

Módulo 1

Proyecto CIAA: Utilizando la EDU-CIAA

Autores: Joaquín Rodríguez, Juan Pablo Vecchio Tutor: Ing. Marcelo Pistarelli Supervisor: Ing. José Ignacio Sosa Asesor: Ing. Gustavo Muro

| | | | <u>Contenido</u> |
|---|-----|--|------------------|
| 1 | Int | roducción | |
| 2 | La | Computadora Industrial Abierta Argentina (CIAA) | |
| 3 | Ins | talación y configuración del entorno IDE | 6 |
| | 3.1 | Instalación del IDE | 6 |
| | 3.2 | Error de Seguridad de Windows | 11 |
| | 3.3 | Fin de la instalación | |
| | 3.4 | OpenOCD | |
| | 3.5 | Corrección del Driver FTDI | |
| | 3.6 | Desinstalación | 17 |
| 4 | Co | nfiguración del entorno CIAA-IDE | |
| | 4.1 | Workspace | |
| 5 | Pri | meros Pasos | |
| | 5.1 | Proyecto "Blinking" | |
| | 5.2 | Indexación de cabeceras | |
| | 5.3 | Configuración del Makefile | |
| 6 | De | bug con Windows | |
| 7 | De | bug en placa EDU-CIAA y Entorno IDE | |
| | 7.1 | Configuración del entorno CIAA-IDE | |
| | 7.2 | Compilación del proyecto | |
| | 7.3 | Depuración sobre la placa: configuración de OpenOCD para Debug | |
| | 7.4 | Posible problema: "No reconocimiento" | |

1 Introducción

El siguiente documento pretende explicar todos los pasos necesarios para comenzar a desarrollar proyectos sobre la plataforma EDU-CIAA. Comenzaremos con todo lo referente a la instalación del entorno, la configuración del mismo para poder cargar el Firmware desarrollado para el proyecto, la carga de un ejemplo (llamado "Blinking", es decir, titilación de LEDS), y la descripción de pautas para desarrollar software desde cero. Parte de esta contribución utiliza la documentación existente en la Wiki del proyecto, pero se trata de realizar una explicación más profunda, de más bajo nivel, para aquéllos que no estén tan familiarizados con la programación Open-Source.

Por otra parte, este documento está exclusivamente desarrollado para la plataforma Windows. También se encuentra disponible un módulo exclusivamente hecho para Linux/Ubuntu, llamado *Módulo 2: Proyecto CIAA: Primeros pasos en Ubuntu / Linux.*

2 La Computadora Industrial Abierta Argentina (CIAA)

El *Proyecto CIAA* nació en el año 2013 como una iniciativa conjunta entre el sector académico y el industrial, representados por la *ACSE* y *CADIEEL*, respectivamente.

Los objetivos del Proyecto CIAA son:

- 1. Impulsar el desarrollo tecnológico nacional.
- 2. Darle visibilidad positiva a la electrónica argentina.
- 3. Generar cambios estructurales en la forma en la que se desarrollan y utilizan los conocimientos.

Todo esto en un marco de trabajo *libre, colaborativo y articulado* entre industria y academia.

Para lograr los objetivos, el primer paso fue articular el trabajo de decenas de Instituciones, Universidades, Empresas y Desarrolladores para diseñar la primera versión de la CIAA, denominada "*CIAA-NXP*" por estar basada en un procesador de la empresa *NXP Semiconductors*. Las Figura 1 y Figura 2 muestran el aspecto de esta placa, con indicaciones sintéticas de sus prestaciones.



Figura 1: Vista frontal de la placa CIAA y sus principales componentes



Figura 2: Vista posterior de la placa CIAA y sus principales componentes

La CIAA-NXP es la primera y única computadora del mundo que reúne dos cualidades:

- 1. Es *industrial*, ya que su diseño está preparado para las exigencias de confiabilidad, temperatura, vibraciones, ruido electromagnético, tensiones, cortocircuitos, etc., que demandan los productos y procesos industriales.
- 2. Es *abierta*, ya que toda la información sobre su diseño de hardware, firmware, software, etc. está libremente disponible en internet bajo la Licencia BSD, para que cualquiera la utilice como quiera.

Para avanzar aún más se desarrollaron versiones de la CIAA basadas en procesadores de otras marcas, como la *CIAA-FSL*, la *CIAA-INTEL*, la *CIAA-PIC*, etc.

En consecuencia, la CIAA además de ser la primera computadora industrial abierta, es también *la primera computadora realmente libre del mundo*, ya que su diseño no está atado a los procesadores de una determinada compañía, como ocurre con otras computadoras abiertas.

No hay que perder de vista que el *Proyecto CIAA* es mucho más que hardware, ya que también incluye un *entorno IDE* para su programación en *lenguaje C*, el *soporte de Linux*, un entorno de programación en *lenguaje tipo PLC*, el diseño de un gabinete y los primeros diseños de algunos de sus circuitos integrados.

Además, se diseñó una versión educativa de la plataforma, la *EDU-CIAA*, más simple y de menor costo, para lograr un impacto en la *enseñanza primaria, secundaria y universitaria*. La Figura 3 muestra una imagen de la misma.



Figura 3: Versión educativa de la Computadora Industrial (EDU-CIAA)

Las distintas versiones de la EDU-CIAA utilizan los mismos procesadores y programas que la CIAA, de modo que los estudiantes aprenden a resolver problemas reales, y pueden aplicar sus conocimientos y desarrollos en aplicaciones laborales y emprendimientos tecnológicos. Otro

detalle a tener en cuenta es que la EDU-CIAA es una versión simplificada de la CIAA. La misma cuenta con el mismo controlador y el mismo circuito integrado encargado de la depuración sobre la placa. Esto es lo mínimo que se necesita para conectar la placa a una computadora y programar sobre ella. Adicionalmente, cuenta con LEDs y pulsadores que pueden usarse para hacer algunas pruebas sencillas. El resto de los componentes se tendrán que agregar en forma de módulos, enchufados a los conectores ubicados en los laterales de la misma., lo que se justifica porque la EDU-CIAA podría usarse como una placa para hacer los primeros ensayos del código desarrollado, y luego trasladar todo el proyecto directamente a una placa CIAA, y realizar las pruebas finales con las interfaces industriales que ésta posee.

3 Instalación y configuración del entorno IDE

3.1 Instalación del IDE

El **IDE** (*inglés. Integrated Development Environment* = *Entorno de Desarrollo Integrado*) provee al desarrollador de Firmware (programador en C/C++) la posibilidad de trabajar en un ambiente amigable y *plug & play*.

Todas las herramientas necesarias para poder desarrollar aplicaciones en el Firmware colaborativo creado para el proyecto se instalarán en forma automática: al descargar el instalador, se provee una copia del Firmware y de los ejemplos, en su última versión. El paquete de instalación incluye:

- *Eclipse*: es el entorno base utilizado para desarrollo de aplicaciones, es decir, el *software IDE*.
- *PHP (Hypertext Pre-processor)* es un lenguaje de programación de uso general de código desde el lado del servidor, originalmente diseñado para el desarrollo de contenido dinámico. En este caso, se utiliza solamente en forma de *scripts* para poder generar algunos archivos del *Sistema Operativo OSEK*.
- *Cygwin*: es una consola que se ejecuta en Windows, de modo de emular la consola de comandos de Linux. Cuenta con todos los comandos, y el *compilador GCC*, propio del sistema operativo libre.

Para descargar el instalador, hay que dirigirse a la página oficial del proyecto, siguiendo este <u>link</u>. En ella, se encontrará el software IDE Suite. El mismo es compatible con las versiones de Windows XP, Windows Vista, Windows 7 y Windows 8, tanto en sus versiones de 32 como de 64 bits. Junto con la descarga, se incluye una copia del Firmware.

Cualquier inquietud, o inconveniente que surja respecto al software, puede dirigirse a la lista de mails: <u>ciaa-ide@googlegroups.com</u>.

El instalador provee todo el entorno necesario, con el cual se podrán instalar y configurar automáticamente de forma sencilla la gran mayoría de las herramientas necesarias para trabajar con la CIAA. Su uso es sumamente intuitivo: se recomienda no cambiar el directorio de instalación y, de ser necesario, elegir un nombre que NO contenga espacios. En las Figura 4 y Figura 5 se pueden ver las primeras dos ventanas correspondientes al instalador.



