# **Manual Uso de Auto Generador y *Framework***

# Inicio

El archivo ejecutable del auto generador de código se llama “Code Generator.jar” y se encuentra dentro de la carpeta principal del repositorio “Framework\_Testing” (figura 1)

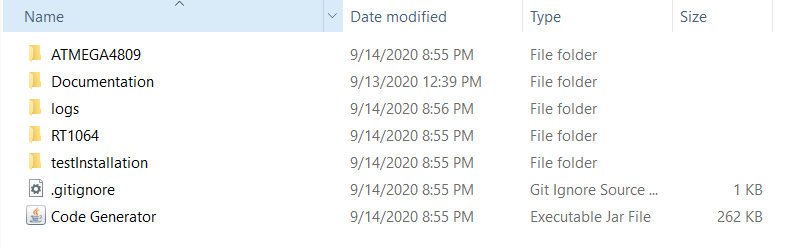


Figura 1. Archivo Ejecutable del Auto Generador de Código

El auto generador puede abrirse:

1. Desde la carpeta de documentos dando doble click sobre el archivo “Code Generator.jar”
2. Desde la línea de comandos, por medio del comando: Java -jar "Code Generator.jar"

En la figura 2 se muestra el auto generador de código. Cuando este inicia, no tiene ningún proyecto cargado y por lo tanto ninguna de las opciones está disponible.

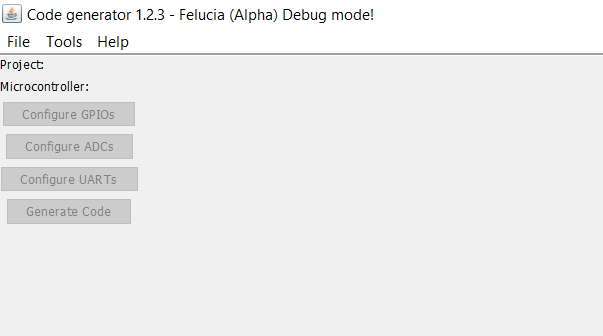


Figura 2. Auto Generador de Código

Para cargar un proyecto es necesario seleccionar la opción “Open Project” desde la pestaña de “File”.

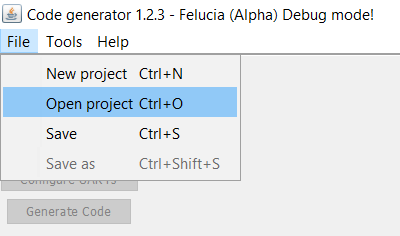


Figura 3. Abrir nuevo proyecto

Al seleccionar esta opción, el auto generador nos pide que seleccionemos el archivo de configuración (.cgs) que queremos utilizar (Figura 4). Cada microcontrolador tiene su propio archivo de configuración y estos pueden encontrarse dentro de la carpeta con el nombre del microcontrolador (Ex. ATMEGA4809)

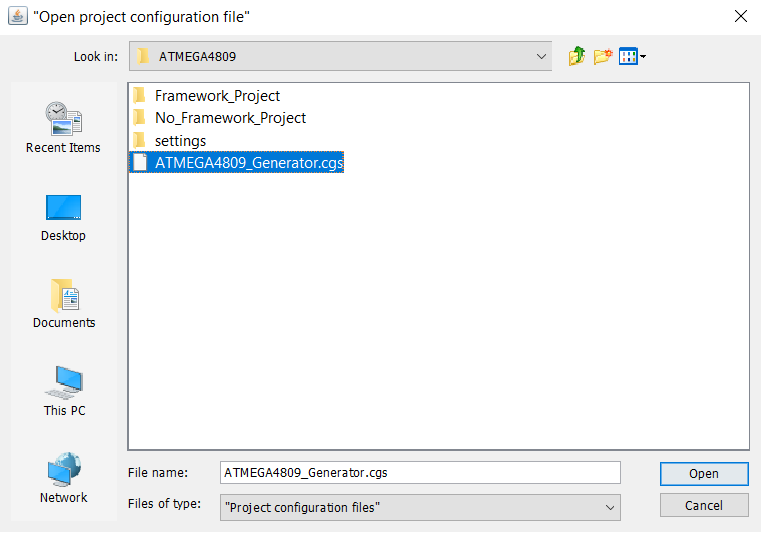


Figura 4. Selección de Archivo de Configuración

Una vez que se selecciona el archivo para el microcontrolador deseado, se debe hacer *click* en “Open” para que el auto generador utilice este archivo. Como puede verse en la figura 5, una vez que se carga un archivo de configuración se despliega el nombre del proyecto y del microcontrolador y las distintas opciones del auto generador ya aparecen disponibles.

