Setup Microchip Studio 7 Programmer to access Arduino board.

Install Microchip Studio

Download and install Microchip Studio from the following link:
https://ww1.microchip.com/downloads/aemDocuments/documents/DEV/ProductDocuments/SoftwareTools/as-installer-7.0.2594-web.exe

When selecting architecture only the AVR is needed as seen Figure 1 (However feel free to install the others as well)

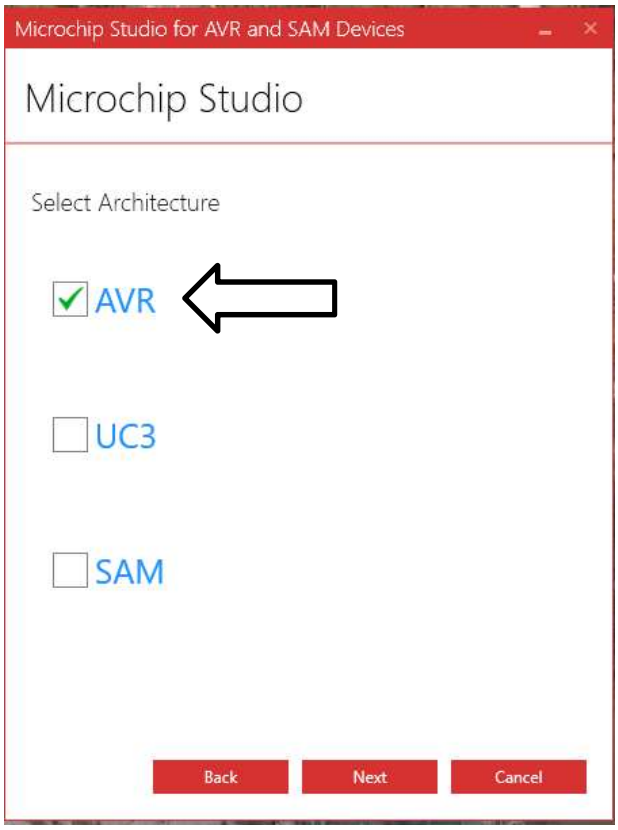
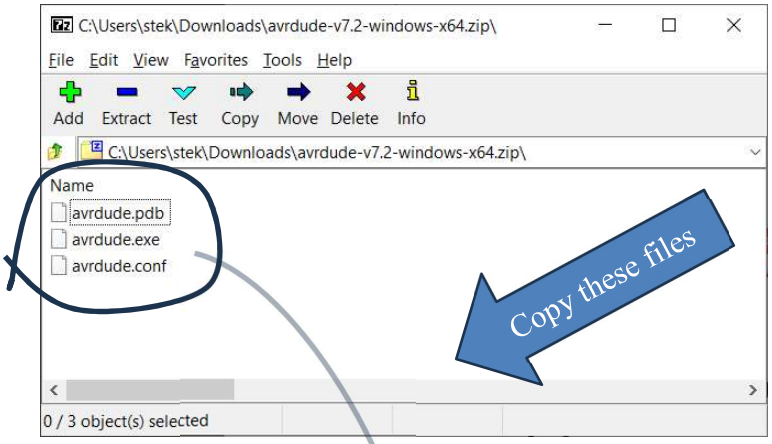


Figure 1: selecting Architecture

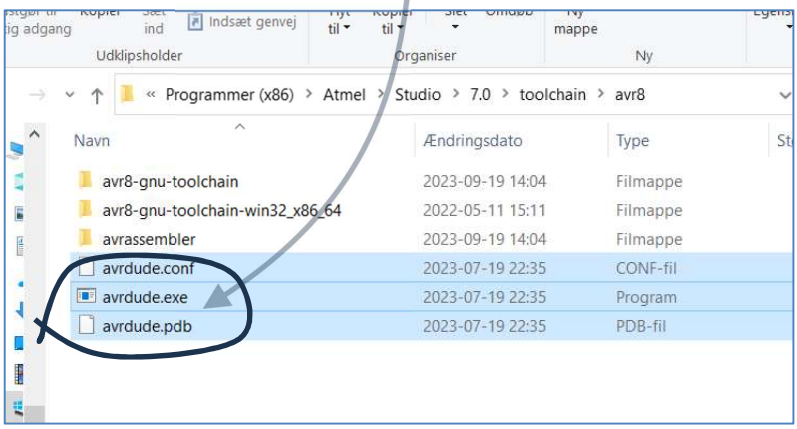
Setup programmer with Microchip Studio

To be able to program Arduino boards from Microchip Studio 7 (Powered by Visual Studio) it is necessary to install a separate Application/Programmer. The Arduino MEGA2560 has an *In System Programmer (ISP)* that are compatible with *avrdud*e which you can download here:
https://github.com/avrdudes/avrdude/releases/download/v7.2/avrdude-v7.2-windows-x64.zip

