

CIAA Firmware Coding Guidelines

INFO: Estamos trabajando en mejorar y extender estas Guidelines.

General

En esta sección se define un estilo de codificación para lenguaje de programación C a través de un conjunto de normas y recomendaciones que deben ser utilizadas en toda la documentación del proyecto. El uso de una guía común asegura la correcta comprensión de todo el código, además de facilitar el mantenimiento.

Lo expuesto en esta sección, trata exclusivamente del estilo de programación en lenguaje C en cuanto a codificación, es decir, no aborda cuestiones de diseño, organización funcional, etc.

Para mayor información sobre guías de estilo de codificación pueden consultarse:

- [C STYLE GUIDE-NASA Software Engineering Laboratory Series SEL-94-003](#)
- [MISRA C](#)

Indentation: 3 espacios y no tabs

Las normas de indentation (sangría, mal traducido como indentación) indican la posición en la que se deben colocar los diferentes elementos que se incluyen en el código fuente. Este corrimiento hacia la derecha, un cierto espacio, de un bloque de texto, permite visualizarlo mejor y más rápidamente del texto contiguo.

El uso de caracteres espacio (3) y no el caracter TAB (tabulador) asegura que los distintos programadores hagan uniformemente el espaciado y no sea dependiente del tamaño (cantidad de espacios) que tienen definido en el TAB.

No trailing spaces

Existe una razón práctica de evitar espacios en blanco innecesarios al final de una línea, y es que si los programadores no tienen cuidado de evitar este uso, cualquier necesidad de hacer una comparación de cambios o buscar diferencias en archivos, surgen diferencias que interfieren en líneas aparentemente sin cambios, sólo porque alguien ha retirado o agregado espacios.

Comentarios

Los comentarios del código deberán estar escritos en inglés.

- Los comentarios deberán tener el mismo nivel de sangrado que el código que comentan.

- Los comentarios no deberán repetir y/o reformular el código, ya que su propósito es explicar la intención del código.
- No se recomiendan los comentarios de fin de línea (doble barra inclinada / /), puesto que su utilización declina en el uso de pocas palabras o abreviaturas que dificultan su comprensión.

¡Comentar no es pérdida de tiempo!

La métrica CP (Comment Percentage) se define como el número de líneas de comentarios dividido por el número de líneas (no en blanco) del código. Un valor aceptable para este parámetro es de alrededor del 20%

Doxygen

Los comentarios de funciones, macros, types, etc. deberán estar hechos siguiendo la sintaxis utilizada por el generador automático de documentación [doxygen](#)

Plantillas (templates)

La estructura de cada nuevo archivo debe estar basada en los templates definidos para tal fin. Lea atentamente el contenido de las distintas secciones y complete de acuerdo con lo solicitado en ellas.

Los modelos de plantilla se encuentra en:

<https://github.com/ciaa/Firmware/tree/master/modules/template>

Nombres de macros

- POSIX or standard macros use the name as defined in the standard with the `ciaaPOSIX_` prefix
 - `#define ciaaPOSIX_stdio_MAXFILDES 20`
- Non standard interface use the following schema `ModuleName_MACRO_NAME`
 - `#define ciaaDioDevices_MAXDEVICES 20`

Nombre de funciones

- POSIX or standard interfaces use the name as defined in the standard with the `ciaaPOSIX_` prefix
 - `int32_t ciaaPOSIX_open(char const * path, uint8_t oflag)`
- Non standard interfaces use following schema `ModuleName_functionName`
 - `void ciaaDioDevices_init(void)`

Commits

Cuando sea necesario hacer un commit en los repositorios, se deberá incluir una breve descripción de

los cambios realizados. Para esto se deberá utilizar el comando:

```
git commit -m "<mensaje>"
```

Donde "mensaje" se usa para asociar al commit con la descripción mencionada.

Para las actualizaciones relacionadas con el firmware, se deberá vincular el commit con los tickets (issue) implicados en los cambios realizados. Esto quiere decir, hacer:

```
git commit -m "#123 Descripción del commit vinculado al ticket 123"
```

```
git commit -m "#123 Descripción del commit vinculado al ticket 123 y también al #246 por tal y cual cosa!"
```

Luego, cuando se hace el push queda vinculado el commit con ese ticket y es mucho más fácil el seguimiento.

Hacer un ticket es muy sencillo, se puede crear desde la web github (botón "issues").

From:
<http://www.proyecto-ciaa.com.ar/devwiki/> -

Permanent link:
http://www.proyecto-ciaa.com.ar/devwiki/doku.php?id=desarrollo:firmware:fw_coding_guidelines

Last update: **2014/11/27 20:28**

