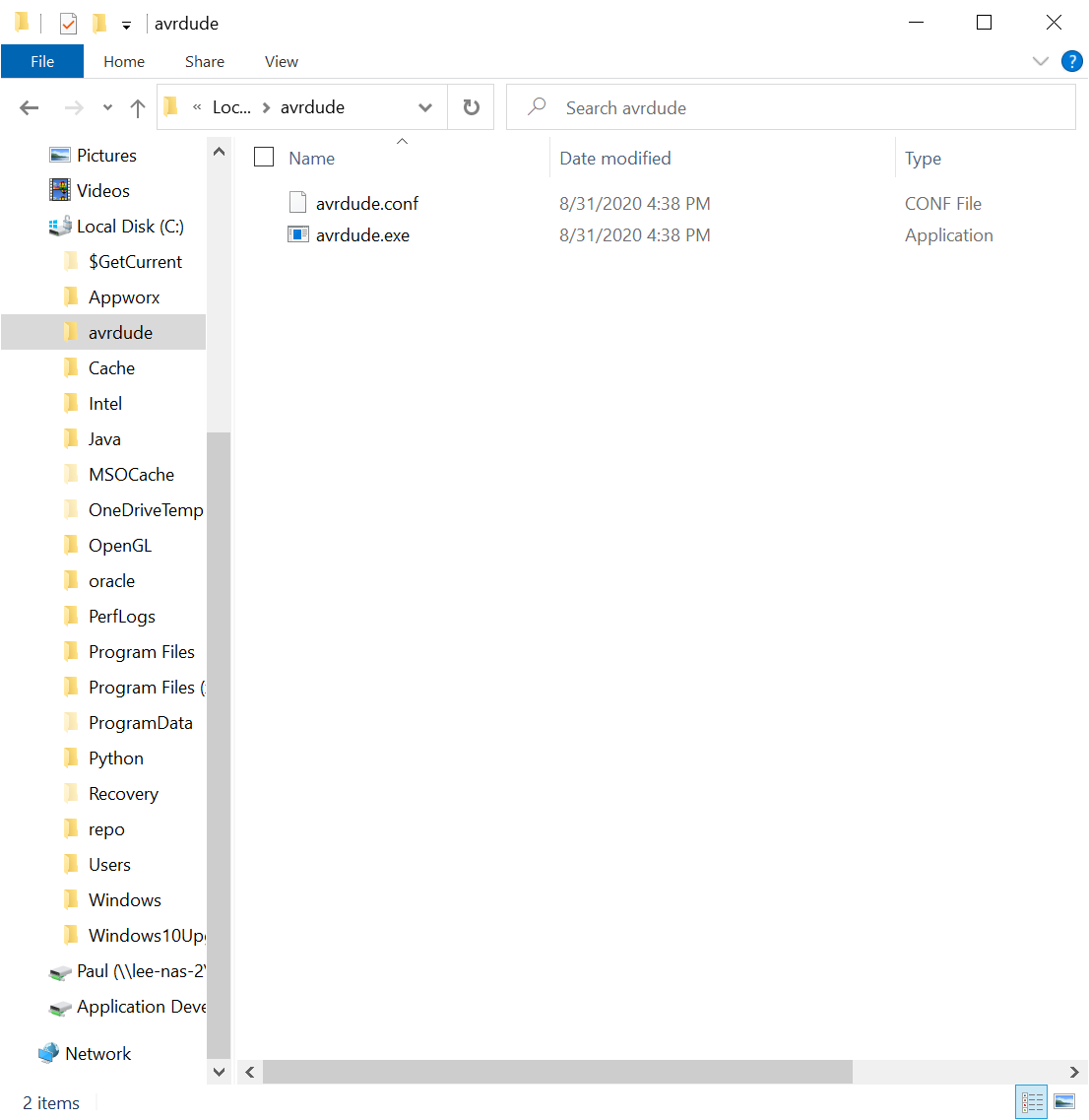
Install Atmel Studio 7.0

* Navigate to the Microchip.com Embedded Software Center:
* <https://www.microchip.com/mplab/Embedded-Software-Center>
* Choose link for Atmel Studio
* Click on link for Atmel Studio 7.0 Web Installer
  + Save the install executable to you downloads directory
* Run the installer for Atmel Studio 7.0.x
* You can accept all of the defaults
  + Select Architecture
    - This course will only use AVR 8-bit MCU
    - you may install AVR 32-bit MCU and SMART ARM MCU if you want