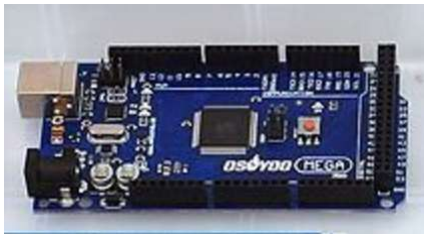
Download the AvrDude file to your PC download folder. Open the the zip so you can see the contents (there's 3 files inside). It looks like this in 7-Zip:



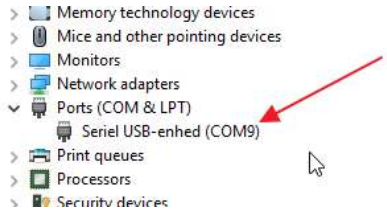
Move the 3 filew to the directory where you want to put them. One suggestion would be to put them in the folder where Microchip Studio was installed, e.g.
`C:\Program Files (x86)\Atmel\Studio\7.0\toolchain\avr8` like this:



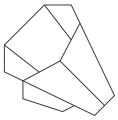
Connect your Arduino board to an USB port on your PC using the blue USB cable. **To make the Microchip studio setup work you must always use the same USB connection (port).**



When the Arduino board is connected to your PC the COM port used will show up in the Windows Device Manager (Hint: Press Windows-key + X, Device manager, Ports). In my case the Device Manager display looks like this:



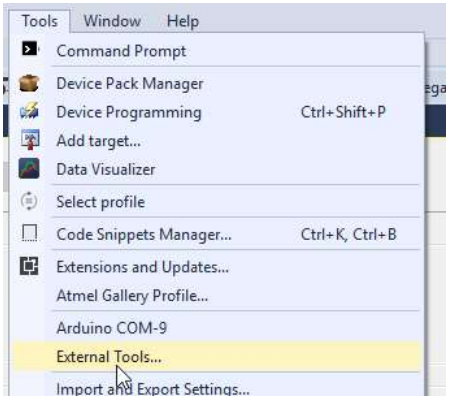
In this case it uses **`COM9`**



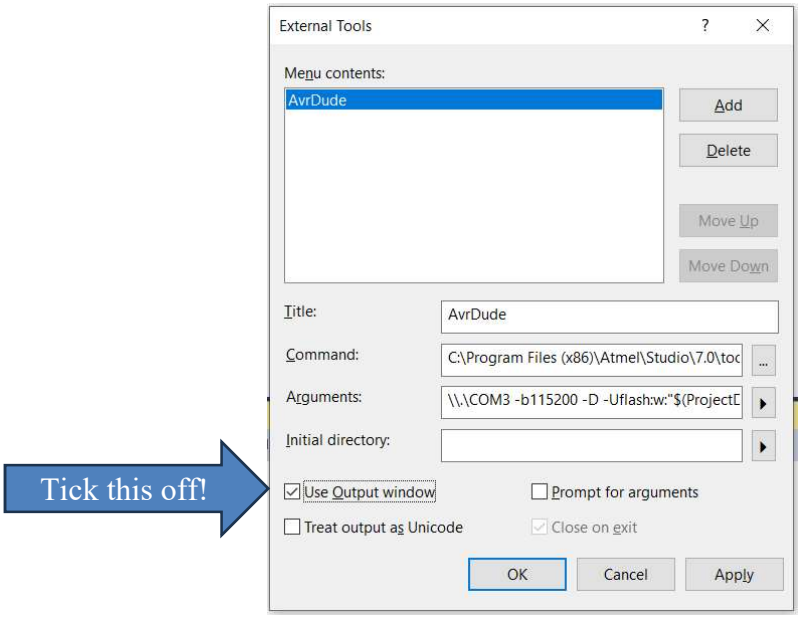
Now we are ready to integrate your Arduino board into the Microchip Studio development environment.

Startup microchip studio and select: File, New, Project, choose “Assembler”, “AVR Assembler Project” and press OK button. In the Search frame write “ATmega2560” and then select the ATmega2560 and press OK button. A small default program in Assembler will show up.