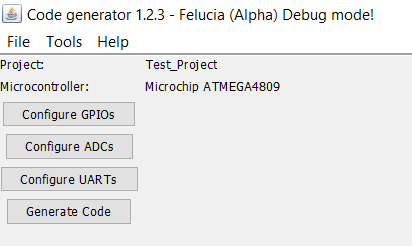


Figura 5. Auto Generador de código con archivo de configuración seleccionado

# Configuración de GPIO

Para configurar los *GPIO´s* del microcontrolador se debe hacer click en el botón “Configure GPIOs”. Al hacerlo se abrirá una ventana como la mostrada en la figura 6.

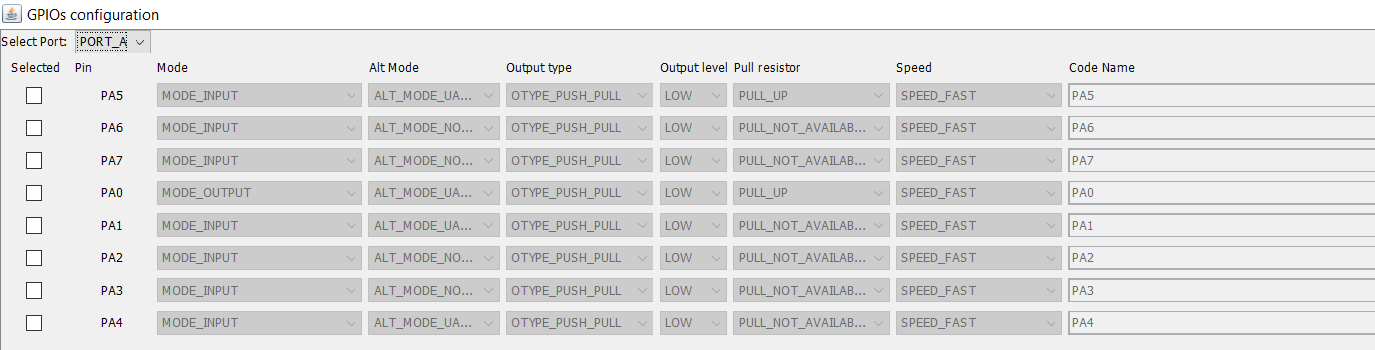


Figura 6. Ventana de Configuración de GPIO´s

Esta ventana nos deja seleccionar cada uno de los puertos del microcontrolador y realizar la configuración de cada uno de sus pines.

Dentro de los campos que se pueden configurar están:

* Si el GPIO será usado como entrada, salida o modo alterno si se utilizará el *GPIO* para algún periférico (Mode)
* La función alterna que usará el GPIO en caso de que el modo seleccionado sea alterno (Alt Mode). Los modos alternos soportados actualmente son Análogo y UART
* Tipo de Salida (Output Type)
* Velocidad del GPIO (Speed).
* Resistencia de Pull Up o Pull Down (Pull Resistor)
* Nivel inicial en caso de que el GPIO se haya configurado como salida (Output Level).
* Nombre que tendrá el GPIO en la aplicación (Code Name)

# Configuración de ADC

Para configurar los *ADC´s* del microcontrolador se debe hacer click en el botón “Configure ADCs”. Al hacerlo se abrirá una ventana como la mostrada en la figura 7.

En esta ventana se puede seleccionar y configurar cada uno de los *ADCs* que contiene el microcontrolador.