Figura 4: Arranque del instalador del software IDE



Figura 5: Acuerdo de licencia del instalador

| 🕞 Instalación de CIAA-Firmwa | re-IDE | |
|--|--|--|
| Selección de componentes Seleccione qué características | de CIAA-Firmware-IDE desea insta | lar. |
| Marque los componentes que o instalar. Presione Siguiente par | desee instalar y desmarque los com ra continuar. | ponentes que no desee |
| Seleccione los componentes a instalar: | Cygwin-Eclipse Firmware-0.4.0 Drivers Acceso directo en Menu Ii Acceso directo en Escritor | Descripción Sitúe el ratón encima de un componente para ver su descripción. |
| Espacio requerido: 1.2GB | 4 III > | |
| Nullsoft Install System v3.0b1 — | < <u>A</u> trás | Siguiente > Cancelar |

En la ventana siguiente, mostrada en la

Figura 6, se deben elegir los componentes que se desea instalar. Si usted no posee una EDU-CIAA, no es necesario que instale los drivers: si la adquiere en algún momento posterior, dado que los drivers se instalan junto con el IDE, los mismos quedarán en la carpeta de destino para su instalación en forma manual. Otra manera de instalar los controladores es ejecutar el instalador del CIAA-IDE Suite y tildar únicamente la opción *drivers* al momento de seleccionar los componentes a instalar.

| 🕞 Instalación de CIAA-Firmwa | re-IDE | |
|--|---|--|
| Selección de componentes Seleccione qué características | de CIAA-Firmware-IDE desea insta | lar. |
| Marque los componentes que c instalar. Presione Siguiente par | desee instalar y desmarque los com 'a continuar. | ponentes que no desee |
| Seleccione los componentes a instalar: | ✓ Cygwin-Eclipse ✓ Firmware-0.4.0 ✓ Firmware-0.4.0 ✓ Privers ✓ Acceso directo en Menu Iı ✓ Acceso directo en Escritor | Descripción Sitúe el ratón encima de un componente para ver su descripción. |
| Espacio requerido: 1.2GB | 4 | |
| Nullsoft Install System v3.0b1 — | < <u>A</u> trás | Siguiente > Cancelar |

Figura 6: Selección de componentes del instalador

A continuación deberá colocar la dirección (*path*) en donde desea instalar el entorno. La ventana correspondiente se muestra en la Figura 7. Como se indicó anteriormente, si se desea cambiarlo debe tenerse la precaución de no elegir una dirección donde los directorios posean espacios en sus nombres. Recomendamos no cambiar la unidad de instalación, pues en los siguientes pasos del documento se utilizarán direcciones que harán referencia a esta carpeta de instalación, y si se cambia, se deberán cambiar consecuentemente dichas direcciones

| 🕞 Instalación de CIAA-Firmware-ID | E | × |
|---|---|-------|
| Elegir lugar de instalación Elija el directorio para instalar CIAA- | Firmware-IDE. | VIIII |
| El programa de instalación instalará instalar en un directorio diferente, p Instalar para comenzar la instalaciór | CIAA-Firmware-IDE en el siguiente directorio. Para resione Examinar y seleccione otro directorio. Presione n. | 2 |
| Directorio de Destino | Examinar | |
| Espacio requerido: 1.2GB Espacio disponible: 16.8GB | | |
| Nullion Carlocali Dystelli V2,70 | < <u>A</u> trás Instalar Cance | lar |

Figura 7: Elección de la ruta de instalación

Hecho lo anterior, la instalación empieza automáticamente. En un momento aparecerá una ventana emergente, similar a la que se muestra en la Figura 8, en donde el programa pregunta si disponemos del hardware, pues para la instalación del driver es necesario conectar la placa. De no ser así, aún puede continuar la instalación haciendo click en '*No*'. Por el contrario, si disponemos de la EDU-CIAA, hacemos click en '*Yes*', y emergerá otra ventana, como se muestra en la Figura 9.



Figura 8: Instalación de los drivers: primera instancia



Figura 9: Instalación de drivers si se dispone del hardware

Concluida esta etapa, se procederá a la instalación de los drivers por defecto del fabricante FTDI para puerto virtual. Las imágenes correspondientes se muestran en las Figura 10, 11, 12 y 13.



Figura 10: Instalador de drivers FTDI



Figura 11: Instalación de drivers FTDI (cont.)



Figura 12: Instalación de drivers FTDI (cont.)



Figura 13: Instalación de drivers FTDI (fin)

El *Proyecto CIAA* está en continua expansión, y día a día se van realizando mejoras. Esto acarrea problemas, algunos de ellos (encontrados y solucionados) relacionados con los drivers: se trata de una falla en la comunicación a través del puerto virtual FTDI, que impide la correcta comunicación entre la placa y el entorno IDE. Su corrección debe efectuarse manualmente, fuera del instalador, y es posible la aparición de una ventana de error emergente como la que se muestra en la Figura 14.



Figura 14: Aviso de falla en la instalación de los drivers

Para la corrección del driver, consulte la sección "*Corrección del Driver FTDI*" cuando termine la instalación del Software IDE.

3.2 Error de Seguridad de Windows

Bajo algunos sistemas operativos, si se detecta que el driver no posee una firma válida, muestra un error Windows Security, como muestra en la Figura 15.



Figura 15: Windows Security Error

En este caso usted puede confiar en la procedencia y hacer clic en "*Install this driver software anyway*".

3.3 Fin de la instalación

Las últimas ventanas del instalador se muestran en las Figura 16 y 17:

| 🕞 Instalación de CIAA-Firmware-IDE | |
|--|------------------|
| Instalación Completada La instalación se ha completado correctamente. | |
| Completado | |
| Ver <u>d</u> etalles | |
| | |
| | |
| | |
| Nullsoft Install System v2.46 – Sigui | iente > Cancelar |

Figura 16: Fin de la copia de archivos y de la instalación de drivers



Figura 17: Fin de instalación del software IDE

Si no ocurrieron errores durante la instalación, usted dispondrá de un link en el escritorio que le permitirá abrir el CIAA-IDE y comenzar a trabajar con el Firmware.

3.4 OpenOCD

El hardware de la CIAA cuenta con un puerto USB para poder realizar la programación y depuración del programa en el microcontrolador: esto está implementado en el chip FTDI *FT2232H*.

La herramienta de código abierto *OpenOCD* (*On-Chip Debugger*) es la encargada de manejar el chip FT2232H a través del USB y a la vez todo lo referido al JTAG. Con esto, el *debugger* (GDB) utilizado en el IDE-Eclipse puede hacer su tarea simplemente conectándose al puerto 3333 (TCP) que el OpenOCD mantiene en escucha esperando la conexión.

Téngase en cuenta que el chip FT2232H posee 2 canales de comunicación independientes (A y B), pero ambos salen por el mismo puerto USB. Es decir, la computadora a la que está conectado verá 2 dispositivos distintos (en realidad uno compuesto): uno de estos dispositivos será el que conecta al JTAG manejado por OpenOCD (como se mencionó), y el otro se verá como un puerto virtual COM: este último puede servir principalmente para *debug*. Cada uno de estos dos dispositivos tendrá un driver asociado.

Lo primero es instalar los drivers por defecto del fabricante FTDI para puerto virtual (VCP). En el Administrador de Dispositivos deberían aparecer 2 nuevos puertos COM, tal como se muestra en la Figura 18.

3.5 Corrección del Driver FTDI

Un inconveniente que se nos puede presentar al momento de conectar nuestra placa con el sistema operativo, está vinculado con los drivers de la placa EDU-CIAA que se incluyen dentro del instalador descargado desde la página. El instalador incluye en la carpeta donde se instaló el software (Por defecto, *C:\CIAA\)* un programa que configura el driver del controlador serie, emulado por la placa, para que uno de ellos pueda ser utilizado como interfaz JTAG. Dicho programa se llama **Zadig_Win_7_2_1_1.exe**.

| Nombre | Fecha de modifica | Tipo | Tamaño |
|-----------------------------|-------------------|---------------------|----------|
| current | 08/05/2015 09:43 | Carneta de archivos | |
| Firmware | 08/05/2015 11-46 | Carpeta de archivos | |
| local-repo | 08/05/2015 09:43 | Carpeta de archivos | |
| lusbdriver | 08/05/2015 09:44 | Carpeta de archivos | |
| CIAA cygwin | 08/05/2015 09:46 | Acceso directo | 2.85 |
| driver_winusb_zadig_ft2232h | 14/04/2015 08:13 | Archivo PNG | 25 K8 |
| R Setup_Win_7_FTDI | 14/04/2015 11:38 | Aplicación | 2.188 KB |
| SetUsers | 09/04/2015 12:27 | Archivo por lotes | 2 K8 |
| 💮 uninstall | 08/05/2015 09:46 | Aplicación | 61 KB |
| 10 zadig_Win_7_2_1_1 | 14/04/2015 11:38 | Aplicación | 5.069 88 |

Figura 19 y Figura 20 muestran, respectivamente, el explorador de Windows situado en la carpeta del software y el entorno del programa corrector.

