**Tutorial de configuración de uC-OSII para el ATMEGA1284P en AVRStudio5**

Autor: Francisco J. Rojas Correa 05 ABR 2011

Basado en el trabajo de las siguientes personas:

* Jean Labrosse – Creador de uC-OSII
* Ole Saether, Jesper Hansen, Julius Luukko – Desarrolladores del port para la plataforma AVR con el compilador AVR-GCC
* Diego Bouvier – Desarrollador del ejemplo que sirvió de base para este tutorial
* Cuauhtémoc Sergio Carbajal Fernández – Facilitador del material didáctico para la creación de este tutorial
* Moderadores del foro AVRFreaks

Requerimientos:

* AVR Studio 5.0 Beta o posterior

<http://www.atmel.com/dyn/resources/prod_documents/avrstudio5.0.beta.exe>

* Source de uc-OSII 2.86 y port para el compilador GCC de AVR (incluido en el archivo anexo)
* Microcontrolador ATMEGA1284P o librería en programa de simulación
* Programador AVRISPmkII o equivalente, o en su defecto el programa Proteus ISIS (de preferencia la versión más reciente) o equivalente.

Desarrollo:

1. Instalar el AVR Studio 5.0, de preferencia usar las opciones predeterminadas.
2. Descomprimir del archivo anexo las carpetas de port y source en **F:\DIR** (F:\DIR es el directorio de trabajo donde se desarrollará el proyecto).
3. Abrir la aplicación de AVR Studio 5, esta se muestra en la figura 1.

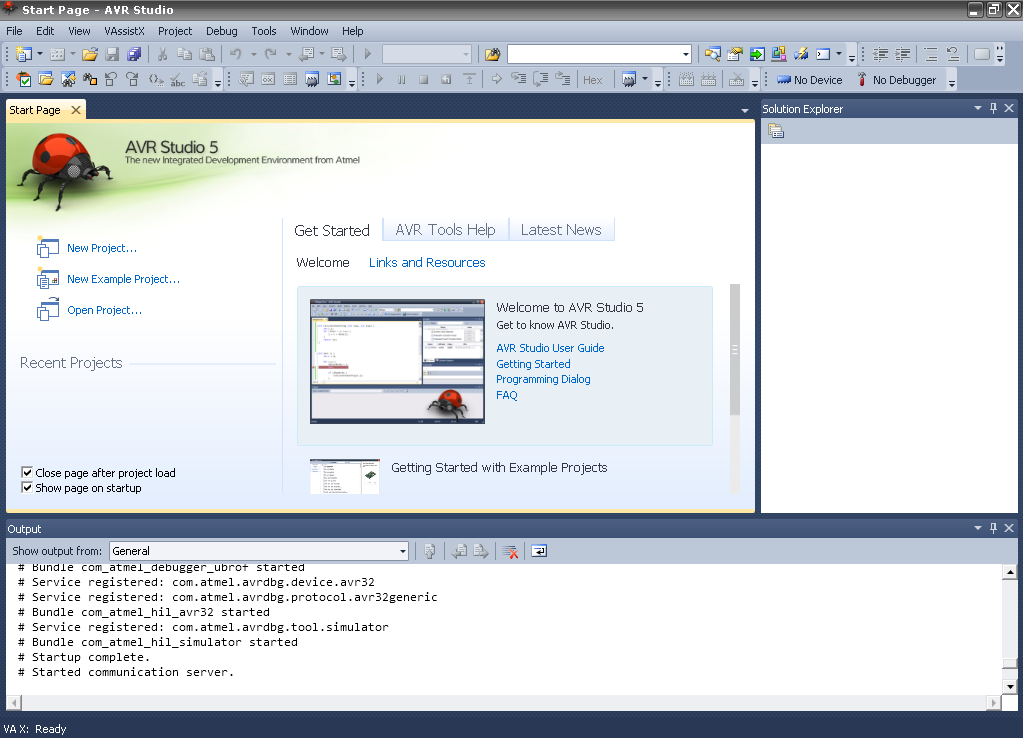


Figura 1. Pantalla de inicio de la aplicación AVR Studio

1. Seleccionar la ruta **File>New>Project**, lo que abre la ventana de la figura 2. Aquí seleccionar la opción **Empty AVR GCC Project** y dar un nombre y locación de la carpeta de trabajo. En este caso, el proyecto se llamará **OS\_TEST\_T** y la carpeta será en **F:\DIR\** y se activará la casilla **Create Directory for Solution.** Una vez realizado lo anterior, dar clic en el botón de OK.