Install and setup AVRDUDE with Atmel Studio 7.0

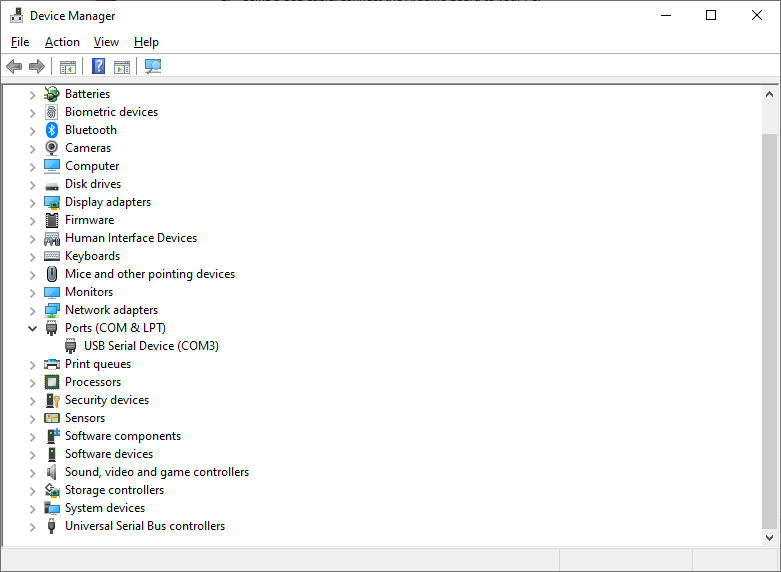
Install AVRDUDE

* Download avrdude.zip from Canvas->Files\Resources
* Extract the zip file to a location on your computer, i.e. C:\



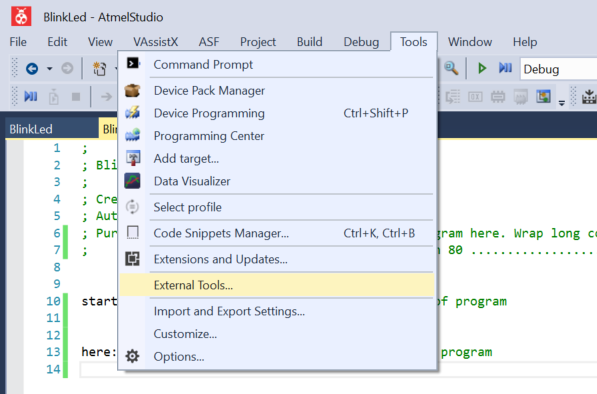
Check your COM Port

* Plug your Arduino into your computer with an USB cable
* 3 quick ways to show the Desktop | Microsoft Windows Tips | Harlow,  Bishop's Stortford, Essex, Hertfordshire | EasykeyPress your **Windows logo key** and type “Device Manager”
* Make note of the COM port associated with the USB listed (may have different names)