| Administrador de dispositivos | |
|--|--|
| Archivo Acción Ver Ayuda | |
| ♦ ⇒ 🖬 🗳 🖬 👰 🕼 🕵 🖏 | |
| Archivo Acción Ver Ayuda Archivo Acción Ver Ayuda Baterías Baterías Controladoras ATA/ATAPI IDE Controladoras de almacenamiento Controladoras de sonido y vídeo y dispositivos de juego Controladoras de sonido y vídeo y dispositivos de juego Dispositivos de interfaz de usuario (HID) Dispositivos de interfaz de usuario (HID) Dispositivos de istema Monitores Monitores Procesadores Procesadores Protes (COM y LPT) BT Port (COM12) BT Port (COM12) BT Port (COM13) BT Port (COM14) BT Port (COM20) BT Port (COM20) BT Port (COM20) BT Port (COM20) BT Port (COM20) BT Port (COM20) Prot (COM3) Prot (COM3) Puerto de impresora ECP (LPT1) | Propiedades: USB Serial Port (COM24) General Configuración de pueto Controlador Detalles USB Serial Port (COM24) Versión del controlador: Fecha del controlador: 26/08/2014 Versión del controlador: 212.0.0 Firmante digital: Microsoft Windows Hardware Compatibility Publisher Detalles del controlador: Actualizar controlador Actualizar controlador Revertir al controlador anterior Si después de actualizar el controlador instalado previamente. Deshabilitar |
| USB Serial Port (COM24) | Desinstalar Desinstalar el controlador (avanzado). |
| USB Serial Port (COM25) | Desinstalar Desinstalar el controlador (avanzado). |
| | Aceptar |

Figura 18: Administrador de dispositivos con la EDU-CIAA conectada

| Incluir en biblioteca 	Con | npartir con 🔹 Grabar | Nueva carpeta | • 🛯 🕷 |
|--|----------------------|---------------------|----------|
| Nombre | Fecha de modifica | . Tipo | Tamaño |
| k cygwin | 08/05/2015 09:43 | Carpeta de archivos | |
| 🕌 Firmware | 08/05/2015 11:46 | Carpeta de archivos | |
| 🎉 local-repo | 08/05/2015 09:43 | Carpeta de archivos | |
| 🎉 usbdriver | 08/05/2015 09:44 | Carpeta de archivos | |
| CIAA cygwin | 08/05/2015 09:46 | Acceso directo | 2 KB |
| 😿 driver_winusb_zadig_ft2232h | 14/04/2015 08:13 | Archivo PNG | 25 KB |
| Restup_Win_7_FTD1 | 14/04/2015 11:38 | Aplicación | 2.188 KB |
| SetUsers | 09/04/2015 12:27 | Archivo por lotes | 2 KB |
| 🛞 uninstall | 08/05/2015 09:46 | Aplicación | 61 KB |
| 10 zadig_Win_7_2_1_1 | 14/04/2015 11:38 | Aplicación | 5.069 KB |

Figura 19: Ventana del Explorador de Windows en la carpeta del Software IDE

| Driver | WinUSB (v6. 1. 7600. 16385) | Edit More Information WinUSB (libusb) |
|-----------------------------|-----------------------------|---|
| USB ID WCID ² | Install WCID Driver | libusb-win32 libusbK WinUSB (Microsoft) |

Figura 20: Entorno del software corrector Zadig para Windows

Para corregir los drivers, conectamos la placa a través del cable USB, hacemos click en *Options* – > *List All Devices*. Aparecerá una lista de dispositivos de comunicación relacionados al USB. Tenemos que buscar aquellos cuyos nombres tengan relación con el puerto serie (puede aparecer Dual RS232-HS, USB Serial Converter, o algo similar). Por lo general, aparecerán 2 con el mismo nombre, excepto que uno es *Interface 0* y el otro *Interface 1*, como se muestra en la Figura 21 (la lista de drivers que se muestra puede diferir, dependiendo de la computadora que se utilice).

| Zadig | |
|--|---------------------|
| <u>D</u> evice <u>O</u> ptions <u>H</u> elp | |
| Dual RS232-HS (Interface 0) | - Edit |
| Dual RS232-HS (Interface 0) Dual RS232-HS (Interface 1) USB Receiver (Interface 0) USB Receiver (Interface 1) Broadcom USH (Interface 0) Contacted SmartCard (Interface 1) Integrated Webcam (Interface 0) DW375 Bluetooth Module | ation) soft) |
| 8 devices found. | Zadig 2.1.1.674 |

Figura 21: Lista de dispositivos vinculados a USB

Para configurar el driver:

- a) seleccionar la *Interfase 0*
- b) elegir el "WinUSB v6.1"
- c) hacer click en el botón "Replace Driver".

La ventana ya configurada se muestra en la Figura 22.

| Dual PS | 232-HS/ | (Interface | 0) | | • === |
|---------|---------|------------|-------|--------------------------------|---------------------------------|
| Driver | FTDIBL | IS (v2.12 | .0.0) | 2 WinUSB (v6. 1, 7600, 163) | 85) More Information |
| USB ID | 0403 | 6010 | 00 | | WinUSB (libusb) libusb-win32 |
| WCID 2 | × | | | 3 Replace Driver | ▼ libusbK |

Figura 22: Configuración del Zadig para el reemplazo del driver

3.6 Desinstalación

Si se instaló el Software de CIAA-IDE y luego se desea desinstalarlo, se debe tener especial cuidado en quitar cualquier contenido que se quiera conservar de la carpeta C:\CIAA, o el directorio de instalación elegido. Esto se debe a que el desinstalador del Software CIAA-IDE elimina el directorio y todo su contenido.

4 Configuración del entorno CIAA-IDE

4.1 Workspace

Al iniciar el entorno CIAA-IDE, éste solicita que seleccione una carpeta de trabajo, como se muestra en la Figura 23:

| G Workspace Launcher | × |
|--|------------------|
| Select a workspace | |
| Eclipse stores your projects in a folder called a workspace. Choose a workspace folder to use for this session. | |
| Workspace: C:\CIAA\workspace | ▼ <u>B</u> rowse |
| Use this as the default and do not ask again | OK Cancel |

Figura 23: Ventana inicial para ubicar el espacio de trabajo (Workspace)

Todo lo relativo al proyecto en que se esté trabajando (configuraciones, variables globales del entorno y demás) se guarda en esta carpeta. Si siempre se va a utilizar la misma ubicación, se puede tildar la opción "*Use this as the default and do not ask again*". En cualquier otro caso, cada vez que se abra el entorno, se preguntará por esta ubicación. En particular puede elegirse dentro del directorio de instalación del CIAA-IDE sin ningún problema.

5 Primeros Pasos

5.1 Proyecto "Blinking"

Para el usuario sin experiencia con esta plataforma de desarrollo, se aconseja abrir el proyecto del Firmware, para luego compilar y ejecutar un proyecto de ejemplo, llamado *Blinking*. Para ello debemos ir al menú '*File→New→Makefile Project with Existing Code*', como se muestra en la Figura 24.

Proyecto CIAA: Utilizando la EDU-CIAA

| 0 | C/C++ - Eclipse | | | |
|----------|--------------------------------------|--------------------------|--|-----------------|
| File | Edit Source Refactor Navigate Search | Projec | ct Run Window Help | |
| | New Alt+Shift+N • | C.+ | Makefile Project with Existing Code | T |
| | Open File | C+ | C++ Project | Ì. |
| | Close Ctrl+W | | C Project Project | I |
| | Close All Ctrl+Shift+W | | Convertes - C/Con Antonio In Project | Г |
| | Save Ctrl+S | C++ | Convert to a C/C++ Autotoois Project Convert to a C/C++ Project (Adds C/C++ Nature) | L |
| u. | Save All Ctrl+Shift+S Revert | | Source Folder Folder Source File | |
| 1 | Move Rename F2 Refresh F5 | C h ² C | Header File File from Template Class | 5 1 t |
| | Convert Line Delimiters To | | Task | L |
| Ð | Print Ctrl+P | | Other Ctrl+N | N |
| | Switch Workspace Restart | Γ | V | ellc |
| 24 24 | Import Export | | | |
| | Properties Alt+Enter | | | |
| _ | Exit | | | |

Figura 24: Comienzo de la carga del código existente

En el caso de no haber instalado el Firmware junto al entorno CIAA-IDE, hay 2 opciones:

- se puede descargar desde la web, siguiendo las instrucciones en la sección Código Fuente (utilizando el directorio de repositorios, cuyo enlace se encuentra <u>aquí</u>), o bien
- se puede volver a ejecutar el Instalador del Software-IDE de la CIAA y, en la ventana de componentes, seleccionar únicamente el ítem Firmware.