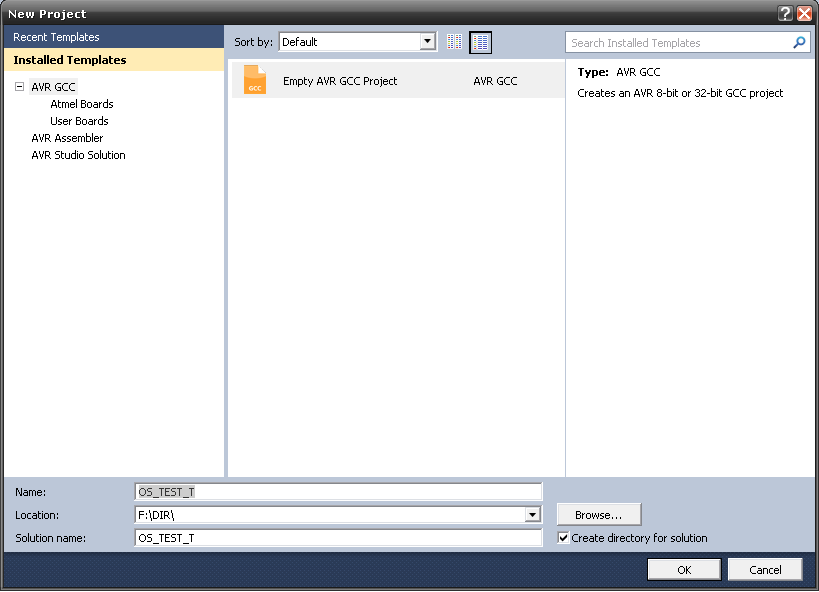


Figura 2. Pantalla 1 de configuración del proyecto

1. A continuación se abre la pantalla de la figura 3. En esta sección se solicita seleccionar el dispositivo a programar. Para efectos de este proyecto, el dispositivo es el ATMEGA1284P, por lo tanto se selecciona este y se da clic en el botón de OK.

