

Guia de configuración de un entorno de desarrollo de software para microcontroladores AVR de 8 bits usando avr-gcc.

Daniel Giovanni Martínez Sandoval

dagmtzs@gmail.com

Febrero 2024

Índice

1. Introducción	2
2. Requerimientos	2
3. Instalando herramientas de software	2
3.1. VisualStudio Code	2
3.2. AVR 8-bit Toolchain	3
3.3. AVRDUDE	3
3.4. Zadig	3
3.5. Variables de Entorno	3
3.6. Configuración adicional	4
4. Probando la configuración	4
4.1. Troubleshooting	5
5. Useful links	5

1. Introducción

La intención de esta guía es explicar el proceso de instalación de las herramientas necesarias para desarrollar software para los microcontroladores de 8 bits de AVR. La guía y los ejemplos que se proveen son particular, pero no exclusivamente, orientados al ATmega328P. Esta guía está pensada para ayudar, pero no para resolver todos los problemas que podrían aparecer durante la configuración del entorno, de tal manera que algunas desviaciones pueden ser necesarias para hacer que todo funcione dependiendo de tu sistema y los dispositivos que decidas usar, sin embargo, dicha información puede ser encontrada fácilmente en internet. La documentación para el software y hardware mencionados puede ser encontrada al final del mismo.

Para facilitar el reconocimiento de nombres de archivos y rutas de archivos, todos estos estarán coloreados en [verde](#), los enlaces a páginas web estarán coloreados en [azul](#).

2. Requerimientos

Esta guía está hecha pensando en laptops ejecutando Windows 10 (o superior). Si bien no se han definido requerimientos mínimos del sistema, casi cualquier sistema moderno (probablemente del 2015 en adelante), estando en buenas condiciones, debería servir. En cuanto al hardware, se necesita lo siguiente:

- Un programador que sea soportado por avrdude, en mi caso, el USBasp.
- Un microcontrolador de 8 bits de AVR, en mi caso, el ATmega328P.
- La circuitería que requiera el microcontrolador para funcionar. Para el ATmega328P yo usé un protoboard, un cristal oscilador de 16 MHz, dos capacitores de 22 pF, un pushbutton, un resistor de 10 kΩ y algo de alambre para el protoboard.

3. Instalando herramientas de software

Para configurar un entorno de desarrollo básico, yo recomendaría las siguientes herramientas de software:

- Un editor de texto, preferentemente uno orientado a desarrollo de software en C.
- Un compilador y sus herramientas relacionadas y librerías.
- Un programa cargador/programador.

En una computadora con Windows, mi preferencia es por VS Code como editor de texto, las herramientas de AVR GNU y AVRDUDE como programador

3.1. VisualStudio Code

Descárgalo de [este enlace](#) y sigue las instrucciones del instalador.

3.2. AVR 8-bit Toolchain

Este conjunto de herramientas se puede descargar del sitio de Microchip en [este enlace](#). Selecciona la descarga que dice **AVR 8-Bit Toolchain (Windows)** y una vez completada, realiza los siguientes pasos:

1. Da click derecho en el archivo ZIP descargado y selecciona *Extraer todo*.
2. En la nueva ventana, deja las opciones por defecto y da click en *Extraer*.
3. Cambia el nombre de la carpeta resultante a **avr8-gnu-toolchain**.
4. Mueve la carpeta a **C:\Archivos de Programa**.

3.3. AVRDUDE

Este programador se puede descargar de [este enlace](#). Y de la misma manera que con el set de herramientas de AVR:

1. Da click derecho en el archivo ZIP descargado y selecciona *Extraer todo*.
2. En la nueva ventana, deja las opciones por defecto y da click en *Extraer*.
3. Cambia el nombre de la carpeta resultante a **avrdude**.
4. Mueve la carpeta a **C:\Archivos de Programa**.

3.4. Zadig

Este programa instala el driver necesario para usar el programador USBasp. Si no vas a usar este programador, puedes omitir esta instalación, pero deberás revisar si necesitas algún otro driver. En mi configuración, esto es indispensable, así que descárgalo de [este enlace](#) y después de descargar sigue estos pasos:

1. Conecta tu USBasp.
2. Da doble click a la aplicación descargada.
3. Puede que necesites permisos de administrador para permitirle al programa instalar el driver, acepta.

4. Debería abrirse una ventana con un menú desplegable en el cuál debe estar seleccionada la opción **USBasp**, de otra forma, revisa la conexión del USBasp o busca ayuda en internet.
5. Si todo está bien hasta este punto, da click en **Install WCID**.