Seguidamente aparecerá una ventana (Figura 25) en donde se debe indicar el proyecto que se va a cargar: en este caso elegiremos la carpeta Firmware que, si usamos los directorios por defecto, se encuentra en la ubicación:

C:\CIAA\Firmware

En la ventana *Toolchain for Indexer Settings* seleccionaremos la opción *<none>*, lo cual dejará las opciones por defecto, configuradas en el Makefile.

| New Project | |
|---|--------|
| Import Existing Code Create a new Makefile project from existing code in that same directory | |
| Project Name | |
| Firmware | |
| Existing Code Location | |
| C:\CIAA\Firmware | Browse |
| Languages Image: Contract of the second s | |
| <none> Cross ARM GCC Cross GCC Cygwin GCC GNU Autotools Toolchain Microsoft Visual C++</none> | |
| Show only available toolchains that support this platform | |
| ? <u>Einish</u> | Cancel |

Figura 25: Creación de un nuevo proyecto usando un código existente

Una vez creado el proyecto, cerramos la pestaña *Welcome* con la que inicia Eclipse, y nos encontraremos con un entorno como el que se presenta en la Figura 26.

| le Edit Source Refactor Navigate Search Prov | ect Run Window Help | | | | | | | 0.9 |
|---|--|-------------------------------|-------------|----------|------|-----|---|----------|
| | G | | a • • • • • | - * | | | Quick Access | 1 1 C/C- |
| p Project Explorer II 📄 🗞 😰 🗢 🖙 Ø Frimware | | | | | | ~ 0 | BE Ou 12 ® Ma An outline is not available. | p . |
| | | | | | | | | |
| | Problems 22 2 Texts | Console III Properties | | | | | | 8 |
| | Problems 12 2 Leds 12 0 items Description | Console E Properties Resource | Path | Location | Туре | | | \$ v = t |
| | Problems 12 2 Taxts 12 0 items Description | Console Properties | Path | Location | Туре | | | \$ |
| | Problems 22 2 Tasks C 0 items Description | Console Properties Resource | Path | Location | Туре | | | ð í |

Figura 26: Entorno IDE-CIAA con el Firmware cargado

5.2 Indexación de cabeceras

Para la compilación del proyecto es necesario que el software IDE sepa dónde encontrar las cabeceras del estándar *POSIX*.

POSIX (Portable Operating System Interface) es un conjunto de interfaces estándar para sistemas operativos basadas en UNIX. Esta estandarización fue necesaria para que los distintos fabricantes de computadoras o desarrolladores de programas pudieran desarrollar sus aplicaciones independientemente de la plataforma en la cual iba a correr.

Para poder indexar esas definiciones se deben agregar los archivos *Includes* del *GCC* (*GNU Compiler Collection*). Esto se efectúa sobre la pestaña *Path and Symbols* \rightarrow *Includes*, en una ventana similar a la que muestra la Figura 27.

| Properties for Firmware | | | |
|---|----------------------|--|-------------------|
| type filter text | Paths and Symbols | | ← ▼ ⇒ ▼ ▼ |
| ▷ Resource Builders ▷ C/C++ Build ▲ C/C++ General ▷ Code Analysis | Configuration: Defau | it [Active] | Configurations |
| Documentation | 🕒 Includes 🗰 Sym | ibols 🛋 Libraries 🖻 Library Paths 😕 Source Location 🍃 Ou | utput Locatic 🔹 🔸 |
| Formatter | Languages | Include directories | Add |
| Indexer Language Mappings | Assembly GNU C | | Edit |
| Paths and Symbols Preprocessor Include Pa | GNU C++ | | Delete |
| Profiling Categories | | | Export |
| Project References Run/Debug Settings | | | Move Up |
| WikiText | () "Preprocessor Inc | lude Paths, Macros etc." property page may define additional entries | WOVE DOWN |
| | Show built-in value | 25 | |
| 4 111 | L | Restore Defaults | Apply |
| ? | | ОК | Cancel |

Figura 27: Indexación de las definiciones POSIX

En esta ventana seleccionar 'GNU C' ó 'GNU C++' según las POSIX que queramos agregar, y luego presionar el botón 'Add'. Emergerá la ventana mostrada en la Figura 28, en la cual se debe hacer click en el botón 'File System...' y luego buscar la carpeta correspondiente a agregar.

| Directory: | |
|---|-------------------------|
| Add to all configurations Add to all languages E a workspace path | Variables Workspace, |
| | File system. |

Figura 28: Indexar POSIX

Los Includes que deben configurarse, según el lenguaje (Language) elegido, son los siguientes:

Language = 'GNU C'

• <dIDE>\cygwin\usr\include. Si se aplican los directorios por defecto, éste quedaría C:\CIAA\cygwin\usr\include • <dIDE>\cygwin\lib\gcc\i686-pc-cygwin\4.9.2\include (por defecto: C:\CIAA\cygwin\lib\gcc\i686-pc-cygwin\4.9.2\include)

(dIDE = directorio de instalación del software-IDE: por defecto es C:\CIAA)

Language = 'GNU C++'

- <dIDE>\cygwin\usr\include
 (por defecto: C:\CIAA\cygwin\usr\include)
- <dIDE>\cygwin\lib\gcc\i686-pc-cygwin\4.9.2\include\c++ (por defecto: C:\CIAA\cygwin\lib\gcc\i686-pc-cygwin\4.9.2\include\c++)

(dIDE = directorio de instalación del software-IDE: por defecto es *C*:*CIAA*)

Finalizada la selección, los Includes quedarán de forma similar a la mostrada en la Figura 29.