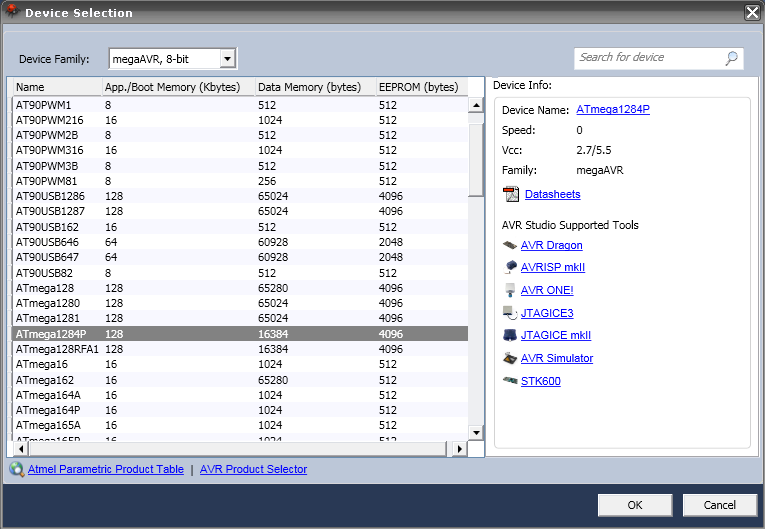


Figura 3. Pantalla de selección de dispositivo del proyecto

1. Si todo hasta el momento ha salido bien, se tiene que visualizar una pantalla como la de la figura 4.

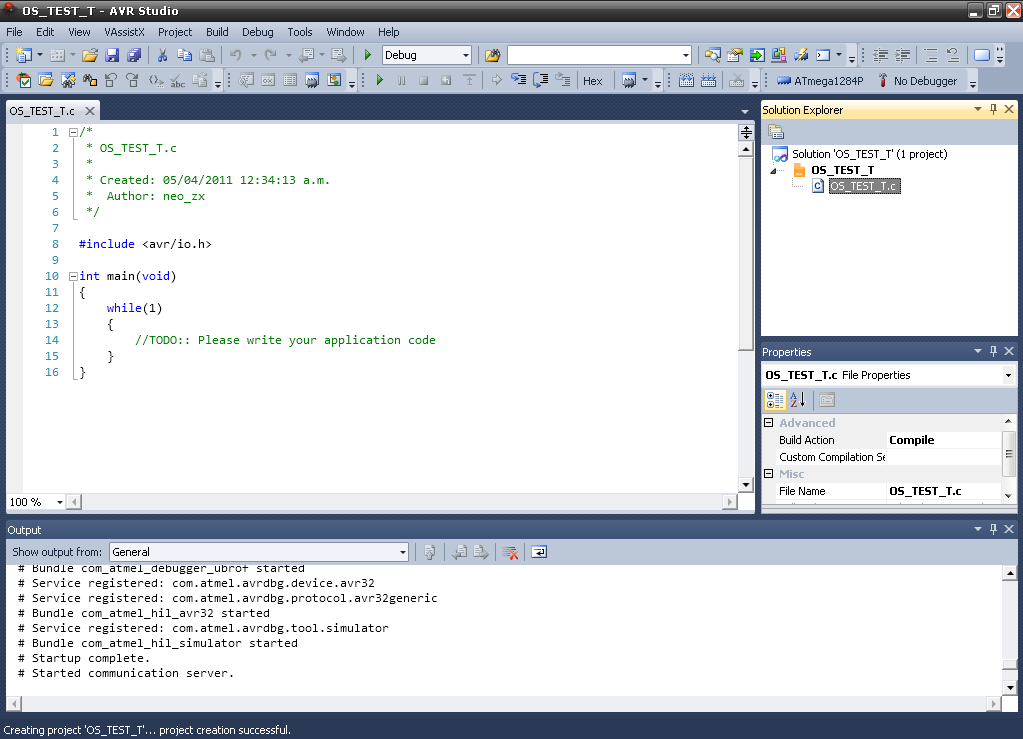


Figura 4. Resultado posterior a la configuración inicial

1. Ir a la carpeta **F:\DIR\port** y copiar los archivos contenidos ahí a la carpeta **F:\DIR\OS\_TEST\_T\OS\_TEST\_T**.
2. Ir a la carpeta **F:\DIR\source** y copiar los archivos contenidos ahí a la carpeta **F:\DIR\OS\_TEST\_T\OS\_TEST\_T.**
3. Verificar que se tengan los archivos ilustrados en la figura 5 en la carpeta **F:\DIR\OS\_TEST\_T\OS\_TEST\_T.**

