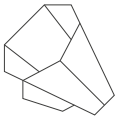


Introduction

When we are working with micro controllers (MCUs) we often would like to have the possibility to use the functions in the standard `stdio.h` library. This library makes it possible to communicate with a console using functions like:

```
puts(…)
printf(…)
getchar()
scanf(…)
Etc.
```

This small document describes how this can be made possible.

You will need a serial terminal

- HTerm (http://www.der-hammer.info/terminal/) it can be used in nearly all situations where you want to work with serial communication. My preferred terminal ☺
- Another possibility is to install a terminal extension to Atmel Studio (https://www.microchip.com/webdoc/terminalwindow/index.html)
- Other serial terminals ca be used

The serial terminal must be setup to use the COM port that is connected to the MCU. It can be found in the *Device Manager* on you PC.

The serial setting must be: **57600, N, 8, 1**

You will also need a standard IO driver library. This library consists of a header file and a static library file that must be linked together with your application. The files needed are

- `stdio_driver.h`
- `libserialLib.a`

These files can be found in *Stdio_lib.7z*.

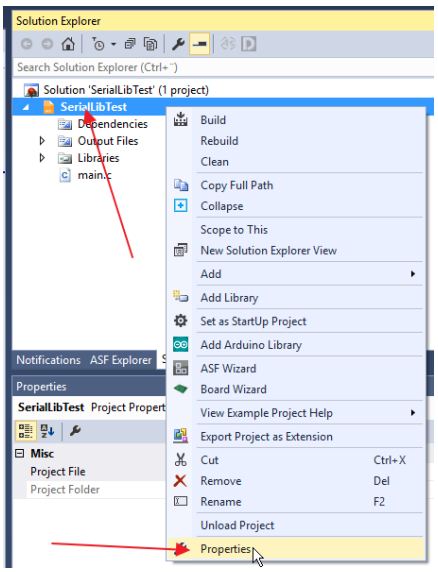
Setting up a project to use STDIO

In this section it is described how a project must be setup before it can use STDIO.

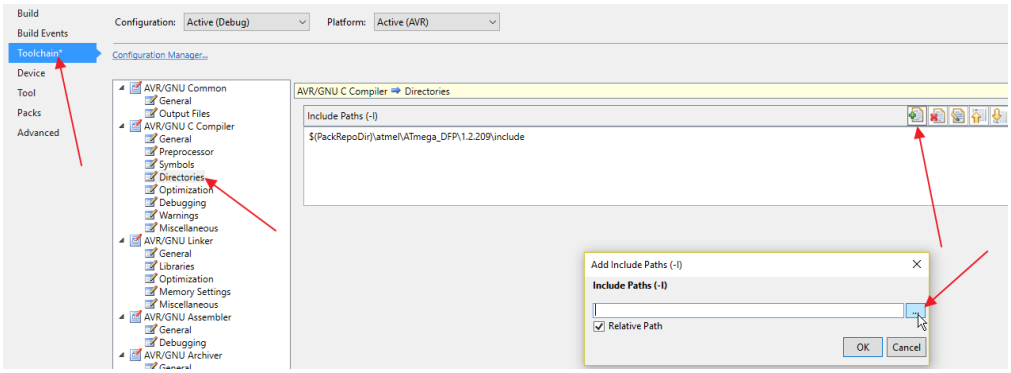
First you must unpack *Stdio_lib.7z* in a folder on your computer. In this example I have unpacked the files in *D:\Tmp\Lib*.

Next step is to do some configuration of the Atmel Studio project.

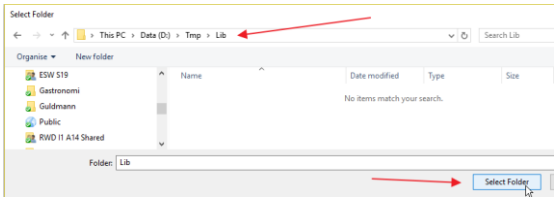
Right click on your project and select *Properties*

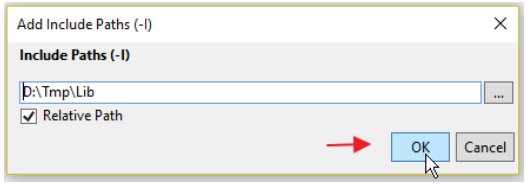


Then select *Toolchain*->Directories* and then the little green + icon  and browse to the folder where you have unpacked the files.