Setup AVRDUDE in Atmel Studio 7.0

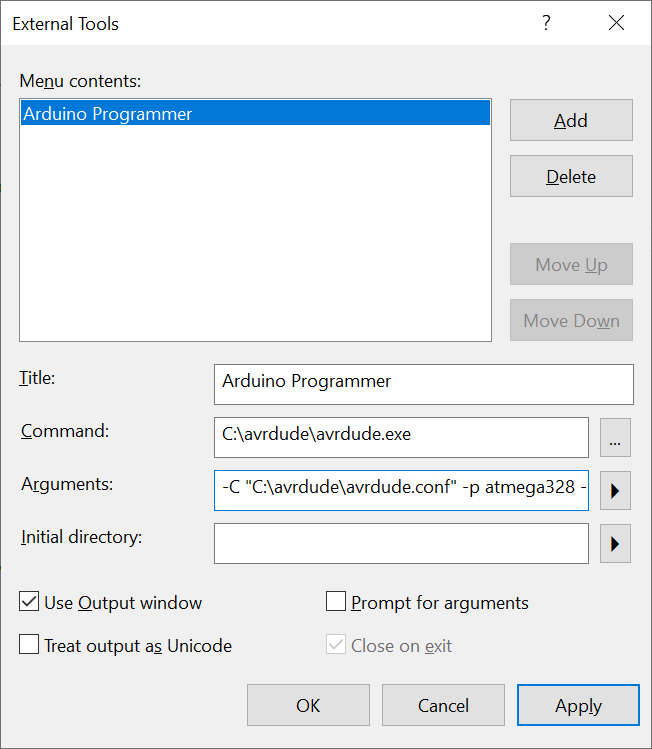
* Menu: Tools\External Tools…



* Configure Arduino Programmer
  + Title: replace “[New Tool 1]” with “Arduino Programmer”
  + Command: Path to avrdude.exe, e.g. “C:\avrdude\avrdude.exe”
  + Arguments:
    - Based on location of AVRDUDE and COM port:

-C "C:\avrdude\avrdude.conf" -p atmega328p -c arduino -P COM3 -b 115200 -U flash:w:"$(ProjectDir)Debug\$(TargetName).hex":i

* + Use Output window: Checked



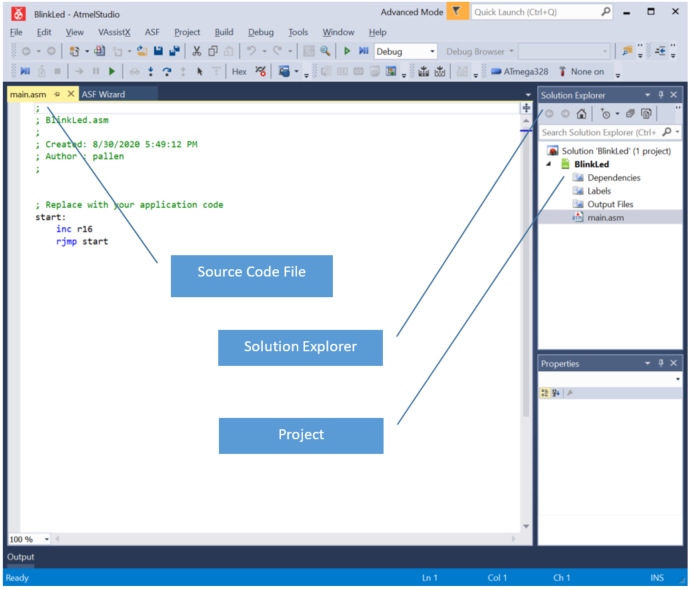
* **Click OK**

Create a New Project in Atmel Studio 7.0

* Start a New Project
  + Start Page – click New Project
  + File Menu – click File\New\Project
* New Project
  + Installed – select Assembler
  + Select AVR Assembler Project
  + Name – Enter a project name, i.e. “BlinkLed”
  + Location – enter a path where the project folder will be saved
    - Default is an Atmel Studio\7.0 directory in your Documents folder with path:
      * c:\users\<username>\Documents\Atmel Studio\7.0
  + Solution name: Leave default or Blank
    - Uncheck “Create directory for solution”
    - This should grey-out the Solution name entry box
  + Click OK
* Device Selection
  + Change “Device Family” selection box to “ATmega”
  + Browse for and select “ATmega328”
    - Do not select ones with extensions “P” or “PB”
  + Click OK

