Mechatronics Engineering

Lab 1

Introduction to Embedded Systems and Microchip Studio

GUC German University in Cairo

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Lab Teaching Assistants:

- MSc. Eng. Menna Mohsen
  Office: C3.123, Email: mennatullah.mohsen@guc.edu.eg
- MSc. Eng. Ragi Adel
  Office: C1.212, Email: ragi.adel@guc.edu.eg
- MSc. Eng. Youssef Fahmy
  Office: C1.203, Email: youssef.fahmi@guc.edu.eg
- MSc. Eng. Abdallah Hossam
  Office: C3.123, Email: abdallah.mohammed@edguc.edu.eg
- Eng. Arsany Youssef
  Office: C1.203, Email: arsani.youssef@guc.edu.eg
- MSc. Eng. Ahmed Fahmy
  Office: C1.212, Email: ahmed.hossam-aly@guc.edu.eg
- MSc. Eng. Abdallah Mohammed
  Office: C3.208, Email: abdallah.abdelrahman@guc.edu.eg

Course Policies

- **You have to attend the lab session only with your group.**
- Validation reports will be submitted **only during your lab session**.
- It is not allowed to negotiate your grades with the your TA. So, check the office hours if you need any support or help before the deadline.
- Taking attendance and entering the lab will not be allowed 10 mins after the starting of Labs.
- You will be informed with any other policies based on your attitude during this semester.

Tutorial Contents

- Introduction to Embedded Systems
- Introduction to Micro Controllers
    - ATmega328P
- Introduction to Microchip Studio
- Lab 1 Validation

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Tutorial Contents

- **Introduction to Embedded Systems**
- Introduction to Micro Controllers
  - ATmega328P
- Introduction to Microchip Studio
- Lab 1 Validation

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Introduction to Embedded Systems

- An embedded system is a computer system that has a dedicated function within a larger mechanical or electrical system. It is a combination of a computer processor, computer memory and input-output peripheral devices.

- An embedded system interacts with the environment and typically controls the physical operations of the machine that it is embedded within; thus, it often has real-time computing constraints.

- Some design issues of real-time embedded systems:
  - Selection of hardware and software.
  - Understanding the programming languages used and their real-time implications resulting from their translation into machine code.
  - Maximizing fault tolerance and reliability of performance.

Introduction to Embedded Systems

- Hardware usually used in embedded systems:
  - Microcontrollers (MCU)
  - Microprocessor (MPU)
  - Field Programmable Gate Array (FPGA)
  - Application Specific Integrated Circuit (ASIC)

- Difference between a microcontroller and a microprocessor:
  - Microprocessor is an Integrated Circuit (IC) which has only the CPU inside them. It doesn't have RAM, ROM, and other peripheral on the chip.
  - Microcontroller has a CPU, in addition with a fixed amount of RAM, ROM and other peripherals all embedded on a single chip. It is like a mini computer on a single chip.

- Modern embedded systems are often based on microcontrollers.
- Microcontrollers examples: PIC, AVR, ARM, … etc.
- We will use AVR microcontrollers in this course.

Tutorial Contents

- Introduction to Embedded Systems
- **Introduction to Micro Controllers**
  - **ATmega328P**
- Introduction to Microchip Studio

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

ATmega328P (AVR Microcontroller)

- From the datasheet: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

- High performance, low power AVR® 8-bit microcontroller
- Advanced RISC architecture
  - 131 powerful instructions – most single clock cycle execution
  - 32 × 8 general purpose working registers
- Peripheral features
  - Two 8-bit Timer/Counters with separate prescaler and compare mode
  - One 16-bit Timer/Counter with separate prescaler, compare mode, and capture mode
  - Real time counter with separate oscillator
  - Six PWM channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    - Temperature measurement
  - Programmable serial USART
  - Master/slave SPI serial interface
  - Byte-oriented 2-wire serial interface (Phillips I^2C compatible)
  - Programmable watchdog timer with separate on-chip oscillator
  - On-chip analog comparator
  - Interrupt and wake-up on pin change
- I/O and packages
  - 23 programmable I/O lines
- Operating voltage:
  - 2.7V to 5.5V for ATmega328P

ATmega328P can be programmed using:

- Assembly Language
- Embedded C Language
- Arduino Framework

Closer to machine level

Higher level and easier to program

GUC German University in Cairo

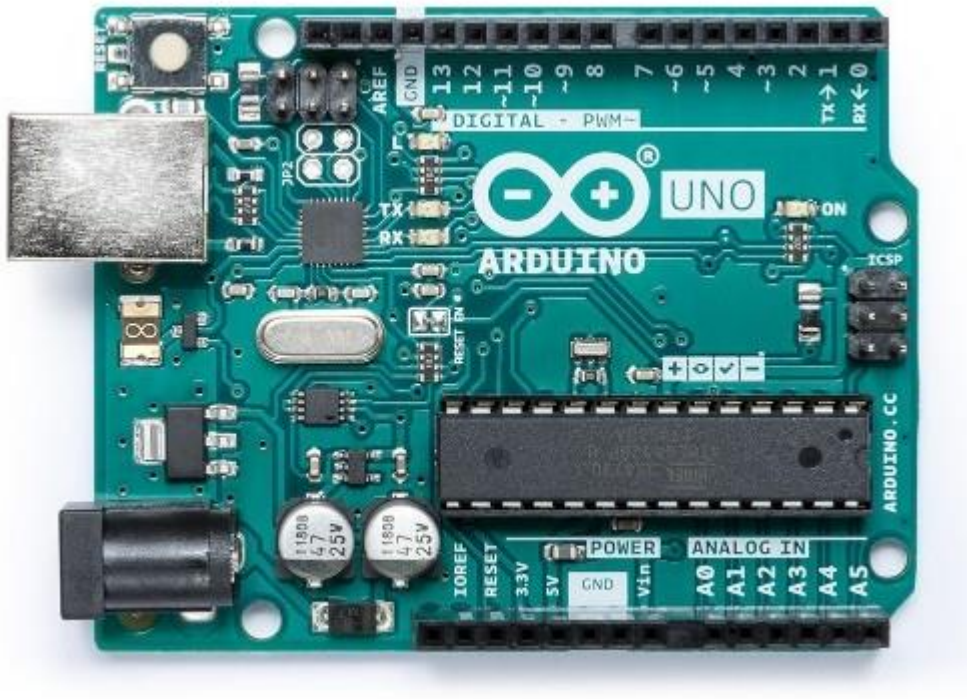
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Example of using Atmega328P as a Microcontroller (Arduino)

- Arduino is a microcontroller board based on the ATmega328P.

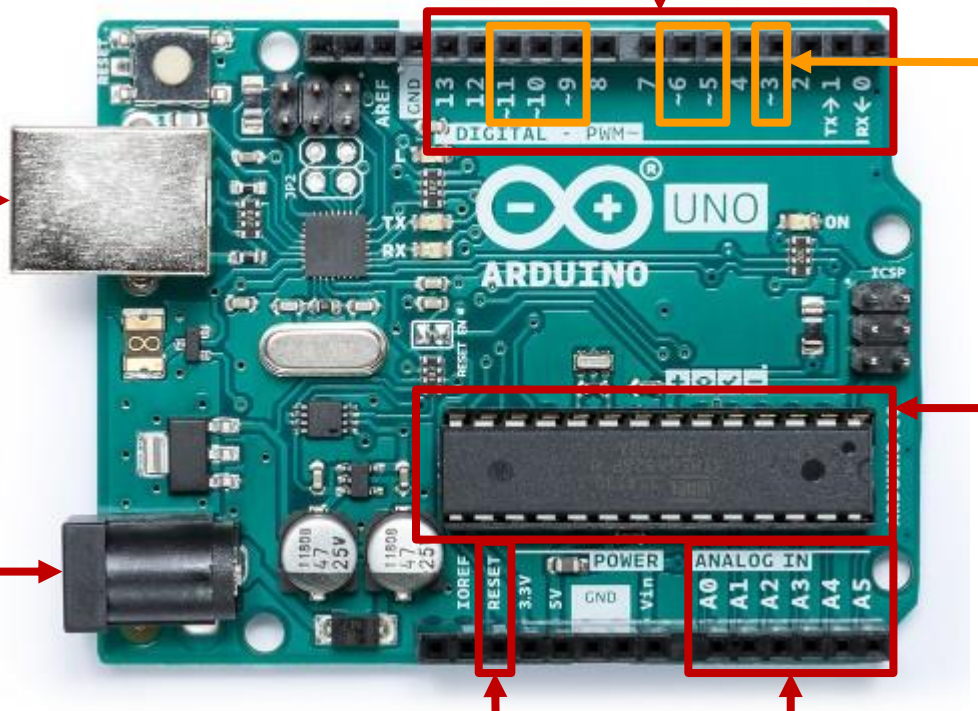
- It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