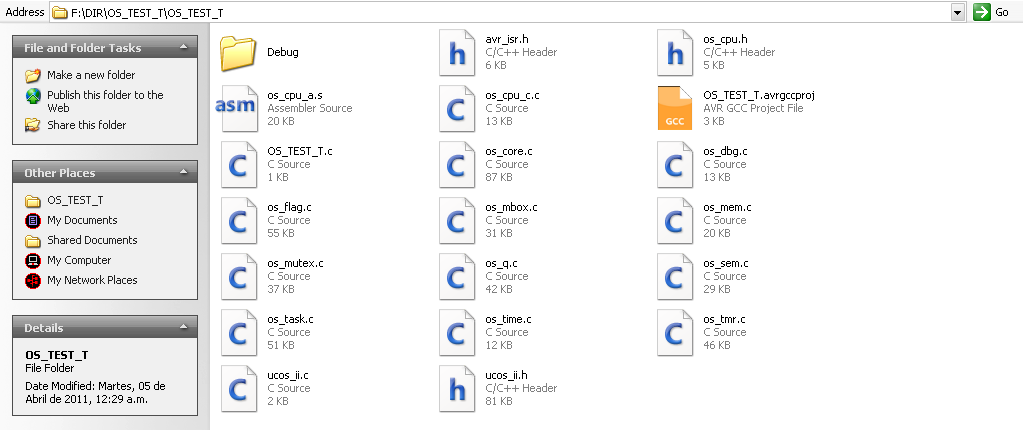


Figura 5. Contenido de la carpeta F:\DIR\OS\_TEST\_T\OS\_TEST\_T.

1. Regresar a la aplicación AVR Studio 5.0, y en la ventana de **Solution Explorer** dar clic derecho en el ícono de color naranja **OS\_TEST\_T** y seguir la ruta **Add>New Item**. Lo anterior resulta en una ventana como la ilustrada en la figura 6. Seleccionar **Include File** y darle el nombre de **app\_cfg.h**. Después dar clic en el botón Add.

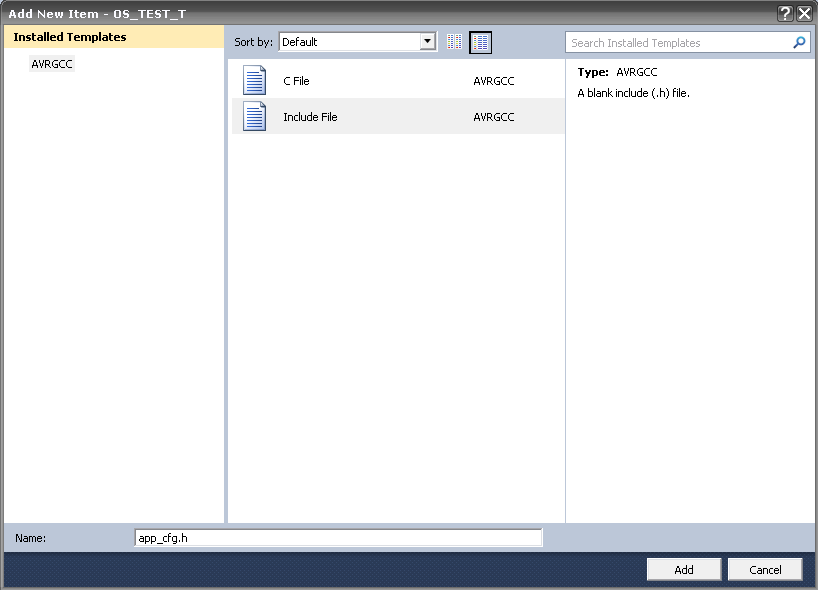


Figura 6. Adición del archivo app\_cfg.h al proyecto

1. En el archivo recién creado, definir todas las constantes que utilizara el archivo principal, que en este caso será el archivo **OS\_TEST\_T.c.** Una vez realizado lo anterior, proceder a guardar el archivo. En el caso de este proyecto, se escribirá el siguiente código el archivo **app\_cfg.h**



1. Repetir el paso 10, pero en esta ocasión crear un archivo de cabecera con el nombre **includes.h**.
2. En el archivo recién creado, definir todas las constantes que utilizara el archivo principal, que en este caso será el archivo **OS\_TEST\_T.c.** Una vez realizado lo anterior, proceder a guardar el archivo. En el caso de este proyecto, se escribirá el siguiente código el archivo **includes.h**