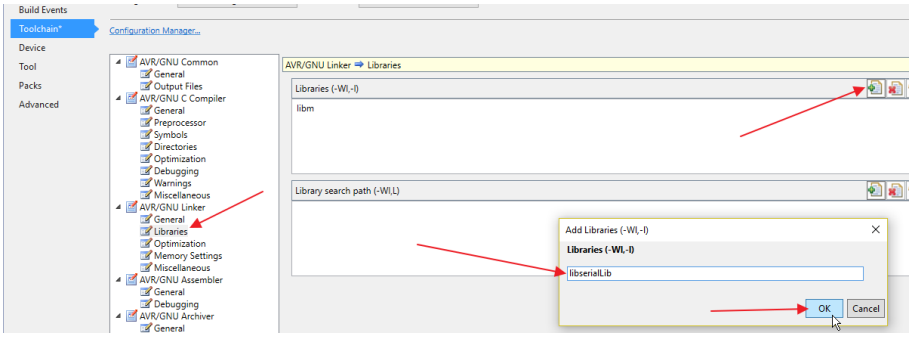
In my example it is *D:\Tmp\Lib* and *Select Folder*





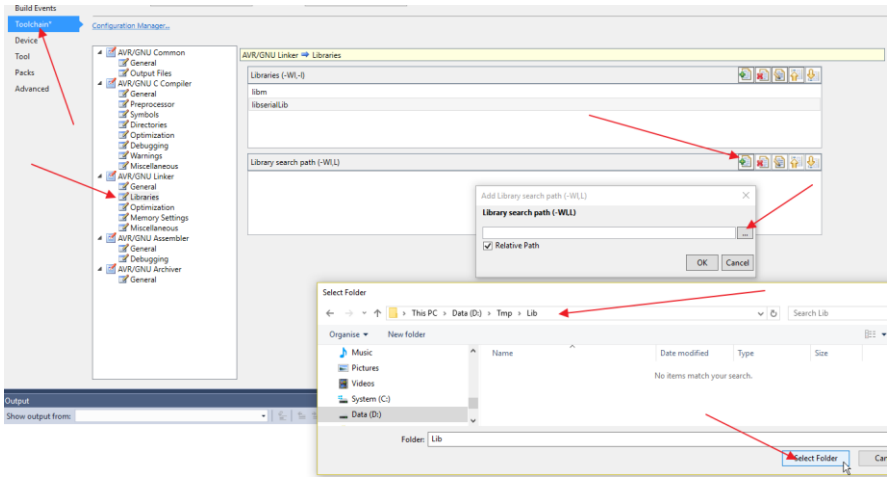
Next step is to tell the Linker that we want the static library linked in to our application.

Select *Toolchain*->Libraries* and then the little green + icon  and type *libserialLib* and *OK*.



Next step is to tell the linker in which folder it can find the library.

Select *Toolchain*->Libraries* and then the little green + icon  in *Library search path* and browse to the folder where you have unpacked the files in my example *D:\Tmp\Lib*.



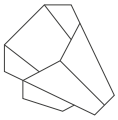
Now your project can use the STDIO library.

Example file using stdio.h

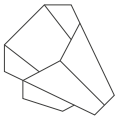
The lines marked with red arrows ⟵ are needed to be able to use the *stdio.h* functions.

The lines marked with green arrows ⟵ shows examples of *stdio.h* functions.

Note: The function *stdioInputWaiting()* I have implemented to make it possible to peek if there are any characters waiting in the input buffer. This can make it possible to avoid blocking calls to *stdio.h* functions like *getchar(), scanf(…)* etc.



```c
#include <stdio.h>
#include <stdio_driver.h>

#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

int main(void)
{
    stdioCreate(0); // Use USART 0 for stdio
    sei(); // Enable interrupt

    puts("Program Started!");

    uint16_t counter = 0;

    while (1)
    {
        printf("The counter value: %05d and in hex: %04X\n", counter, counter);
        counter++;

        if (stdioInputWaiting())
        {
            printf("###>%c\n", getchar());
        }

        _delay_ms(500);
    }
}
```

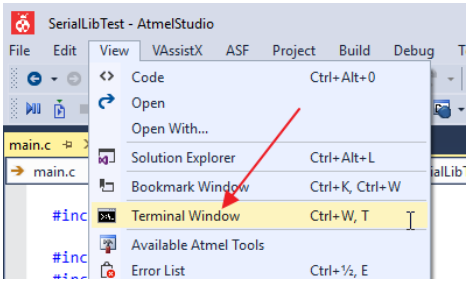


Terminal Extension in Atmel Studio

This is not necessary if you use HTerm!