Setup the application/programmer as an external tool in Microchip Studio



Add your new external tool Title (e.g AvrDude) like this



The Command field must point to *avrdude.exe*. In this case it is `C:\Program Files (x86)\Atmel\Studio\7.0\toolchain\avr8\avrdude.exe`

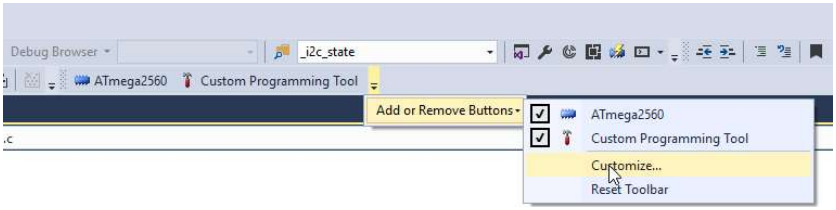
The Arguments frame contents should look like this:

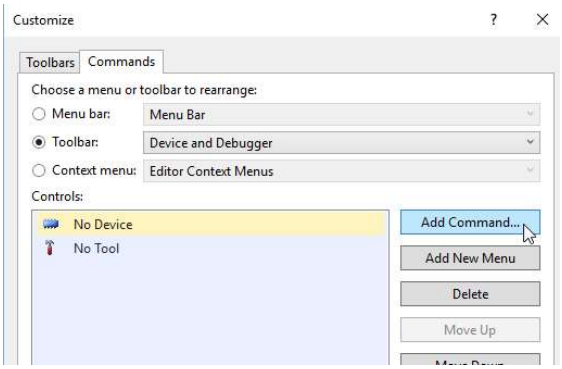
```
-C"C:\Program Files (x86)\Atmel\Studio\7.0\toolchain\avr8\avrdude.conf" -patmega2560 -cwiring -P\\.\COM3 -b115200 -D -Uflash:w:"$(ProjectDir)Debug\$(TargetName).hex":i
```

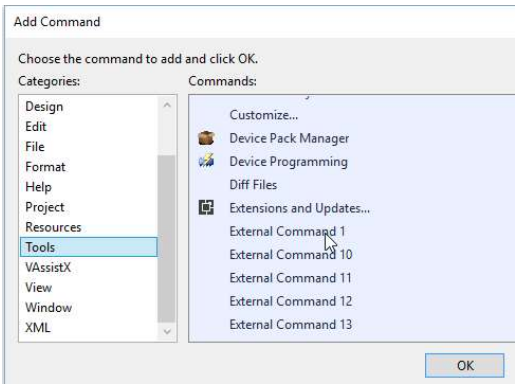
Where the red text should match your installation of *avrdude* and the COM-port the Arduino board is connected to. (Cut and Past the Arguments frame above and edit the path and COM port so it matches your settings – when you paste it in it’s a single long line).

Press “Apply” and “OK”.

To make the new programming tool easier to use it is possible to add it to the tool bar in Microchip Studio.

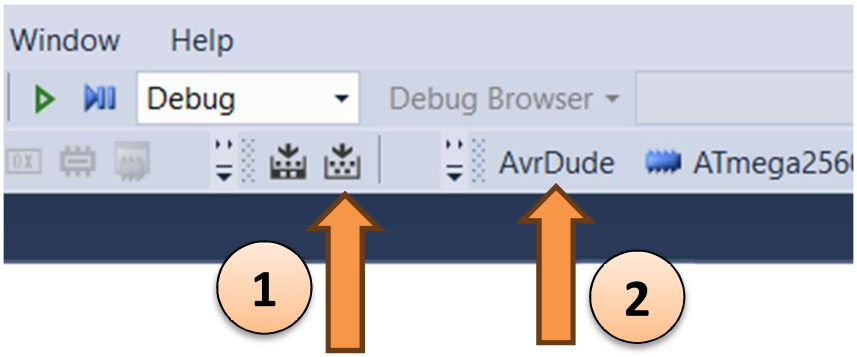






After that, your Arduino board can be accessed from the tool bar

Now it is possible to program your Arduino board from Microchip Studio by pressing the new button (AvrDude) you just need to transfer the program to the Arduino board.



Program your Arduino with microchip studio

Paste the code in the textbox below into main.asm (removing anything else). Build it (F7), and program it to the Arduino by pressing on the AvrDude button. You can also do it like this:

First click the “Build Solution” button and then the “AvrDude” button. If all goes well our small assembly program gets compiled, converted into a binary file with machine code instructions and downloaded into the Arduine/AtMega2560.

When the program runs it blinks the red LED once a second.

```
start:
ldi r16, HIGH(RAMEND)
out SPH, r16
ldi r16, LOW(RAMEND)
out SPL, r16
ldi r16, 0xff
out ddrb, r16
call delay
ldi r16, 0
out portb, r16
call delay
ldi r16, 0xFF
out portb, r16
rjmp start


delay:
ldi r16, 82
l1:
ldi r17, 255
l2:
ldi r18, 255
l3:
dec r18
brne l3
dec r17
brne l2
dec r16
brne l1
ret
```