1. En el archivo **OS\_TEST\_T.c**, escribir el siguiente código:



1. Si hasta el momento todo ha salido bien, debemos poder ver una pantalla como la ilustrada en la figura 7.

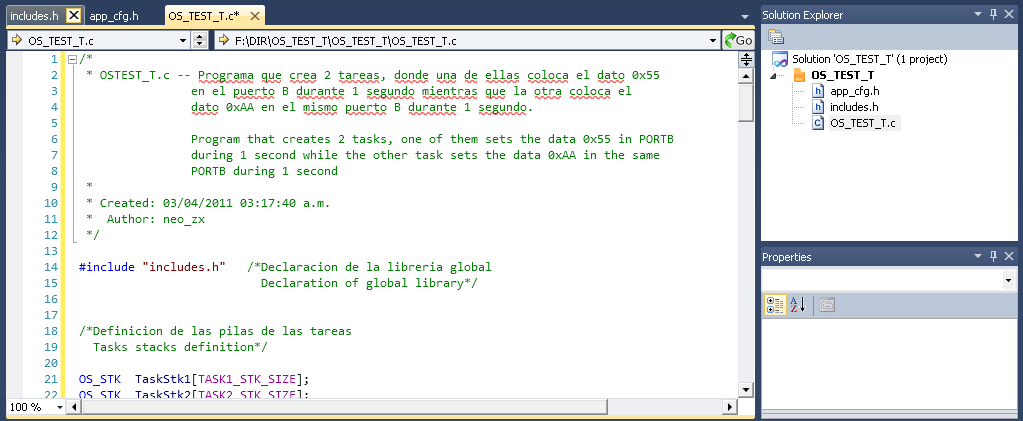


Figura 7. Adición de los archivos includes.h y app\_cfg.h

1. El siguiente paso será configurar cuestiones propias del entorno. Seguir la ruta **Project>OS\_TEST\_T Properties**. Lo anterior resulta en la apertura de la figura 8. En la sección **Build**, seleccionar **Release** en el campo de **Configuration.** Posteriormenteseleccionar la casilla **.hex**.

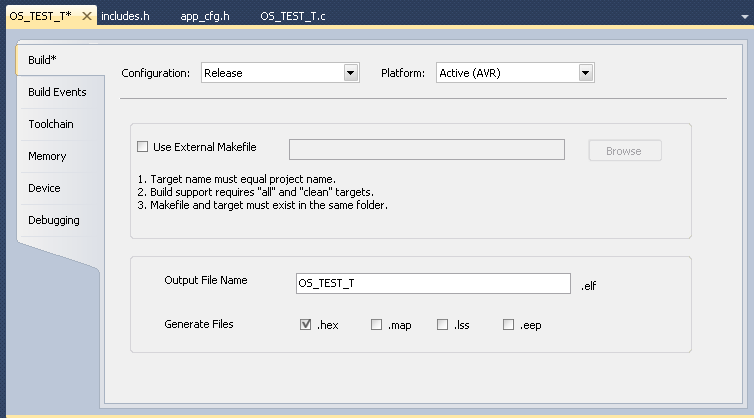


Figura 8. Opción Build de las propiedades de los objetos

1. En la opción **Toolchain**, en la sección **AVR/GNU C Compiler>Directories** hacer clic en el símbolo  y seleccionar la casilla **Relative Path**, posteriormente seleccionar la carpeta donde está el archivo principal (en este caso **OS\_TEST\_T.c**). Si la configuración fue correcta, debe de visualizarse una pantalla similar a la que aparece en la figura 9.