| permet con | Paths and Symbols | | (⇒ ▼ ⇔ |
|--|---|--|---|
| Recourse | Facilis and Symbols | | <u>√</u> · ¬∕ |
| Builders | | | |
| C/C++ Build | Configuration: Defi | ault [Active] | ge Configuratio |
| C/C++ General | - | | |
| Code Analysis | | | |
| Documentation | 🕒 Includes # 🛇 | ymbols 🛋 Libraries 👼 Library Paths 🖓 Source Location 🏟 | Output Locatio |
| File Types | , , , , , , , , , , , , , , , , , , | | output cocuts |
| Formatter | Languages | Include directories | Add |
| Indexer | Assembly | (C)(IAA)(cygwin)ust)include | |
| Language Mappings | GNU C | C:\CIAA\cygwin\usi\uncideC | Edit |
| Paths and Symbols | GNU C++ | e (churic)guin (ib)gee (bob pe c)guin (ib)z (include (e) | Delete |
| Preprocessor Include Pa | | | beren |
| Profiling Categories | | | Expor |
| Linux Tools Path | | | |
| Project References | | | Movel |
| Run/Debug Settings | | | INOVE C |
| Task Repository | | | Move Do |
| WikiText | Preprocessor I | Include Paths, Macros etc." property page may define additional entrie | is |
| | Show built-in va | duar | |
| | Show built-in va | | |
| | import Setting | gs 🦗 Export Settings | |
| | | | |
| · | | Restore Defau | lts App |
| • | | | |
| 3 | | | |
| * / | | | Contract |
| Properties for Firmware | Paths and Symbols | <u>.</u> | |
| Properties for Firmware /pe filter text | Paths and Symbols | ok. | Cance Cance |
| Properties for Firmware /pe filter text > Resource Builders | Paths and Symbols | , | Cance |
| Properties for Firmware /pe filter text > Resource Builders > C/C++ Build | Paths and Symbols | OK s ault [Active] → Mana | Cance |
| Properties for Firmware ype filter text > Resource Builders > C/C++ Build = C/C++ General | Paths and Symbols Configuration: Defa | oK s ault [Active] •) [Mana | Cance Cance Cance Cance Cance |
| Properties for Firmware ype filter text Builders b C/C++ Build a C/C++ General b Code Analysis | Paths and Symbols | oK s ault [Active] | Cance |
| Properties for Firmware per filter text b Resource Builders b (C++ Build a C/C++ General b Code Analysis Documentation | Paths and Symbols Configuration: Defin | OK | Cance |
| Properties for Firmware Properties for Firmware pre filter text Resource Builders C/C++ Build C/C++ General C/C++ General C/C++ General C/C++ General C/C++ General C/C++ General C/ | Paths and Symbols Configuration: Defi Configuration: Defi Definition: Definition Symposium (Statement Statement Stat | oK ault [Active] → Mana ymbols ➡ Libraries ➡ Library Paths ఊ Source Location ఊ | Cance |
| Properties for Firmware ype filter text > Resource Builders > C/C++ Build 4 C/C++ Build > Code Analysis Documentation File Types Formatter | Paths and Symbols Configuration: Defa | oK ault [Active] | Cance |
| Properties for Firmware ype filter text b Resource Builders b (/C++ Build a C/C++ General b (Cde Analysis Documentation File Types Formatter Indexer | Paths and Symbols Configuration: Defa Configuration: Defa Includes # Sy Languages Assembly | oK ault [Active] mbols Libraries Library Paths Source Location Source Location CALANCyminius/include | Cance |
| Properties for Firmware ype filter text > Resource Builders > C/C++ Build 4 C/C++ General > Code Analysis Documentation File Types Formatter Indexer Language Mappings | Paths and Symbols Configuration: Defi Includes # Sy Languages Assembly GNU C | ok ault [Active] ymbols Mana ymbols Libraries Library Paths Source Location Include directories CxCIAAcygwinVusYInclude CxCIAAcygwinVusYInclude CxCIAAcygwinVusYInclude CxCIAAcygwinVusYInclude | Cance Cance Cance Configuration Configuration Contput Location Add Edit |
| Properties for Firmware ype filter text > Resource Builders > C/C++ Build > C/C++ Build > Code Analysis Documentation File Types Formatter Indexer Indexer Indexer Analysis Paths and Symbols | Paths and Symbols Configuration: Definition Configuration: Definition: Definition: Definition: Definition: Definition: Def | OK ault [Active] Juntools Libraries Claray Paths Concentration Clarateries Include directories ChCIAA kygwin \usir \include ChCIAA kygwin \usir \include | Cance |
| Properties for Firmware ype filter text > Resource Builders > (Cc+ Build > (Cc+ General > Code Analysis Documentation File Types Formatter Indexer Language Mappings Paths and Symbols Preprocessor Include P; Preprocessor Include P; | Paths and Symbols Configuration: Defa Configuration: Defa Includes # Sy Languages Assembly GNU C GNU C++ | ok ault [Active] Mana ymbols Libraries Library Paths Source Location Childe directories CAAAkygwin'Uis'uiclude CAAAkygwin'Uis'uiclude | Cance |
| Properties for Firmware pre filter text Besource Builders C(C++ Build C(C++ General C(C++ Genera | Paths and Symbols Configuration: Defi Implementation: Definition Implementation Configuration: Definition Configuration: Definition Configuration Configur | ok ault [Active] Vimbols Libraries (B) Library Paths (B) Source Location (B) Include directories (CCLAAAcygwin/UB/gcc/i686-pc-cygwin/4.9.2/include | Cance C |
| Properties for Firmware pe filter text > Resource Builders > (C++ Build > (C++ General > Code Analysis Documentation File Types Formatter Indexer Language Mappings Paths and Symbols Preprocessor Include Pi Profiling Categories Linux Tools Path Document Volter | Paths and Symbols Configuration: Defa Configuration: Defa Includes # Symbol Languages Assembly GNU C++ | oK ault [Active] Mana ymbols Libraries Library Paths Source Location Childude directories Childude directories Childude directories Childude directories Childude directories Childude directories Childude directories | Cance |
| Properties for Firmware rpe filter text > Resource Builders > C/C++ Build > Code Analysis Documentation File Types Formatter Indexer Language Mappings Paths and Symbols Preprocessor Include Pi Profiling Categories Linux Tools Path Project References PundDebus Cestinger | Paths and Symbols Configuration: Defa Configuration: Defa Languages Assembly GNU C GNU C++ | ault [Active] multi [Active] multi [Active] multi [Active] multi [Active] multi [Active] multiplice [Active | Cance |
| Properties for Firmware ype filter text b Resource Builders b C/c++ Build c C/c++ General b Code Analysis Documentation File Types Formatter Indexer Language Mappings Paths and Symbols Preprocessor Include Pi Profect References Run/Debug Settings Tark Banocitons | Paths and Symbols | oK ault [Active] Vimbols ALibraries Paths Concentration Content of the second | Cance |
| Properties for Firmware prefilter text Resource Builders C/C++ Build C/C++ General C/C++ General C/C++ General C/C++ General C/C++ General Code Analysis Documentation File Types Formatter Indexer Language Mappings Paths and Symbols Preprocessor Include Pi Profiling Categories Linux Tools Path Project References Run/Debug Settings Task Repository WithTert | Paths and Symbols Configuration: Defa Configuration: Defa Includes # 5 Languages Assembly GNU C+ GNU C++ | oK ault [Active] Mana ymbols Libraries Library Paths Source Location Childe directories CCICIAA/cygwin/UB/gcc/686-pc-cygwin/4.9.2/include | Cance |
| Properties for Firmware ype filter text > Resource Builders > Code Analysis Documentation File Types Formatter Indexer Language Mappings Paths and Symbols Preprocessor Include Pr Profiling Categories Innux Tools Path Project References Run/Debug Settings > Task Repository WikiText | Paths and Symbols | ault [Active] Mana ymbols Libraries Library Paths Source Location Criticude directories Criticude directories | Cance |
| Properties for Firmware ype filter text > Resource Builders > C/C++ Build > C/C++ Build > C/C++ Build > Code Analysis Documentation File Types Formatter Indexer Indexer Indexer Indexer Profite References Run/Debug Settings > Task Repository WikiText | Paths and Symbols Configuration: Definition Configuration: Definition: Definitio: Definition: Definitio: Definitio: Definition: Definition | ault [Active] Mana ymbols Libraries Mana ymbols Library Paths Source Location Include directories C:\CIAA\cygwin\usivinchude Source Location Include directories Source Location Source Location Source Location Source Location Source Location Include directories Source Location Source Location Include directories Source Location Source Location Include Paths, Macros etc." property page may define additional entrie Include Paths, Macros etc." property page may define additional entrie | Cance |
| Properties for Firmware ype filter text Resource Builders C/C++ Build C/C++ Genral C/C++ Genra | Paths and Symbols Configuration: Definition Configuration: Definition Configuration: Definition Configuration: Definition Configuration: Definition Configuration: Configuration Configuration: Configuration Configuration Configuration: Configuration Configuration: Configuration | ault [Active] Mana ault [Active] Mana ymbols Libraries Mana include directories Source Location Source Location include directories CACLAAkcygwinViBblgccVi686-pc-cygwin\4.9.2Vinclude include Paths, Macros etc." property page may define additional entrie lues Source Source Source | Cance |
| Properties for Firmware pe filter text > Resource Builders > C/C++ Build 2 C/C++ Build > Code Analysis Documentation File Types Formatter Indexer Language Mappings Paths and Symbols Preprocesor Include Pi Profiling Categories Linux Tools Path Project References Run/Debug Settings > Task Reportory WikiText | Paths and Symbols Configuration: Defi Implication: Definition Implication GNU Ci GNU C++ | ault [Active] Mana ymbols Mina ymbols Libraries Include directories Source Location Include directories CACIAAkcygwin/Usi/Include Include directories CACIAAkcygwin/Usi/Include Include Paths, Macros etc." property page may define additional entrie Include Paths, Macros etc." property page may define additional entrie Include Paths, Macros etc." property page may define additional entrie Include Paths, Macros etc." property page may define additional entrie Include Paths, Macros etc." property page may define additional entrie Include Paths, Macros etc." property page may define additional entrie Include Paths, Macros etc." property page may define additional entrie Include Paths, Macros etc." property page may define additional entrie Include Paths, Macros etc." property page may define additional entrie Include Paths, Macros etc." property page may define additional entrie Include Paths, Macros etc." property page may define additional entrie Include Paths, Macros etc." property page may define additional entrie Include Paths, Macros etc." property page may define additional entrie Include Paths, Macros etc." property page may define additional entrie | Cance |
| Properties for Firmware pe filter text b Resource Builders c (C++ General b) Code Analysis Documentation File Types Formatter Indexer Language Mappings Paths and Symbols Preprocessor Include Pi Profiling Categories Linux Tools Path Project References Run/Debug Settings Tack Repository WikiText | Paths and Symbols | ok ault [Active] ymbols mail Libraries bibrary Paths Childred directories bibrary Childred | Cance |
| Properties for Firmware ype filter text | Paths and Symbols Configuration: Defe Configuration: Defe Languages Assembly GNU C GNU C++ O "Preprocessor I Show built-in val Configuration: Defense Configuration: | ok ault [Active] ymbols millitraries billitraries | Cance |

Figura 29: Fin de la configuración de Includes

5.3 Configuración del Makefile

Los archivos *Makefile* son archivos de texto escritos con una sintaxis predeterminada. Junto con la utilidad *'Make'*, permiten construir el software desde sus archivos-fuente, en el sentido de organizar el código, su compilación y enlace (*link*) correcto.

El *Proyecto CIAA* tiene su propio Makefile, por lo que se debe indicarle al IDE cómo manejarse con él: de lo contrario generaría un Makefile automáticamente, lo cual nos traería muchos dolores de cabeza.

Cada vez que hacemos un comando clean, estamos borrando los archivos objeto generados previamente, pero antes de poder volver a compilar, necesitamos que esté previamente procesado el código PHP correspondiente al sistema operativo **RTOS-OSEK** (ver nota siguiente). Éste procesado se hace con un comando llamado **generate**. Para no tener que hacer cada función por separado, lo que se hace es pasarle al MakeFile el comando 'clean_generate', que realiza ambas operaciones, en forma consecutiva y automática. Si se trabaja sin RTOS-OSEK sólo hace falta usar el comando 'clean'.

La primera vez que se compila el proyecto, es necesario hacer un *Clean Project*. Esto ejecutará el comando *Clean_generate* del make, creando todos los archivos necesarios para la compilación con el RTOS.