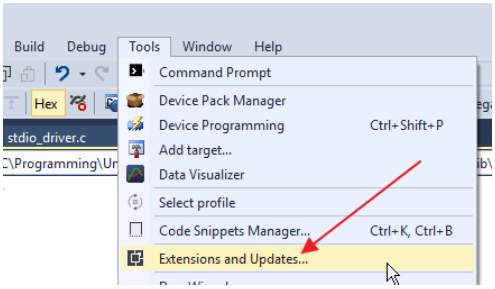
Here is a description of the installation process for the extension.

First check if you already have it. If you find *Terminal Window* in the *View menu* you are good to go and do not need to do more.

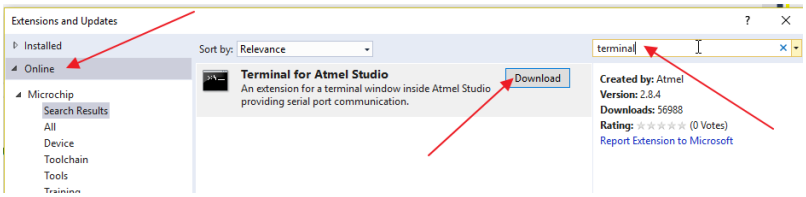


If not, you need to install the extension just follow these steps:

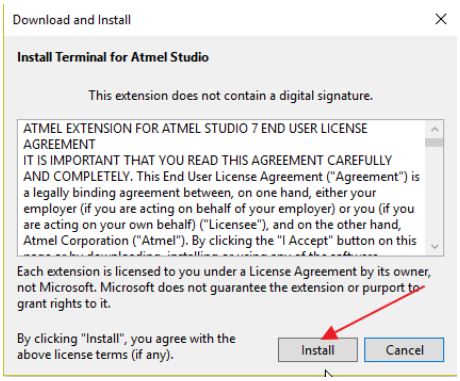
Select *Extensions and Updates…* in the *Tools menu*



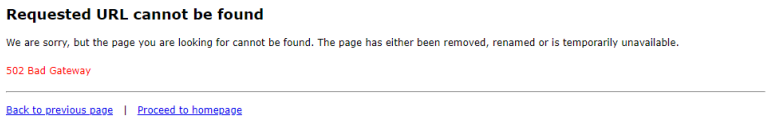
Select *Online*, search for *terminal* and *Download*.



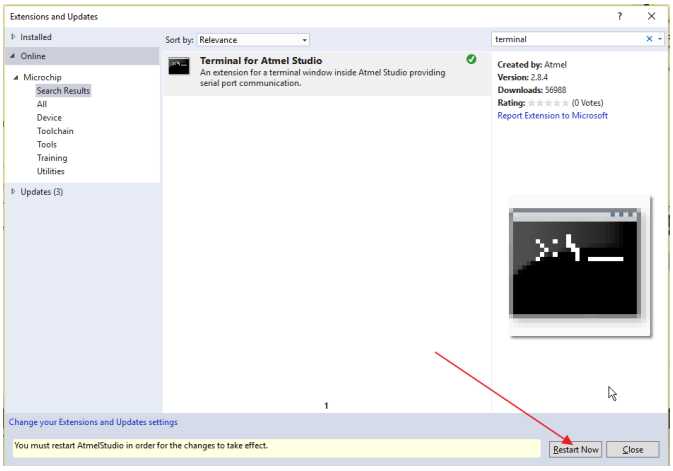
Accept the license by clicking on *Install*



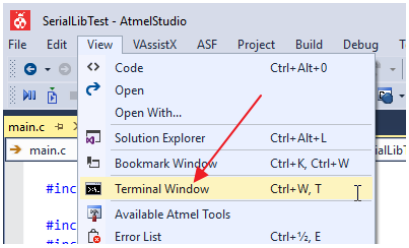
You will properly see something like this, you can just ignore it



Click *Restart Now* to restart Atmel Studio



Now the *Terminal Window* should be available in the *View Menu*



FAQ

Why do I get the following error message when I try to upload my program?

```
avrdude.exe: stk500v2_ReceiveMessage(): timeout
avrdude.exe: ser_send(): write error: sorry no info avail
avrdude.exe: stk500_send(): failed to send command to serial port
avrdude.exe: ser_recv(): read error: The handle is invalid.
```

If you use the USB connection to program the Arduino board, the programmer and terminal use the same COM port. Click *Disconnect* on the terminal before you try to upload any programs, and then click *Connect* after the upload has succeeded.