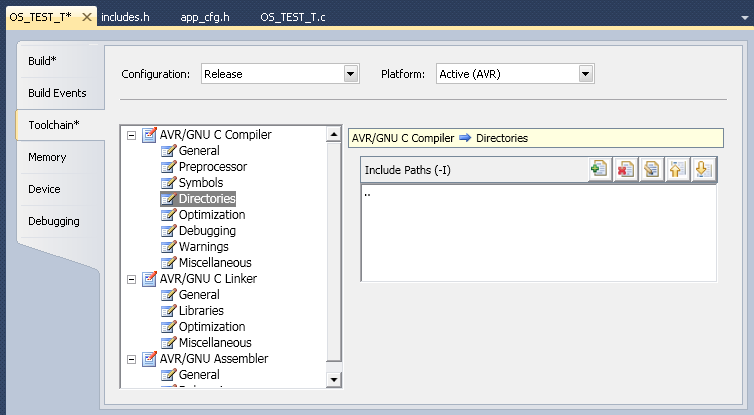


Figura 9. Búsqueda de los directorios usados por el archivo principal

1. En la sección **Directories** hacer clic en el símbolo  y escribir **F\_CPU=20000000**. Este último valor es la frecuencia de operación del CPU. El resultado debe verse en una pantalla similar a la que aparece en la figura 10.

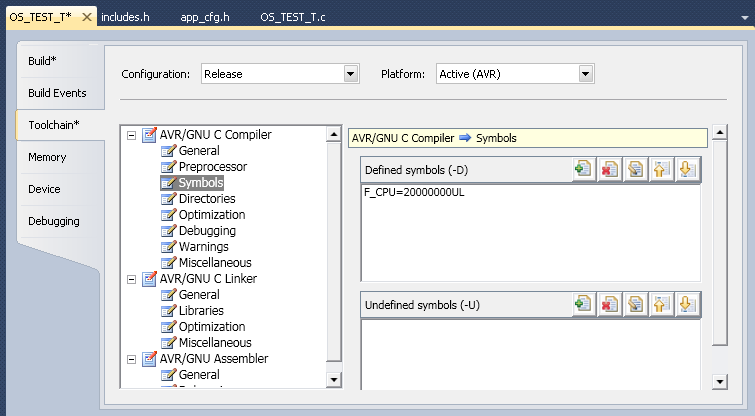


Figura 10. Información de la frecuencia de operación del dispositivo

1. En la ventana de **Solution Explorer** dar clic derecho en el ícono de color naranja **OS\_TEST\_T** y seguir la ruta **Add>Existing Item**. Seleccionar los archivos que se señalan en la figura 11:

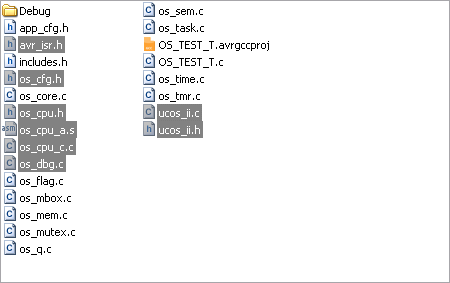


Figura 11. Archivos a incluir en el proyecto

Si todo salió correctamente, en la ventana de **Solution Explorer** debemos poder ver algo similar a lo que aparece en la figura 12.



Figura 12. Lista de archivos en el proyecto

1. Abrir el archivo **os\_cfg.h** y en la línea donde viene el comando #define CPU\_CLOCK\_HZ asignar el valor de 20000000, que de nuevo viene a ser el valor de frecuencia del CPU del dispositivo. El cambio se puede apreciar en la figura 13.

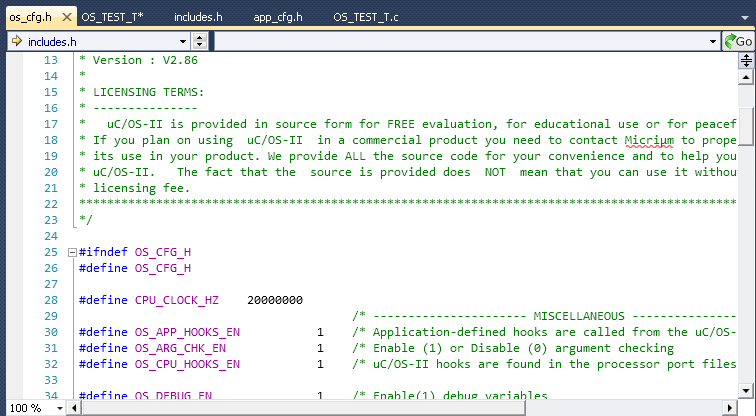


Figura 13. Archivo os\_cfg.h

1. Abrimos la carpeta \Atmel\AVR Studio 5.0\extensions\Application\AVR Toolchain\avr\include\avr del programa AVR Studio 5. Una vez ahí, buscamos el archivo correspondiente al archivo de cabecera del microcontrolador ATMEGA1284P, que en este caso es el **iom1284p.h**. Esto último se puede apreciar en la figura 14.