Dentro de los campos configurables están:

* Nombre para uso del ADC en la aplicación. (Code Name)
* Número de muestras a tomar (Samples)
* Señal de reloj (Clock Source)
* Justificación (Bits justification)
* Pre escalador para el reloj (Clock Prescaler)
* Resolución (resolution)
* Referencia de voltaje (Voltaje Reference)
* Selección de los canales a ser utilizados por el ADC y selección de un nombre para cada canal.

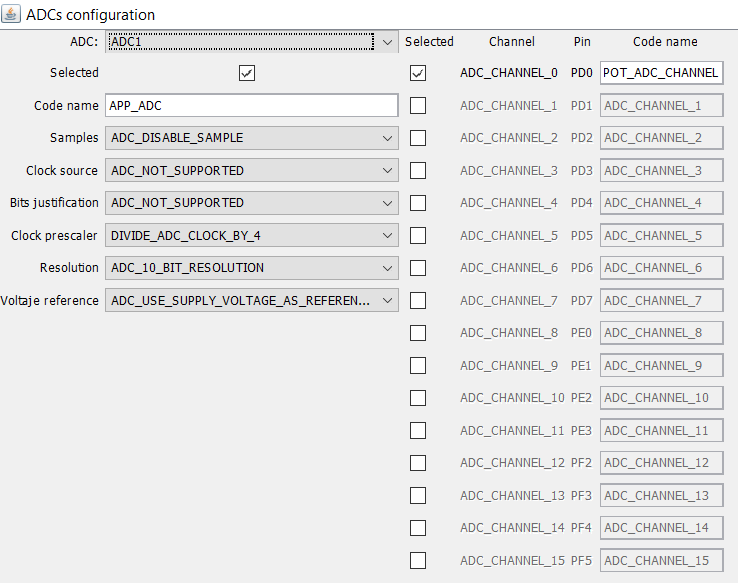


Figura 7. Ventana de Configuración de ADC

# Configuración de UART

Para configurar los *UART´s* del microcontrolador se debe hacer click en el botón “Configure UARTs”. Al hacerlo se abrirá una ventana como la mostrada en la figura 8.

En esta ventana se puede seleccionar y configurar cada uno de los *UARTs* que contiene el microcontrolador.

Dentro de los campos configurables están:

* Nombre para uso del *UART* en la aplicación (Code Name)
* Señal de Reloj (Clock Source)
* Pre escalador de la señal de reloj (Prescaler)
* Baud Rate (Debe ser ingresado a mano por el usuario)
* Bits de Datos (Data Bits)
* Bits de Paro (Stop Bits)
* Paridad (Parity)

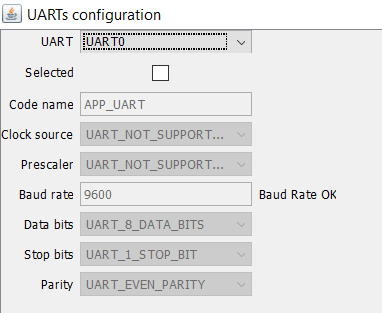


Figura 8. Ventana de Configuración de UART

# Generación de Código

Una vez que los periféricos se hayan configurado y se desee generar el código correspondiente, hay que dar click en el botón “Generate Code” y si no hay ningún error, una ventana como la de la figura 9 debe aparecer para indicar que el código ha sido generado exitosamente

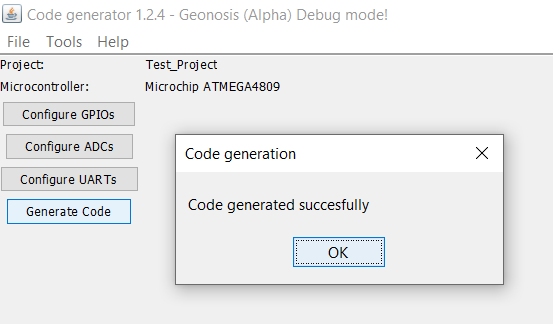


Figura 9. Generación Exitosa de Código

Los archivos auto generados son colocados directamente dentro del proyecto que hace uso del *framework* del microcontrolador seleccionado.