3.5. Variables de Entorno

Para poder usar el compilador y programador que acabas de instalar, es necesario actualizar las Variables de Entorno¹ de la siguiente manera:

1. Haz click en el menú de Windows (o presiona la tecla de Windows en tu teclado) y escribe *variables*.
2. Uno de los resultados debe ser *Edit the system environment variables*, haz click en él.
3. En la nueva ventana, titulada *System Properties* da click en el botón *Environment Variables*, en la parte de abajo de la ventana.
4. En la siguiente ventana, deberías ver dos campos llamados *User variables for ...* y *System variables*. En *User variables for ...*, busca una variable llamada **Path**.
 - a) Si no tienes una variable llamada **Path**, haz click en *New...*
 - 1) En la ventana *New User Variable*, en el campo *Variable name*, escribe **Path**
 - 2) En el campo *Variable value*, escribe `C:\Program Files\avr8-gnu-toolchain\bin`.
 - 3) Haz click en *OK*.
 - b) Si ya tienes una variable **Path**, selecciónala y haz click en *Edit...*
 - 1) En la nueva ventana, haz click en *New*.
 - 2) En el espacio activo, escribe `C:\Program Files\avr8-gnu-toolchain\bin`.
5. Una vez que el Toolchain de AVR ha sido agregado a la variable **Path**, repite el paso 4b para agregar la carpeta AVRDUDE de tal modo que tu variable **Path** contenga estos dos directorios:
 - `C:\Program Files\avr8-gnu-toolchain\bin`
 - `C:\Program Files\avrdude\`
6. Puede que sea necesario que salgas de tu sesión de usuario y vuelvas a entrar, o en el peor de los casos, que tengas que reiniciar tu computadora para que las variables de entorno se actualicen.

3.6. Configuración adicional

Para hacer del flujo de trabajo una experiencia más amigable, recomendaría seguir las instrucciones de [este artículo](#) en la sección *Edit C/C++ Configuration* para añadir algunas funcionalidades bastante útiles a VS Code.

¹Nótese que esa guía se creó con base en un sistema con Windows en inglés, si tu sistema está en español, tal vez debas poner especial atención en buscar los nombres equivalentes para lo aquí mencionado.

4. Probando la configuración

Una vez que el entorno haya sido configurado, puedes seguir estos pasos para crear, compilar y cargar un programa a tu microcontrolador:

1. Crea una carpeta para guardar tus proyectos de AVR, dentro de ella, crea una carpeta para este programa de ejemplo.
2. Ejecuta VS Code y abre la carpeta que acabas de crear. (File ->Open Folder).
3. En el explorador de archivos de VS Code, crea un nuevo archivo y nómbralo: `main.c`.
4. De mi página de GitHub, abre el ejemplo de HelloWorld en [este enlace](#).
5. Copia mi código y pégalo en el archivo que acabas de crear en VS Code.
6. Guarda el archivo.
7. Conecta el USBasp a tu microcontrolador (y a tu computadora, si no está conectado aún).
8. En el explorador de archivos de VS Code, da click derecho a la carpeta en la que estás trabajando y selecciona la opción *Open in Integrated Terminal*, o en el menú **Terminal** en la opción **New Terminal**.
9. En la Integral Terminada, ingresa uno por uno, los siguientes comandos:
 - `avr-gcc main.c -mmcu=atmega328p -o main.elf`
 - `avr-objcopy -O ihex main.elf main.hex`
 - `avrdude -p atmega328p -c usbasp -U flash:w:main.hex`

El resultado debería ser que tu microcontrolador quede programado con ese código.

4.1. Troubleshooting

Here are some things that can make this process fail:

- Your system might not recognize the commands you write immediately, you can try restarting programs, logging out and back in to your Windows user account or if that doesn't work, you might restart your computer.
- Typos might prevent your commands from working, or even might cause damage to your microcontroller, double check the commands you write.
- One of the most common problems are wires and connections. When connecting each signal manually:
 - Take special care in labeling or color-coding your long wires so you don't mix them up.
 - Make sure you identify correctly the pins of your microcontroller.
 - Check the values of your crystal oscillator and capacitors.
 - Check specially the power connections to the programmer, and make sure they are close to the microcontroller, i.e., try to minimize the length of the power lines to the microcontroller power pins.
 - Sometimes the uploading speed might be too high for certain setups, so you can append to the avrdude command the option `-B125kHz`, or even `-B32kHz`.

5. Useful links

- [AVRDUDE User Manual](#)
- [AVR Libc Home Page](#)
- [avr-gcc Toolchain Wiki](#)
- [AVRDUDE Home Page](#)
- [GCC Online Documentation](#)