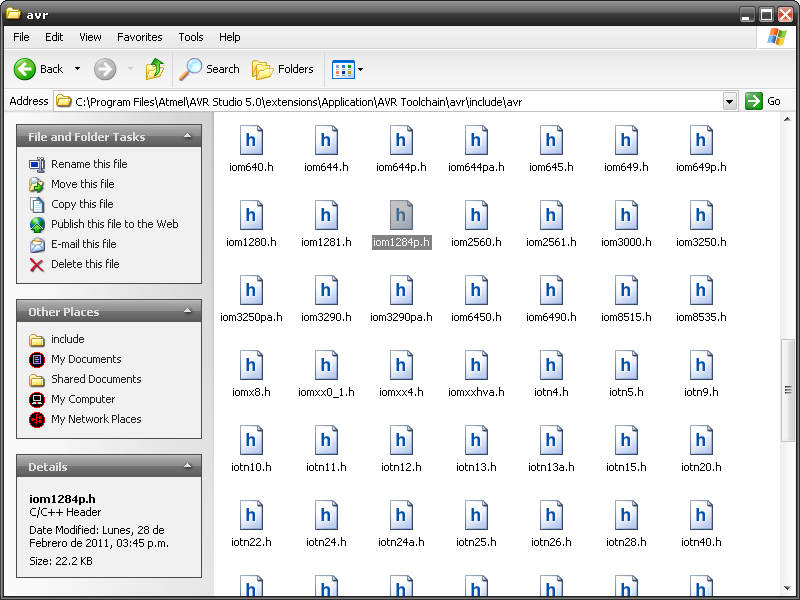


Figura 14. Carpeta de archivos de cabecera de dispositivos AVR

1. Abrir el archivo del paso anterior, y buscar la sección de **/\* Interrupt Vectors \*/.** Una vez ahí, localizar la definición del overflow del timer a utilizar. En este caso se utilizará el timer 0, por lo tanto la línea correspondiente es aquella que tiene **#define TIMER0\_OVF\_vect \_VECTOR(18) /\* Timer/Counter0 Overflow \*/** . NO MODIFICAR ESTE ARCHIVO ni cerrarlo aún.