Los archivos generados para cada periférico son:

* adc\_cfg.c y .h para ADC (localizados en framework\drivers\adc\cfg)
* gpio.c y .h para GPIO (localizados en framework\drivers\gpio\cfg)
* uart\_cfg.c y .h para UART (localizados en framework\drivers\uart\cfg)

# Uso de *Framework*

En las figuras 10, 11 y 12 se muestran los pasos que deben seguirse para inicializar los *GPIOs, ADCs* y *UARTs* por medio del *framework*, ya sea por medio de los archivos auto generados (Elemento dinámico) o utilizando las funciones del elemento estático.

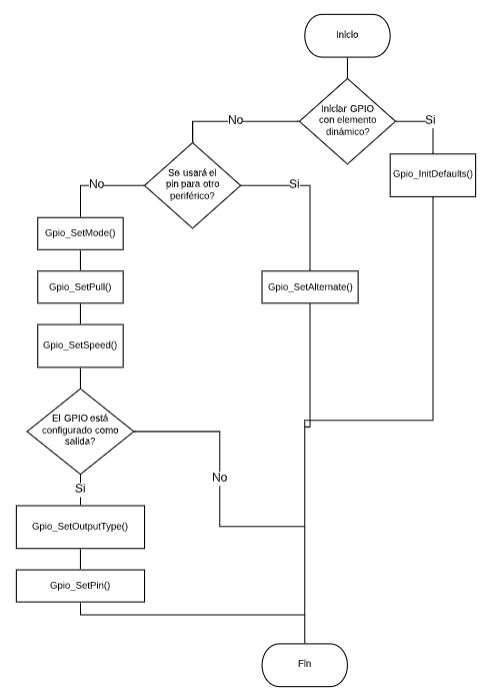


Figura 10. Inicialización de GPIO

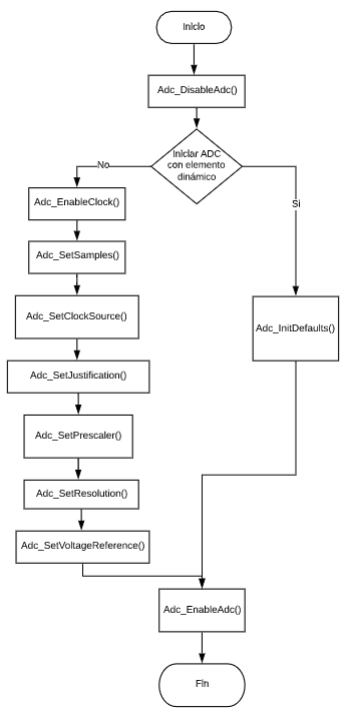


Figura 11. Inicialización de ADC

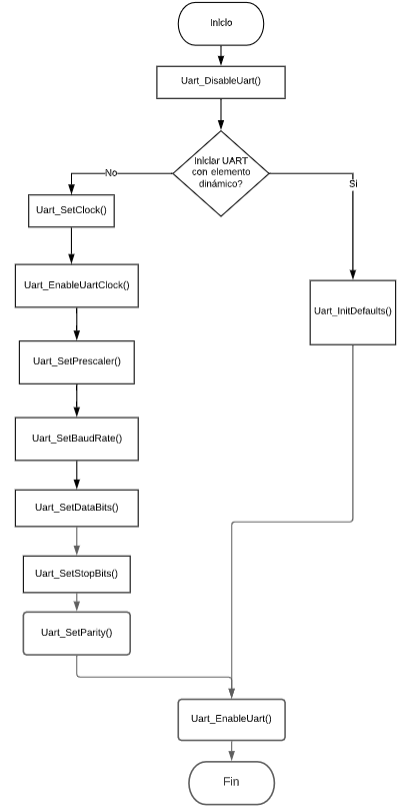


Figura 12. Inicialización de UART

El *framework* puede utilizarse para inicializar e interactuar con los distintos periféricos del microcontrolador. Actualmente los periféricos soportados son *GPIO, ADC* y *UART.*

A continuación, se muestran algunos ejemplos:



Figura 13. Inicialización de GPIOs a través del elemento dinámico



Figura 14. Iniciar conversión de ADC



Figura 15. Transmisión de un Byte por UART

Las distintas etiquetas que pueden ser utilizadas para interactuar con el *framework* se encuentran definidas en los *wrappers* de cada uno de los periféricos.