<u>NOTA</u>

OSEK(alemán: **O**ffene **S**ysteme deren und Schnittstellen für *die* **E***lektronik* in Kraftfahrzeugen) (inglés: Open Systems and their Interfaces for the Electronics in Motor *Vehicles*) es un gremio de estandarización que ha producido especificaciones para sistemas operativos embebidos, un stack de comunicación, y un protocolo de control de red para un sistema embebido automotriz, además de otras especificaciones relacionadas. OSEK ("Sistemas abiertos y sus interfaces para la electrónica en automóviles") se diseñó para proveer una arquitectura de software abierta estándar para varias unidades de control electrónicas (ECU=Electronic Control Unit) incorporadas en vehículos, para facilitar la integración y portabilidad de software de diferentes proveedores, ahorrando costos y tiempos de desarrollo. Por otra parte, es un sistema operativo muy seguro, dado que todas sus funciones son determinísticas: al momento de iniciar el SO, cada tarea ya tiene asignado su espacio de memoria, evitando la necesidad de contar con instrucciones tipo 'malloc' para la asignación dinámica de memoria. Así, ante una situación de riesgo (p.ej. al producirse un choque), no se producen retardos en la ejecución de las tareas.

Para realizar esta configuración, ubíquese en la ventana de propiedades del proyecto *Firmware*, y seleccione la rama C/C++ *Build*: se verá una pantalla similar a la que muestra la Figura 30.

| Properties for Firmware | |
|--|---|
| type filter text | C/C++ Build 🗢 🔹 🔿 👻 💌 |
| Resource Builders C/C++ Build C/C++ General Linux Tools Path | Configuration: Default [Active] |
| Project References Run/Debug Settings Task Repository | Builder Settings Behaviour Refresh Policy Builder |
| WikiText | Builder type: External builder |
| | Build gommand: make Variables |
| | Makefile generation |
| | Build location Build girectory: S(workspace_loc:/Firmware)/ |
| | Workspace] File system] Variables |
| | Restore Defaults Apply |
| ? | OK Cancel |

Figura 30: Configuración del Makefile

Dentro de la rama '*C*/*C*++ *Build*', configure la pestaña '*Behaviour*' como muestra la Figura 31. Las configuraciones importantes son las siguientes:

- tilde la opción 'Stop on first build error' y destilde 'Enable parallel build'
- destilde el casillero 'Build on resource save' y tilde 'Build (Incremental Build)' y 'Clean'. En el campo Clean, escriba: clean_generate
- borre el contenido del campo *Build* y déjelo en blanco

| Properties for blinking | | | |
|---|-------------------------------------|-------------------------|------------------|
| type filter text | C/C++ Build | | (-) ▼ −) ▼ ▼ |
| Resource | | | |
| Builders ▷ C/C++ Build ▷ C/C++ General | Configuration: Default [Active] | ▼ Manage | e Configurations |
| Git | | | |
| Linux Tools Path | 🗏 Builder Settings 🛞 Behaviour 🦑 | Refresh Policy | |
| Project References Run/Debug Settings | Build settings | | |
| ▶ Task Repository ₩ikiText | Stop on first build error | | |
| | Workbench Build Behavior | | |
| | Workbench build type: | Make build target: | |
| | Build on resource save (Auto build) | all | Variables |
| | Note: See Workbench automatic build | preference | |
| | 🕼 Build (Incremental build) | | Variables |
| | 🔽 Clean | clean_generate | Variables |
| | | Restore <u>D</u> efault | s <u>Apply</u> |
| ? | | ОК | Cancel |

Figura 31: Configuración de comportamiento del IDE con el MakeFile

Luego de hacer esto, damos presionamos en Ok, y luego vamos sobre el proyecto Firmware, hacemos clic derecho, y buscamos la opción *Clean Project*. Por último, damos clic en *Build Project*. Dichos opciones se muestran en la Figura 32, y si todo salió correctamente, en la consola del IDE vamos a ver una ventana como la que se muestra en la Figura 33. Particularmente, vamos a ver una línea que nos dice que se ha creado un archivo Blinking.axf



Figura 32: Menú de opciones del proyecto para hacer clean y build Project



Figura 33: Consola del software IDE luego de una compilación exitosa

6 Debug con Windows

Podríamos querer probar este ejemplo para ver de qué manera interactúa el software creado con el hardware de la EDU-CIAA, o verificar que no hay errores en nuestra lógica de programación. Para ello, debemos compilar el proyecto (Sección 5), y luego efectuar una depuración (*debug*). Aunque resulte trivial decirlo, si no contamos con el hardware, definitivamente no podremos hacer lo segundo. No obstante, para esos casos se ha desarrollado una pequeña aplicación para el ejemplo '*Blinking*' que hace funcionar la aplicación del Firmware en nuestra PC, generando las respuestas similares a las que haría la placa (si corriéramos el código en una EDU-CIAA, se produciría el encendido y apagado del LED). Aunque no sea tan lindo y emocionante como ver que las luces se prenden y se apagan, lo más cercano y simple que se puede hacer es que en cada conmutación del LED aparezca un mensaje en la consola del Software IDE que diga algo como '*Blinking*'. Esto es lo que haremos en la sección presente, y le daremos por nombre '*Win Debug*'. Dado que la aplicación compilada correrá en nuestra PC, el código está configurado para dar como resultado un archivo .exe.

Para que 'Win Debug' funcione debemos configurar un par de cosas. En principio debemos decirle al IDE dónde se encuentra el ejecutable que genera nuestro Makefile: haga click derecho sobre el proyecto, seleccione '*Debug As'* y a continuación '*Debug Configurations...*', como muestra la Figura 34.

| 🖨 C/ | C++ - E | | New | • | - |
|--------------|--------------|-----------|-----------------------------|---------------------|---------------------------------|
| <u>F</u> ile | <u>E</u> dit | | Go Into | | <u>H</u> elp |
| 1 | - 🛛 (| | Open in New Window | | • 🔾 • 🤮 • 隆 • 🖄 • |
| | 🔁 Pr | | Сору | Ctrl+C | |
| ۲ | ⊿ 🙋 | Ē | Paste | Ctrl+V | |
| | Þ | × | Delete | Delete | |
| 9 | | <u>®_</u> | Remove from Context | Ctrl+Alt+Shift+Down | |
| | | | Source | + | |
| | Þ | | Move | | |
| | Þ | | Rename | F2 | |
| | | r de la | Import | | |
| | | 2 | Export | | |
| | | | Build Project | | |
| | | | Clean Project | | |
| | | ন্ত্রী | Refresh | F5 | |
| | | | Close Project | | |
| | | | Close Unrelated Projects | | |
| | | | Build Configurations | • | |
| | | | Make Targets | + | |
| | | | Index | • | 1s 🧔 Tasks 📃 Console 🔀 🔲 Proper |
| | | | Show in Remote Systems view | | to display at this time. |
| | | | Profiling Tools | • | |
| | | | Convert To | | |
| | | | Profile As | ۱. | |
| | | | Debug As | ۱. | 1 Local C/C++ Application |
| | | | Run As | - - | Debug Configurations |
| | | | Compare With | - | |
| | | | Restore from Local History | | |
| | | * | Run C/C++ Code Analysis | | |
| 😂 ы | inking | | Team | • | |
| | | | Properties | Alt+Enter | |

Figura 34: Acceso a Debug Configurations

Una vez hecho esto, hacemos doble click en la opción 'C/C++ *Application*', lo cual nos creará un módulo donde podremos configurar un programa para que emule el comportamiento de la placa. El resultado se muestra en la Figura 35.

| 1 (2) 60 (100) -+L | / | | (| |
|---|---|--|----------------|--|
| | Mame: Firmware Default | | | |
| C C/C++ Application | Main ⋈= Arguments C/C++ Application: | Environment 🏇 Debugger 🦆 Source 🗖 <u>C</u> ommon | | |
| C/C++ Attach to Applic C/C++ Postmortem Del C/C++ Remote Applical GDB Hardware Debuggi | Project: | Variables Searc <u>h</u> Project | Browse | |
| GDB OpenOCD Debugg | Firmware | | <u>B</u> rowse | |
| Launch Group | Build (if required) before launching | | | |
| | Build configuration: | Use Active | * | |
| | | Select configuration using 'C/C++ Application' | | |
| | 🖱 Enable auto build | Disable auto build | | |
| | Use workspace settings | Configure Workspace Settings | | |
| ۰ | Using GDR (DSE) Create Process | Launcher Select other | Revert | |

Figura 35: Creación de una opción de depuración por medio de un emulador

El ejecutable que genera el Makefile se almacena en la carpeta <Carpeta de instalación del software-IDE>*Firmware\out\bin* bajo el nombre de *'blinking.exe'*. La configuración final se muestra en la Figura 36, y si se usan las rutas por defecto, los valores son los siguientes:

- C/C++ Application: C:\CIAA\Firmware\out\bin\blinking.exe
- Project: *blinking*
- Tildar la opción 'Use Workspace settings'
- Tildar la opción 'Select configuration using 'C/C++ Application'

Es muy posible que el IDE informe que no puede encontrar el Código Fuente cuando se intenta hacer Debug sobre este proyecto. Aparecerá una leyenda en el medio de la consola que se parece a la de la Figura 37. Para solucionarlo se debe crear un Path Mapping que convierta el estilo de ruta de CygWin al formato Windows. Abra la ventana de configuración del software IDE haciendo click en 'Windows \rightarrow Preferences'. Seleccione la rama 'C/C++', luego la rama 'Debug' y ubique la opción 'Source Lookup Path'. Finalmente, en el lado derecho haga click en el botón 'Add': aparecerá una ventana como la que se muestra en la Figura 38.