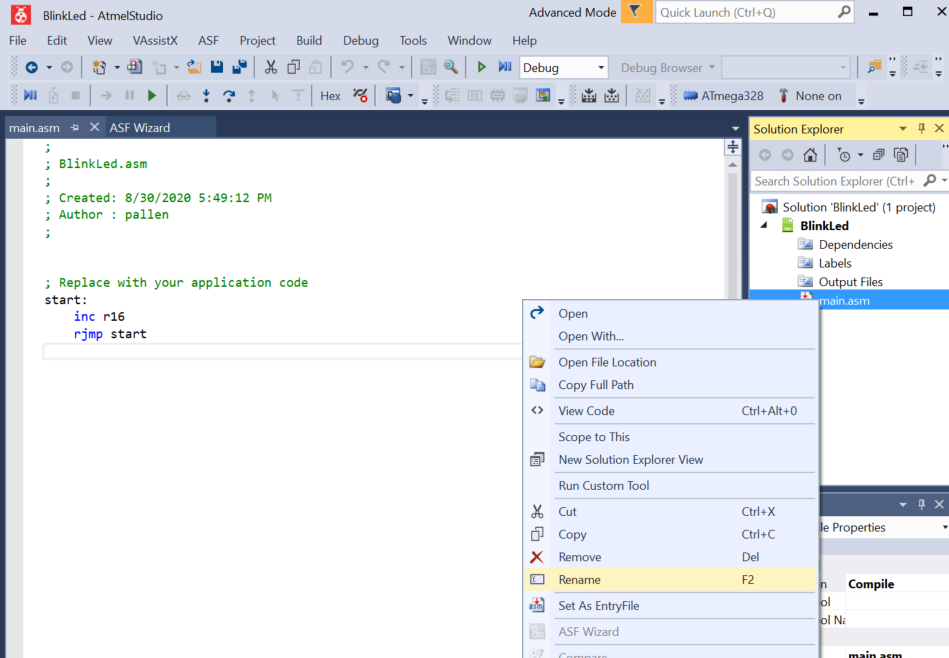
New Project Setup

Your Atmel Studio window should look similar to the following:

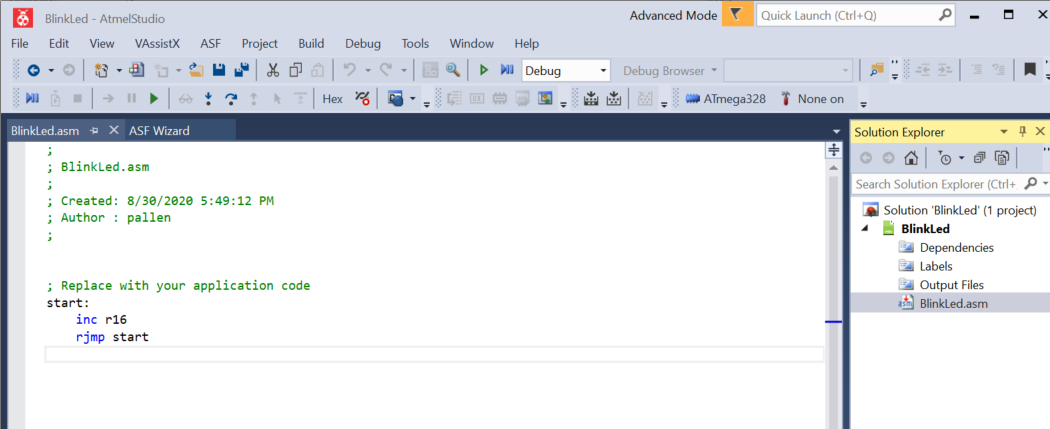


Rename “main.asm” to your project name:

* Right-click “main.asm” in Solution Explorer and choose Rename
* Edit the source code file name to match your project, e.g. “BlinkLed.asm”

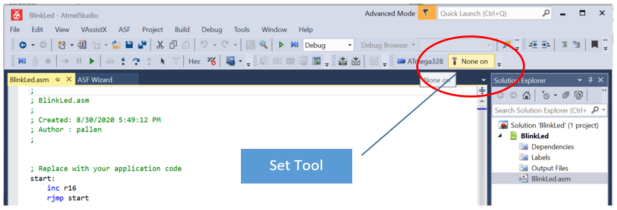


After:

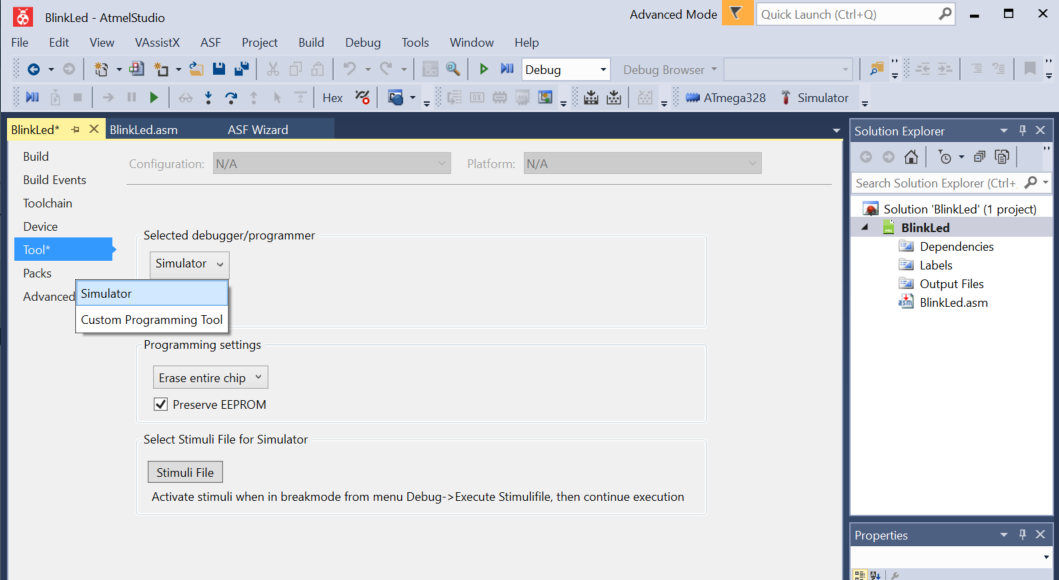


Set the Simulator Debugger/programmer Tool

* Click the Tool Icon in the toolbar

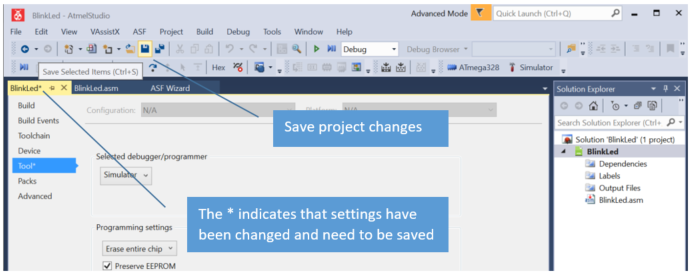


* Select the Simulator debugger/programmer



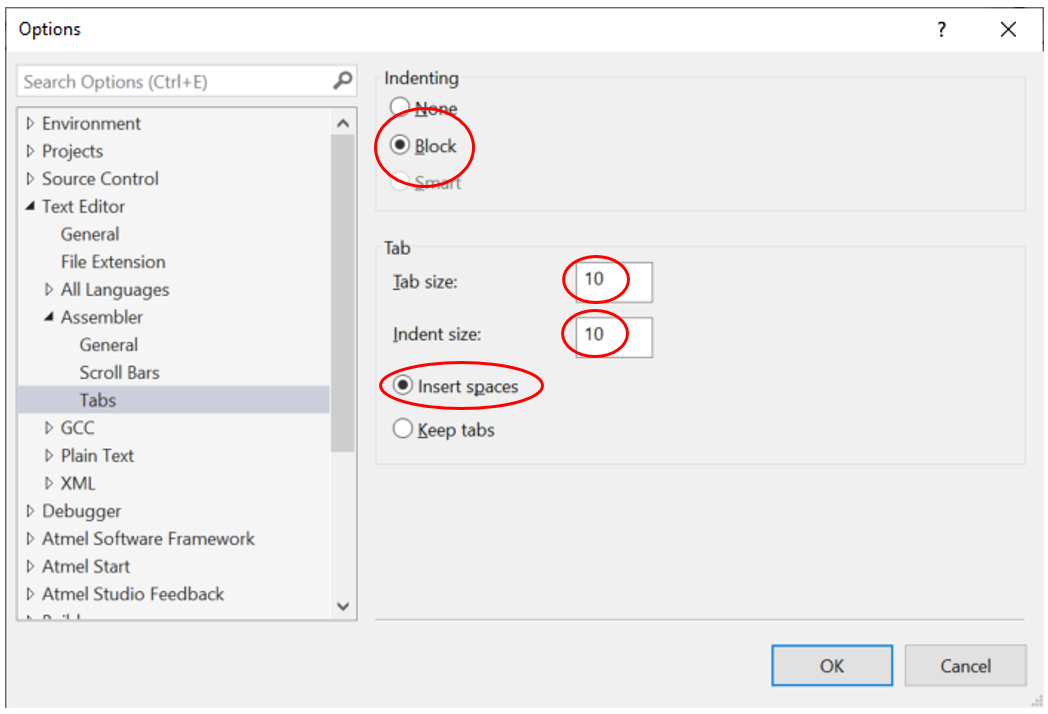
We will cover other options later.

