas del Hardware de la Computadora Industrial Abierta Argentina:
<http://www.sase.com.ar/asociacion-civil-sistemas-embebidos/files/2013/11/CIAA-Hardware-v1.1.pdf>

4. Presentación conceptual del firmware

4.1. Esquema general

Con la disponibilidad de recursos que tienen los microcontroladores seleccionados y sus herramientas de desarrollo, es posible armar una arquitectura de código basadas en capas de abstracción, interfaces y bibliotecas.



Recorriendo de arriba hacia abajo esta estructura se encuentran las siguientes capas:

- **Aplicación:** Es el programa de usuario que debe cumplir un determinado propósito para el cual fue programada la CIAA. El objetivo del modelo de capas propuesto es lograr la independencia de la plataforma hardware utilizada. Esta capa o nivel se comunica con la capa inferior (API) para mantener la portabilidad buscada
- **API:** En la capa API (Application Programming Interface) se encuentran las bibliotecas de código, que también tienen como objetivo la independencia respecto del hardware utilizado. Estas bibliotecas contienen un conjunto de funciones que pueden utilizarse para diferentes proyectos, como por ejemplo:
 - Bibliotecas de funciones para manejo de protocolos
 - Bibliotecas de funciones para manejo de controladores (por ej: PID)
 - Bibliotecas de funciones para manejo de operaciones matemáticas
 - Bibliotecas de funciones de interfaz de usuario (teclados, displays, etc.)
- **RTOS:** El uso de un sistema operativo de tiempo real permite al desarrollador hacer aplicaciones en forma más sencilla, flexible y escalable. La plataforma RTOS seleccionada para este proyecto es FreeRTOS, el cual incluye una subcapa de portabilidad para lograr la independencia del microcontrolador.
- **BSP:** La capa Board Support Package, agrupa a las interfaces públicas de los periféricos que contiene el microcontrolador, tales como UARTs, Timers/counters, ADCs, DACs, etc. Las funciones que manejan estos periféricos cambian de acuerdo con el

microcontrolador utilizado, pero las interfaces públicas de estas funciones –definidas en este nivel- permanecen iguales. En general los RTOS ya cuentan con recursos necesarios o BSP propios para operar los microcontroladores, por este motivo es que ambas capas se encuentran en el mismo nivel. Las interfaces públicas mencionadas encapsulan el código para manejo de periféricos, como por ejemplo:

- Timers: Timer_Config(); Timer_Start(); Timer_Stop();
 - UARTs: UART_Config(); UART_putchar(); UART_getchar();
 - I2C: I2C_Config(); I2C_Send(); I2C_Receive();
 - ADCs: ADC_Config(); ADC_Read();
- **CMSIS:** Esta capa de abstracción es promovida por ARM para sus microcontroladores de la familia Cortex M. Esta capa es independiente de cada fabricante y es una biblioteca de código para acceder en forma normalizada al núcleo del Cortex M y a los periféricos incluidos en este
 - **Vendors Drivers:** Para la plataforma hardware seleccionada, cada fabricante suministra la capa de abstracción para el manejo de los periféricos que permite a los desarrolladores utilizar los periféricos sin acceder a la complejidad del manejo de los registros asociados, permitiendo de esta manera reducir el tiempo de desarrollo de las capas de bajo nivel del hardware

5. Requisitos del firmware

5.1. Requisitos para el desarrollo de las capas API y BSP

Por lo anteriormente expuesto, el desarrollo de firmware para la primera versión de la CIAA, estaría centrado en las capas API y BSP.

API	
Supervisión e integridad FW+HW	Control de integridad de firmware. Inicio seguro del dispositivo
Gestión RTOS	Interfaz para FreeRTOS
Stack TCP/IP	Interfaz para LwIP
Stack Modbus	Funciones para gestión del protocolo Modbus, de acuerdo con estándar: MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3 http://www.modbus.org
Real Time Clock	Funciones para gestión del RTC
Memoria flash externa	Manejo de la memoria flash externa <ul style="list-style-type: none"> ▪ SD ▪ Serial flash
Comunicación RS-485	Interfaz para control de la UART en modo RS-485
Lectura entradas analógicas	Manejo de entradas analógicas, formato, resolución
Lectura/escritura de entradas/salidas digitales	Manejo de E/S digitales, nivel de activación
Controladores PID	Gestión de las instancias de controladores PID

BSP	
Timers	Definición de las interfaces públicas
UARTs	Definición de las interfaces públicas
I2C	Definición de las interfaces públicas
SPI	Definición de las interfaces públicas
ADC	Definición de las interfaces públicas
GPIO	Definición de las interfaces públicas

5.2. Requisitos para la implementación del stack Modbus

Para la construcción del Stack Modbus se formulan los siguientes requisitos:

- 5.2.1. Implementar Modbus Slave: Crear la API de Modbus, de manera tal que mediante punteros a funciones se adquiriera la función de escritura y lectura que corresponda
- 5.2.2. Implementar Modbus independiente del enlace de comunicación (definir una interfaz común para escrituras y lecturas en Ethernet Sockets y UART), que sea compatible con:
- 5.2.3. Modbus RTU/ASCII
- 5.2.4. Modbus TCP/IP

5.3. Requisitos para el programa de ejecución del PLC

5.3.1. Capacidades del firmware

El firmware debe ser capaz de soportar las capacidades de memoria, periféricos, E/S y registros necesarios para implementar un programa que controle el comportamiento del PLC. Los recursos que se deben aplicar para realizar el PLC, se muestran en la siguiente tabla:

Memoria		
Memoria de programación	1	Memoria que almacena el código de mensajes o instrucciones que tiene que ejecutar la unidad lógica del PLC.
Datos de usuario		
Registros de entrada		Registro que recibe y almacena datos señalados desde una entrada conectada directamente
Registros de salida		Registro que almacena datos sobre el estado de una salida conectada directamente
Registros de datos		Registro de un PLC que almacenan bits de información
Registros con retención		Es el equivalente de los relés tradicionales para los

		PLC. Los registros de retención retienen datos que luego se ejecutarán en el procesador
Registros contadores		Grupo de instrucciones del PLC que cuenta, calcula o guarda un registro del número de veces que sucede algo
Registros temporizadores		Grupo de instrucciones del PLC que arrancan o paran automáticamente las máquinas y otros dispositivos cuando se ha excedido un período de tiempo.
Registros de configuración		
Contadores		Contadores de alta velocidad: contadores de 32 bits incrementales con auto reiniciación (reset), frecuencia máxima de respuesta de entrada 10 kHz
Temporizadores (módulo 32 bits)		Tipo I: temporizadores 0,1 s Tipo II: temporizadores 0,01 s Tipo III: temporizadores 0,001 s
Generadores PWM		
Interrupciones		
Interrupciones externas		
Comunicaciones		
Puertos serie	2	1 RS-232, 1 RS-485 half dúplex
Puertos USB	1	Firmware NO DISPONIBLE (Versión 1)
Puertos Ethernet	1	
Puertos CAN	1	Firmware NO DISPONIBLE (Versión 1)
Especificaciones I/O digitales		
Entradas locales	4	4 entradas optoacopladas
Salidas locales	8	4 salidas a relé, 4 a MOSFET
Especificaciones I/O analógicas		
Entradas locales	3	2 entradas analógicas de tensión Rango: 0-10 V Resolución: 12 bits 1 entrada analógica de corriente Rango: 4-20 mA Resolución: 12 bits
Expansión		
Bus de expansión (*)		
Capacidad de expansión (*)		
Controladores		
Módulos controladores PID		

(*) Dependiente de la variante CIAA (NXP, ST, Freescale), a definir en la etapa de desarrollo de arquitectura del hardware.

5.3.2. Carga del programa para ejecutar las tareas del PLC

- El programa que controla las acciones que realiza el PLC se recibe en la CIAA a través de la interfaz RS-485 o Ethernet.
- El programa, en lenguaje intermedio, será almacenado en la memoria disponible en la CIAA.
- El firmware se encargará de guardar el programa del PLC en la memoria SD, realizado las comprobaciones de integridad necesarias y tomando las precauciones de robustez y seguridad adecuadas.
- El firmware enviará el programa guardado en la SD a la memoria flash interna del correspondiente microcontrolador.

5.3.3. Ejecución del programa del PLC

El firmware interpretará el programa guardado en la memoria flash interna del microcontrolador y lo ejecutará paso a paso.

5.3.4. Protección del código del programa PLC

Por las características abiertas de la plataforma, tanto en hardware como en software y firmware, no es posible implementar un esquema de cifrado que permita resguardar la propiedad intelectual del programa de ejecución del PLC almacenado en la memoria SD de la CIAA. Por este motivo se deben poner a disponibilidad bibliotecas de código para los desarrolladores que deseen proteger su trabajo.

5.4. Requisitos para herramientas de desarrollo y depuración para los desarrolladores del firmware

5.4.1. IDEs, compiladores y depuradores

Free Scale	MK60FN1M0VLQ15
IDE	CodeWarrior (Eclipse IDE 4.2.1 (Juno) and CDT 8.1.1)
Compilador	CodeWarrior
Debugger	CodeWarrior
NXP	LPC4337JBD144
IDE	LPCXpresso
Compilador	Gcc
Debugger	LPCXpresso
ST	STM32F407ZGT6
IDE	Lanin SDK: Eclipse+OpenOCD+gcc (empaquetado para Windows y Linux)
Compilador	Gcc
Debugger	gdb+OpenOCD

5.4.2. Herramientas para análisis estático

- Resource Standard Metrics (versión limitada): <http://msquaredtechnologies.com/>

5.5. Requisitos de documentación para los desarrolladores del firmware (todas las plataformas)

5.5.1. Convenciones de codificación para el Lenguaje C

- Guía de estilo y convención de nombres:
 - C STYLE GUIDE-NASA Software Engineering Laboratory Series SEL-94-003
- Documentación de archivos de código fuente para soportar formato Doxygen
 - www.doxygen.org

5.6. Requisitos de documentación para el usuario

[Lista de la documentación para el usuario a ser suministrada como parte del producto]

5.6.1. Especificaciones técnicas para acompañar al producto

5.6.2. Manuales de uso (si no están cubiertos por la especificación técnica)

5.6.3. Manuales de instalación

5.7. Requisitos de mensajes de error y las trazas de los logs

5.8. Requisitos de mantenimiento y soporte

El mantenimiento y soporte se realizará mediante una página web administrada por ACSE

5.9. Requisitos para actualizaciones del firmware

El firmware de la CIAA se programa a través de la interfaz FT2232 del MPU + el software OpenOCD.