| Debug Configurations | | |
|--|---|---|
| Create, manage, and run conf | gurations | To a |
| Image: Second system Image: Second system Image: C/C++ Application Image: C/C++ Attach to Application Image: C/C++ Attach to Application Image: C/C++ Remote Application | Name: Firmware Default Main Image: Arguments Image: Environment C/C++ Application: C:\CIAA\Firmware\out\bin\blinking.exe Draject: Firmware Build (if required) before launching Build configuration: Default Image: Imag | |
| ۲ الله الله الله الله الله الله الله الل | Enable auto build Use workspace settings Using GDB (DSF) Create Process Launcher - Settings | © Disable auto build <u>Configure Workspace Settings</u> elect other Apply Revert |
| ? | | Debug Close |

Figura 36: Configuración del emulador para el proyecto Blinking

| ☆ Debug ⊠ | 🍇 # i⇒ | ∇ \Box |
|--|----------|-----------------|
| ✓ C Firmware Default [C/C++ Application] | | |
| <pre>{commated, exit value: 0>gdb</pre> | | |
| ▲ [™] blinking.exe [2224] | | |
| Thread [1] 0 (Suspended : Breakpoint) | | |
| main() at blinking.c:105 0x4011ae Thread [210 (Suspended : Container) | | |
| P Thread [2] 0 (Suspended : Container) P Thread [3] 0 (Suspended : Container) | | |
| ▷ monopological provided and provided an | | |
| 💽 main() at /cygdrive/c/CIAA/Firmware/examples/blinking/src/blinking.c:105 0x4011ae 🙁 | | |
| Can't find a source file at "/cygdrive/c/CIAA/Firmware/examples/blinking/src/blinking.c" Locate the file or edit the source lookup path to include its location. | | |
| View Disassembly | | |
| Locate File | | |
| Edit Source Lookup Path | | |
| | | |
| | | |
| | | |

Figura 37: Error de IDE. No se encuentra el código fuente

| e filter text | Source Lookup Path | |
|--------------------------|------------------------------|------------------------|
| General ^ | Default Source Lookup Path: | |
| Appearance | 🕆 Absolute File Path | Add |
| Autotools | 📴 Program Relative File Path | 4 |
| ▷ Build | Project | Edit |
| Code Analysis | | Remove |
| A Debug 7 | | Un |
| Breakpoint Actions | | Up |
| Debugger Types | | Down |
| Disassembly | | |
| Floating Point Memory F | | |
| GDB MI | | |
| Source Lookup Path | | |
| 5 Tracepoint Actions | | |
| Traditional Memory Ren | | |
| Editor | | |
| File Types | | |
| Language Mappings | | |
| New C/C++ Project Wizard | | |
| Profiling | | |
| Property Pages Settings | | |
| Task Tags | | |
| Changel og | | |
| Help | | |
| Install/Update | | |
| Library Hover | | |
| Mylyn | | |
| Remote Systems | | <u> </u> |
| D | | Restore Defaults Apply |
| Run/Debug | | The store Delaging |

Figura 38: Configuración de mapeo de rutas para el entorno Windows

Cuando queramos agregar una fuente (*Source*), nos aparecerá el recuadro de la Figura 39, donde debemos elegir el tipo '*Path Mapping*'.

| Select the type of source to add to the source l | | |
|--|------------|--|
| A path mapping. | ookup path | |
| 🕏 Absolute File Path | | |
| 😂 Compilation Directory | | |
| Bile System Directory | | |
| 🖉 Path Mapping | | |
| IIII Program Relative File Path | | |
| 🔁 Project | | |
| 🕞 Project - Path Relative to Source Folders | | |
| 🗁 Workspace | | |
| 😂 Workspace Folder | | |

Figura 39: Agregando un Path Mapping

Nos aparecerá un nuevo recuadro, como el de la Figura 40, donde debemos elegir la opción '*Add*', y rellenamos con los siguientes datos:

- Name: *Firmware*
 - Compilation Path:
 - \cygdrive\c\CIAA\Firmware
- Local file system path: C:\CIAA\Firmware (si usamos los directorios por defecto)

| 🖉 Path Mappings | | |
|---------------------------|-------------------------|-----------|
| Specify the mapping paths | | |
| Name: Firmware | | |
| Compilation path: | Local file system path: | Add |
| Cygdrive\c\CIAA\Firmware | C:\CIAA\Firmware | Remove |
| | | Up |
| | | Down |
| | | |
| | | |
| | | |
| • | | 4 |
| ? | | OK Cancel |

Figura 40: Configuración para crear un Path Mapping en Windows

Con todas estas configuraciones ya podremos depurar el ejemplo Blinking, emulando las respuestas a través de la consola.

<u>Importante</u>

Si bien la creación de un *Path Mapping* nos ayuda a emular el comportamiento del titileo de LEDs, nos genera conflictos cuando hacemos Debug con la propia placa, pues bloquea el uso de *breakpoints*. Si contamos con la placa y deseamos hacer Debug con la misma, debemos borrar el Path Mapping creado.

7 Debug en placa EDU-CIAA y Entorno IDE

7.1 Configuración del entorno CIAA-IDE

Si se cuenta con una placa EDU-CIAA y se quiere depurar sobre el hardware, lo primero que hay que hacer es una limpieza mediante '*Clean Project*'. Esto es necesario porque en todos los *builds* que se hicieron para el Win Debug se crearon archivos que pueden entorpecer la compilación del Debug sobre la placa, pues los mismos estaban pensados para el CPU de la PC, y no para el μC que forma parte de la EDU-CIAA.

A continuación necesitamos que el compilador reconozca que se quiere compilar sobre la placa EDU-CIAA. Ello implica que, en función de la placa disponible, se incluya el código acorde al μC utilizado. Dado que el código cuenta con sentencias *if* de pre-procesado, que dependen del hardware disponible, es necesario indicarle con qué versión de la plataforma se cuenta. Para ello, vamos a modificar manualmente el archivo MakeFile.mine, que es una rama del MakeFile, más reducido, que tiene los datos mínimos y necesarios para compilar. Para hacerlo, desde el software IDE, al mismo nivel que el proyecto Firmware (Figura 41), vamos a encontrar el MakeFile.mine que debemos modificar.



Figura 41: Project Explorer - MakeFile.mine

Le hacemos doble clic y nos va a aparecer a nuestra derecha, un archivo de texto. Si bajamos un poco, vamos a encontrar el código que se muestra en la Figura 42.

| 📄 *Ma | kefile.mine 🛛 | | | | | - | E |
|--|--|---|-----------------------------------|-----------------|--|-----------------|---|
| 52 # 53 ### 54 # / | ARCH, CPUTYPE a | and CPU following | are supported | | | | * |
| 56# | ARCH | CPUTYPE | CPU | COMPILER | BOARD | T | |
| 57 # • 58 # 59 # | x86 | ia32 ia64 | | gcc | ciaa_sim_ia32 ciaa_sim_ia64 | * | |
| 61 # 62 # 63 # | cortexM4 | lpc43xx k60_120 | lpc4337 Â mk60fx512vlq15 | gcc gcc | edu_ciaa_nxp ciaa_nxp ciaa_fsl | * | |
| 65 # | mips | pic32 | pic32mz | gcc | ciaa_pic | T | |
| 67 # 68 # : 69 #AF 70 #CF 71 #CF 72 #CC | if you define t RCH = PUTYPE = PU = = MPILER = = | the BOARD the oth = x86 = ia32 = none = gcc | ers parameters are | e optional. | | - | |
| 73 BO/ 74 ### | ARD ?= | • ciaa_sim_ia32 | ****** | | ***** | | - |
| <pre>75 # rtostests options 76 # 77 # RTOSTESTS_DEBUG_CTESTS ?= 1 , set debug flag in ctest.pl 78 # RTOSTESTS_LEAN_GENERATE ?= 1 , skips make clean and generate (use it only if you are running a single sub test case, but after gen 79 # RTOSTESTS_CTESTS_?= 'ctest_tm_01:Test Sequence 1' , test case name, based on ctestcases.cfg (use it for single ctest) 80 # RTOSTESTS_SUBTEST ?= 'full-preemptive' , sub test case name, based on ctestcases.cfg or empty to start running from this test to th 81 82 # Projects 83 #</pre> | | | | | | | |

Figura 42: Contenido del archivo MakeFile.mine

En él, podemos ver un cuadro donde, dependiendo de la plataforma en la que corramos el Firmware, debemos cambiar el valor de una variable que está debajo de él (la llamada BOARD. las otras no tiene efecto pues aparecen en forma de comentario, debido al símbolo # colocado al principio de cada línea). Como nosotros contamos con una EDU-CIAA, con micro NXP, entonces corresponde el siguiente valor de la variable:

BOARD ?= edu_ciaa_nxp

Y el archivo modificado nos quedará como se muestra en la Figura 43. Después de ello, tenemos que guardar los cambios.

| | ₫ • @ • ☆ • | 0 • 9 • 9 • 6 | 🖨 🖋 🕶 🔳 🕴 | • 🖗 • 🌾 | ← ▼ ⇒ ▼ | Quick Access | 5 |
|--|---|--|--|---|--|--|---|
| 🎦 Project Explorer 😒 🛛 📄 🤹 🗢 🗖 | Makefile.mine | 3 | | | | • | c |
| ▲ Simware > Simulates > Accepted as a second se | 52 # 53 ############ 54 # ARCH, CPU | YPE and CPU followi | ng are supported | | | | |
| examples externals | 56 # AR | TH CPUTYPE | CPU | COMPILER | BOARD | | |
| > modules > > out | 58 # x86 59 # | ia32 ia64 | | gcc | ciaa_sim_ia32 ciaa_sim_ia64 | | |
| be projects iii gdb.exe.stackdump Makefile Makefile | 60 # + | lpc43xx k60_120 | lpc4337 Â mk60fx512vlq15 | gcc gcc | edu_ciaa_nxp ciaa_nxp ciaa_fsl | I | |
| Makefile.config Makefile.mine README | 64 # + + + + + + + + + + + + + + + + + + | <pre>pic32 fine the BOARD the o</pre> | pic32mz thers parameters are set debug flag in , skips make clean 01:Test Sequence 1' eemptive', sub test | gcc e optional. :test.pl and generate ; test case n t case name, | ciaa_pic | u are running a single sub test case, but after ger tcases.cfg (use it for single ctest) .cfg or empty to start running from this test to th | - |

Figura 43: MakeFile.mine luego de la configuración

7.2 Compilación del proyecto

Una vez configurado el MakeFile, se debe limpiar (*Clean Project*) y luego compilar el proyecto (*Build Project*). Como se puede ver en la Figura 44, la extensión del archivo generado ya no es .exe sino .axf. Esta extensión es propia de la arquitectura ARM. Si esto no ocurre, vuelva al paso anterior, y verifique que no existen errores de tipeo en el nombre o en el valor de las variables que se modificaron en los MakeFile.

Para saber si se produjo una correcta compilación, vea la consola del IDE, ubicada en la parte inferior de la pantalla. En caso de que se hayan seguido todos los pasos y no se pueda compilar, hacer un *Clean* primero y luego un *Build*.

| 🎦 Project Explorer 🐹 📄 🖻 🖘 🖓 🗖 | |
|--------------------------------|--|
| ⊿ 😴 Firmware | |
| Includes | |
| b 🗁 doc | |
| Examples | |
| Externals | |
| > indules | |
| | |
| gdb.exe.stackdump | |
| Makefile | |
| Makefile.config | |
| Makefile.mine | |
| README | |
| | |
| | |
| | |
| | |
| | 📳 Problems 🧔 Tasks 📮 Console 🕱 🔲 Properties |
| | CDT Build Console (Firmware) |
| | |
| | Linking file: ./out/bin/blinking.axf |
| | arm-none-eabi-gcc ./out/obj/blinking.o ./out/obj/crp.o -Xlinkerstart-group ./out/lib/p |
| | |
| | Post Ruilding blinking |
| | · · · · · · · · · · · · · · · · · · · |
| | arm-none-eabi-objcopy -v -O binary ./out/bin/blinking.axf ./out/bin/blinking.bin |
| | copy from ./out/bin/blinking.axf [elf32-littlearm] to ./out/bin/blinking.bin'[binary] |
| | 22:03:21 Build Finished (took 1m:50s.216ms) |
| | |

Figura 44: Entorno del Software-IDE luego de la primera compilación

7.3 Depuración sobre la placa: configuración de OpenOCD para Debug

Durante la instalación del CIAA-IDE, además de los drivers para la conexión de la placa se instalan las herramientas para Debug. Se utilizará OpenOCD (una herramienta OpenSource creada para estos propósitos) para hacer el nexo JTAG-GDB mediante la conexión USB. Para configurar esta herramienta, desde el menú '*Run→Debug Configurations…*' se debe crear un módulo nuevo de '*Debug configuration*' del tipo '*GDB OpenOCD Debugging*'. A continuación,

coloque los valores que se muestran en las Figura 45 y Figura 46. Las mismas se listan a continuación:

- Pestaña Main:
 - 1. Name: *Blinking OpenOCD*
 - 2. Project: *blinking*
 - 3. C/C++ Application: C:\CIAA\Firmware\out\bin\blinking.axf
 - 4. Tildar *Disable auto build*
 - 5. Build configuration: *Default*
- Pestaña Debugger:
 - 1. OpenOCD Setup: Destildar la opción Start OpenOCD locally
 - 2. GDB Client Setup
 - a) Executable: C:\CIAA\cygwin\usr\arm-none-eabi\bin\arm-none-eabigdb.exe
 - b) Other options y Commands: dejarlo como está
 - c) Destildar 'Force thread list update on suspend'

Le damos clic a *Apply* para guardar los cambios, y no cerramos esta ventana.

| Create, manage, and run configu | rations | |
|--|--|--------------|
| Image: Image | Name: Firmware OpenOCD Main Debugger Startu C/C++ Application: C\CLAA\Firmware\out\bin\blinking Project: Firmware Build (if required) before launching Build configuration: Enable auto build Use workspace settings | p) ⅓ Source |
| Filter matched 9 of 9 items | | Apply Revert |
| ? | | Debug |

Figura 45: Configuración para poder hacer Debug sobre la placa: pestaña Main

| Debug Configurations | | 8 |
|-----------------------------------|---|----|
| Create, manage, and run configur | ons | ñ. |
| | 24 | |
| | ame: Firmware OpenOCD | |
| type filter text | 🛾 Main 🕸 Debugger 🗼 Startup 🦆 Source 🔲 Common | |
| C/C++ Application | OpenOCD Setup Start OpenOCD locally | |
| C C/C++ Postmortem Debugger | Executable: \${openocd_path}/\${openocd_executable} Browse Variables | |
| C/C++ Remote Application | GDB port: 3333 | |
| GDB OpenOCD Debugging | Telnet port: 4444 | |
| Firmware OpenOCD Launch Group | Config options: | E |
| | Allocate console for OpenOCD | |
| | GDB Client Setup | |
| | Executable: C:\CIAA\cygwin\usr\arm-none-eabi\bin\arm-none-eabi-gdb.exe Browse Variables | |
| | Other options: | |
| | Commands: set mem inaccessible-by-default off | |
| | - | - |
| Filter matched 9 of 9 items | Apply Revert | |
| ? | Debug | : |

Figura 46: Configuración para poder hacer Debug sobre la placa: pestaña Debugger

Para hacer Debug, es necesario que se abra un puerto a través del OpenOCD. Hay muchas formas de hacerlo. En este tutorial, lo haremos usando la consola CygWin, que viene incluida con el instalador del CIAA-IDE.

Abrimos la consola. Nos aparecerá una ventana como la que se muestra en la Figura 47.



Figura 47: Consola CygWin

Entonces, escribimos los siguientes comandos:

| Comando | Descripción |
|---------------------|---|
| cd C:/CIAA/Firmware | Nos posiciona en la carpeta Firmware (si usamos |
| | los Path por defecto) |
| make openocd | Comienza a correr el servicio del OpenOCD |

Si se cuenta con una Placa EDU-CIAA "virgen", cuya flash nunca ha sido programada, o si por algún motivo se ha borrado completamente la flash del microcontrolador, es posible que al intentar iniciar una sesión de Debug, el CIAA-IDE muestre un error. Para poder iniciar la sesión de debug, recomendamos seguir las instrucciones de la sección *Primeros pasos con el Hardware de la CIAA*, accesible en la página oficial del proyecto.

Luego de iniciado el servicio del OpenOCD, dejamos la ventana abierta, volvemos al entorno IDE, y damos clic al botón *Debug* de la ventana que dejamos abierta previamente.

7.4 Posible problema: "No reconocimiento"

Si se alcanzó este punto, si todo fue exitoso, al intentar hacer debug sobre la placa no debería haber problemas. No obstante, si por casualidad se desconectó la placa de la PC y se la volvió a conectar en un puerto USB distinto al anterior, Windows detectará el nuevo Hardware encontrado, instalará los drivers y la placa se encenderá; pero al intentar hacer Debug, en la consola CygWin es posible que emerja un mensaje de error como la de la Figura 48.



Figura 48: Ventana de error del Software IDE

Si se hiciera un par de pruebas, este error también corresponde al que se daría si la placa no estuviera conectada. La causa es que el driver que se acaba de instalar no es el corregido por el programa Zadig, sino es el instalado por el instalador del Software-IDE. Para solucionar este inconveniente simplemente hay que repetir los pasos que se destacan en la *Sección 3.5*, y con ello, la placa vuelve a funcionar correctamente.