* Save Project Settings



Change Tab Settings for Assembly Source Code

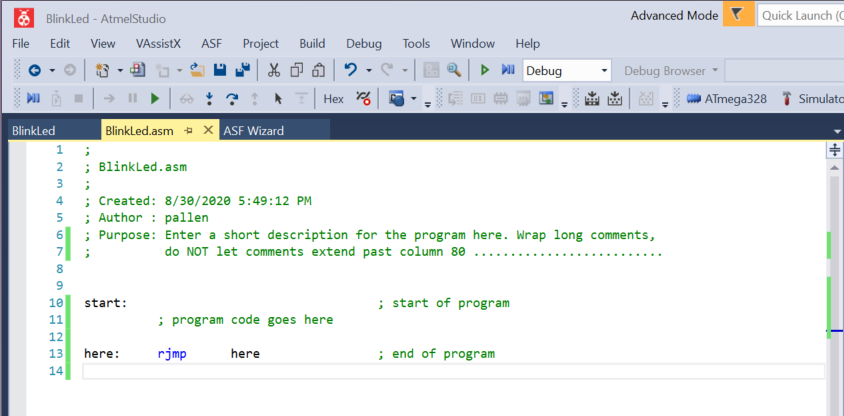
* Menu: Tools\Options
* Options: Text Editor\Assembler
  + General
    - Line numbers => checked
  + Tabs
    - Indenting => Block
    - Tab size => 10
    - Indent size => 10
    - Insert spaces => selected



* Click OK

Assembly Program Source Code Template

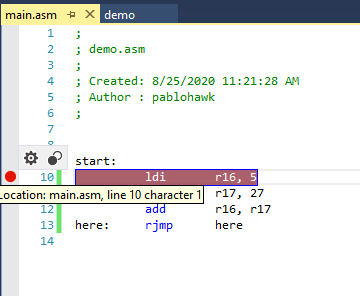
* Modify the default template source code as follows:
* Purpose:
  + Add a project description to the file header comment block
  + This MUST be edited for each program with a short description of the assignment or project
  + Wrap long comments and or lines of code so that no text extends past column 80



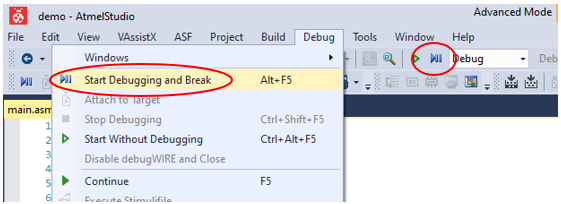
* Delete existing lines in default template and replace with code shown above
  + Remove the comment “program code goes here” when writing your program
  + Line 13 sets up and endless loop that will keep your program active until you end your debug session or remove power from your device
* Note columns set with tab-size=10
  + [label:] – column 0
  + [mnemonic] – column 11
  + [operands] – column 21
  + [; comment] – column 41