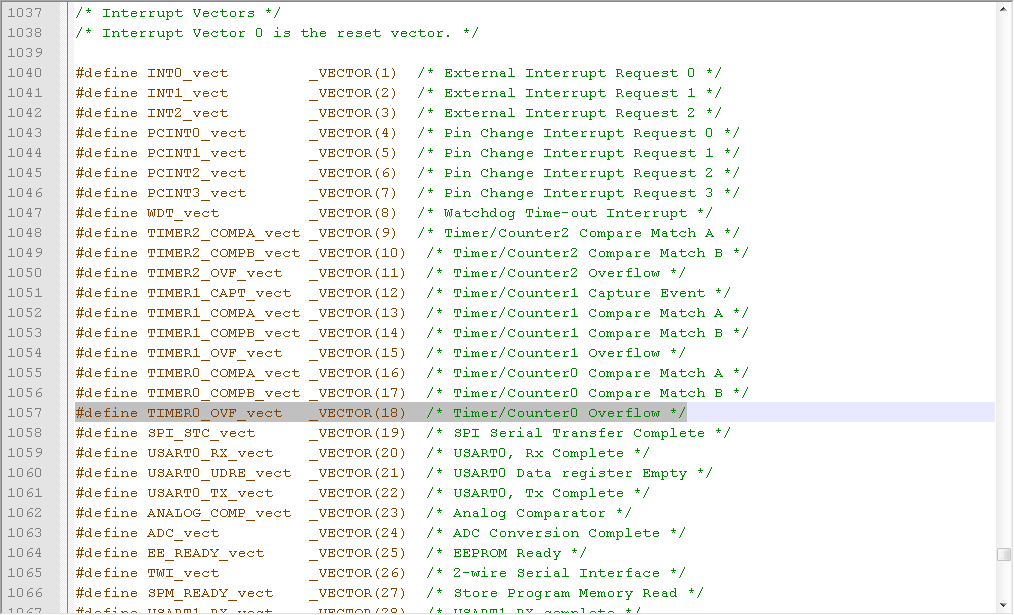


Figura 15. Contenido del archivo iom1284p.h

1. Regresar al editor AVR Studio 5 y abrir el archivo **os\_cpu\_a.s**. Reemplazar el texto **SIG\_OVERFLOW0** con **TIMER0\_OVF\_vect** en todo el archivo. SIG\_OVERFLOW0 corresponde al nombre de la interrupción de desbordamiento del ATMEGA1281, que es el dispositivo para el que originalmente este port fue hecho. Sin embargo, el nombre de esa interrupción en el ATMEGA1284P tiene un nombre distinto, en este caso, para el timer0 se llama TIMER0\_OVF\_vect. Si se quiere usar otro timer, se puede utilizar el nombre la interrupción de cualquier otro timer incluido en el archivo iom1284p.h

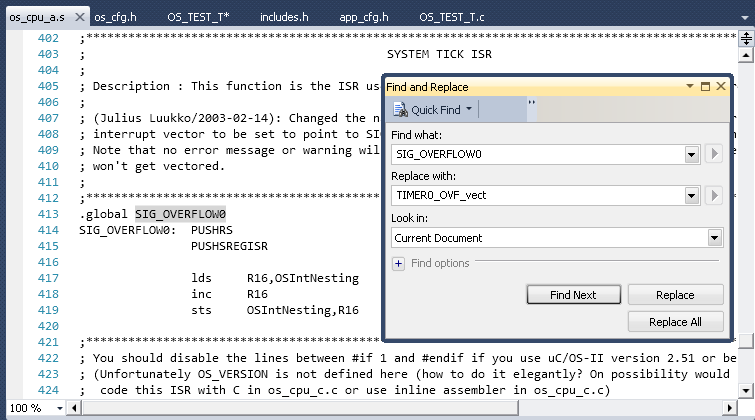


Figura 16. Reemplazando el nombre de la interrupción de desbordamiento del timer

1. En la sección de las barras de herramientas, al lado derecho del botón  de la parte de arriba, hay un menú, seleccionar la opción **Release.** Esto se ilustra en la figura 17.



Figura 17. Configuraciones de la solución

1. Para crear la solución del proyecto, seguir la ruta **Build>Build Solution.**
2. Una vez que concluido el proceso de construcción, procedemos a programar nuestro dispositivo físicamente o a crear un archivo en el programa Proteus ISIS para simular el microcontrolador. En el caso de la segunda, el resultado se puede apreciar en la figura 18.

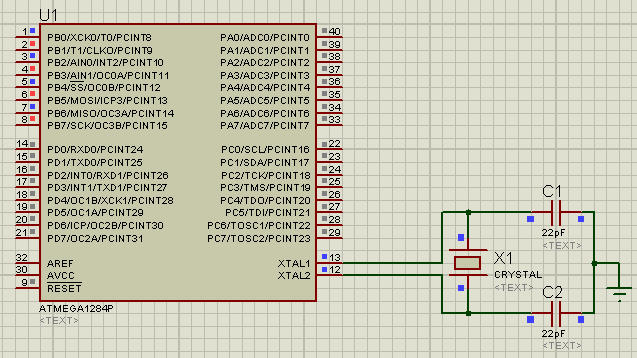


Figura 18. Simulación de la solución cargada en el ATMEGA1284P