Debug a Program in Atmel Studio 7.0

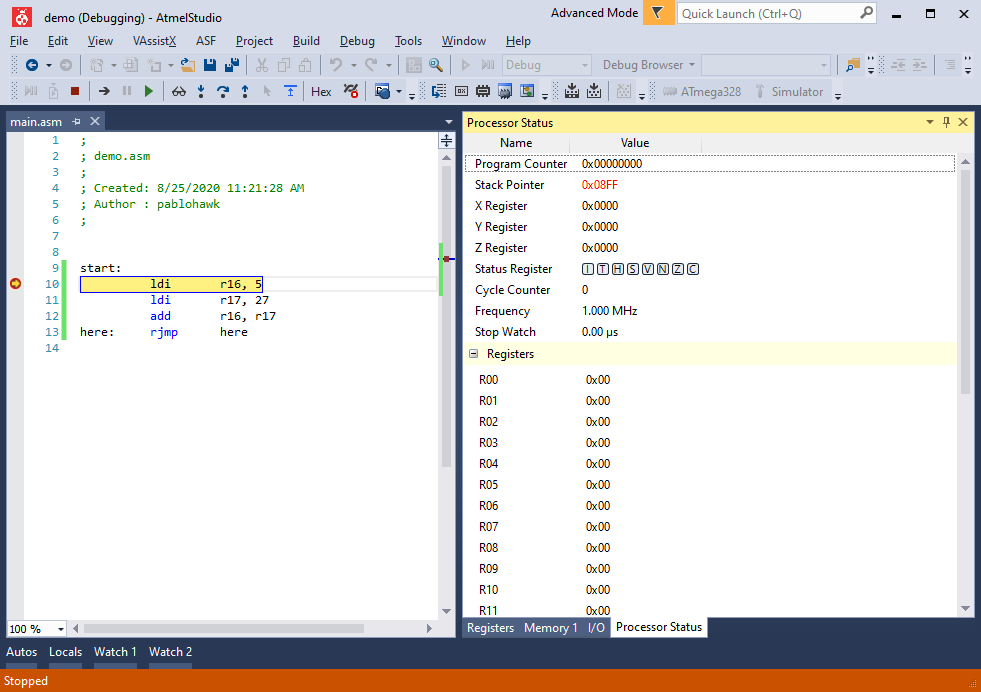
* Place a breakpoint on your first instruction
  + Place breakpoints by left-clicking in the grey-bar on the left side of the line-number



* + Optionally you can set a breakpoint by moving your cursor to the line you want and using Menu: Debug\Toggle Breakpoint (or F9)
* Start the debugger – either through Debug menu or toolbar



Debugger Running



Debug Windows – use Menu: Debug\Windows to select which ones are shown

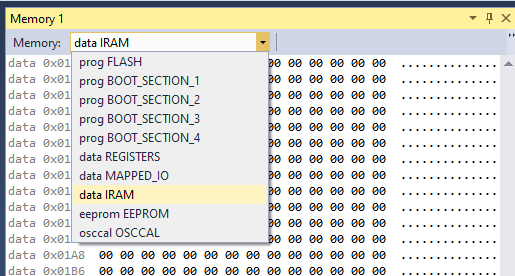
Step-over: Execute the currently highlighted instruction.

Yellow arrow and highlighted row indicate next instruction to be executed.

You must execute an instruction (step-over) to see the result of the instruction in the debug windows such as Processor Status, I/O, Memory, etc.

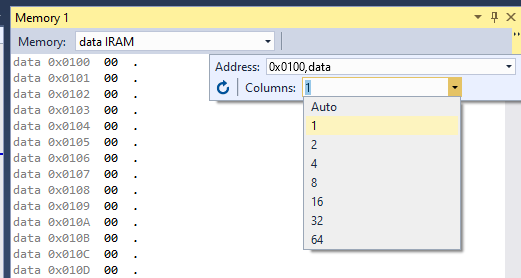
The main debug windows needed are:

* Processor Status – Shows GPR, Status Register (SREG) and other values
* I/O – expend each port to see bit values for pins for the port
* Memory – you may open multiple Memory windows
  + Set the type of memory you want to see, such as data IRAM (Internal SRAM)



* + You can adjust the number of columns of data shown in the memory window

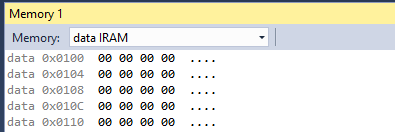
The memory window here is set to 1 column, i.e. one byte value associated with each memory address, with a starting address of 0x0100 (or the starting address of Internal SRAM)



The Memory type you set re-orients the memory window the starting address of that type of memory, i.e. GPR 0x0000, I/O 0x0020, etc.

You are seeing multiple addressable Bytes when Columns is anything greater than 1, i.e. the address numbers will increase by the number of columns shown.

e.g.: we are seeing 4 bytes of memory per row, so the first row represents bytes at addresses 0x0100, 0x0101, 0x0102, 0x0103. Values are shown in hex by default, and each 2 hex digits are 1-Byte.



* Press Step-Over (F9) to execute each instruction in your program
  + Note values that change in the different debug windows (they will typically be highlighted in red)
  + Your program will end and the debug will stop if you do not include an endless loop at the end of your program (all debug windows will become empty or unreadable)