- It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

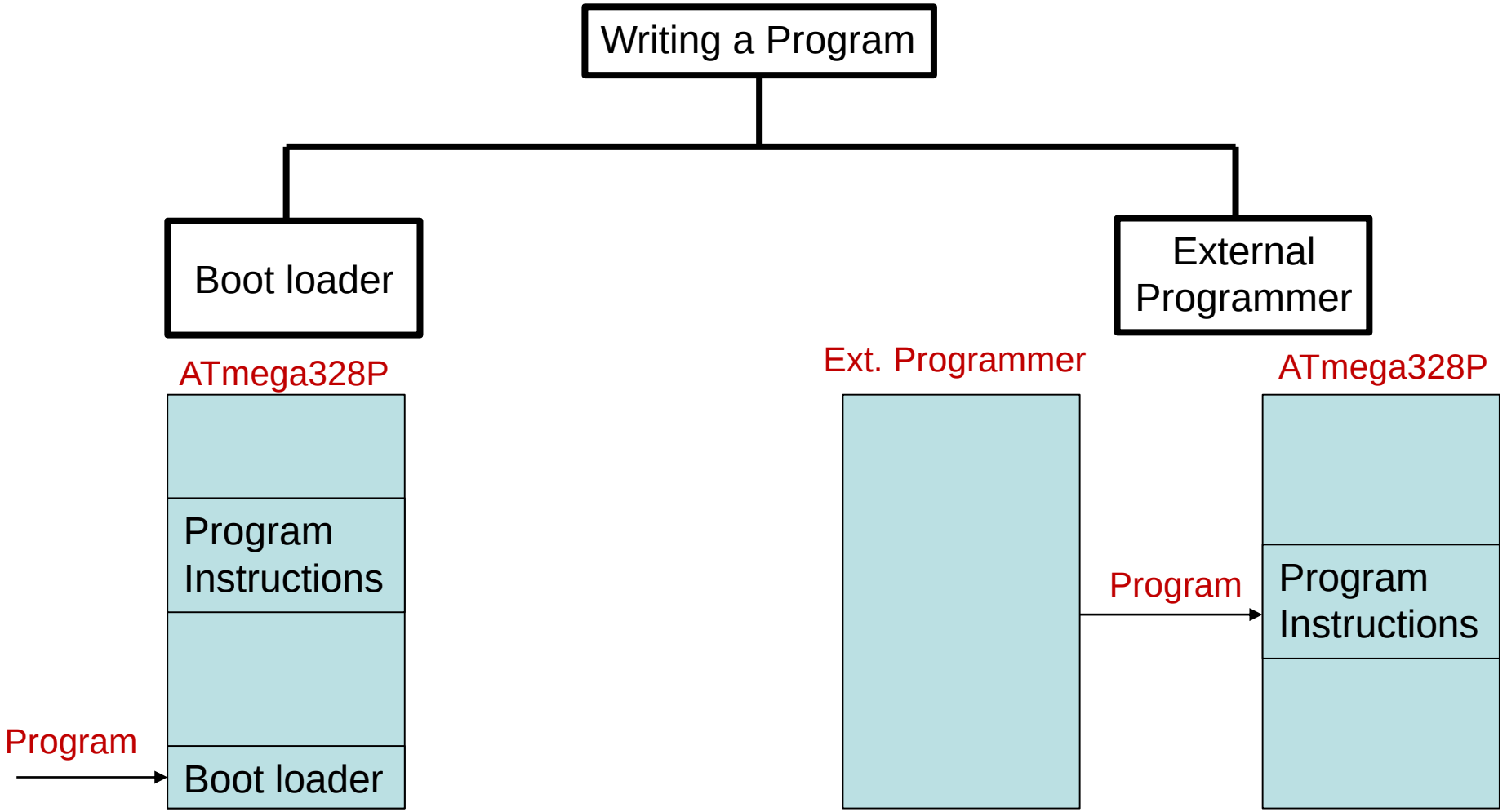


GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

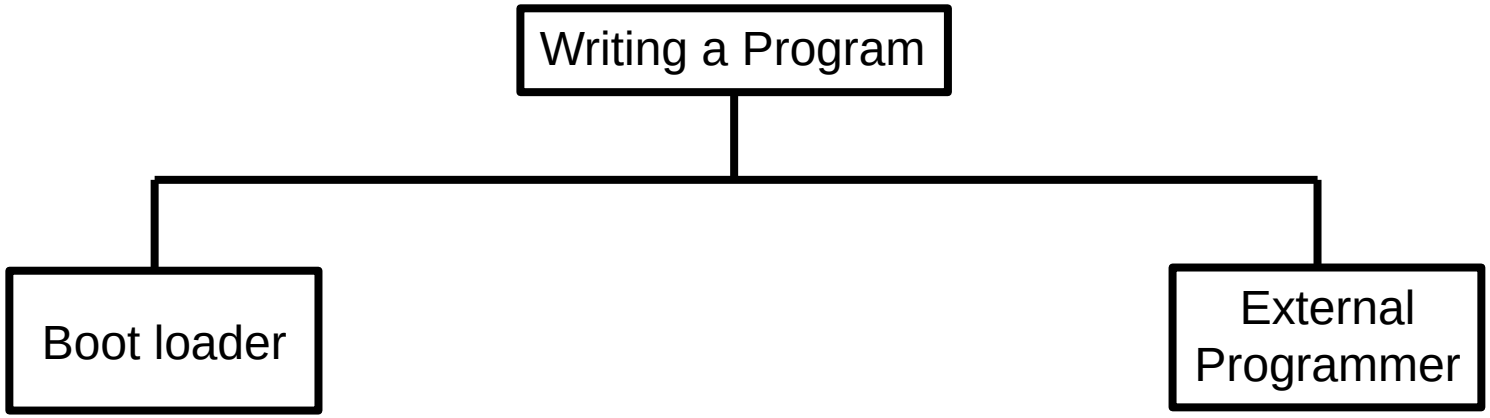
Arduino Uno



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science
German University in Cairo

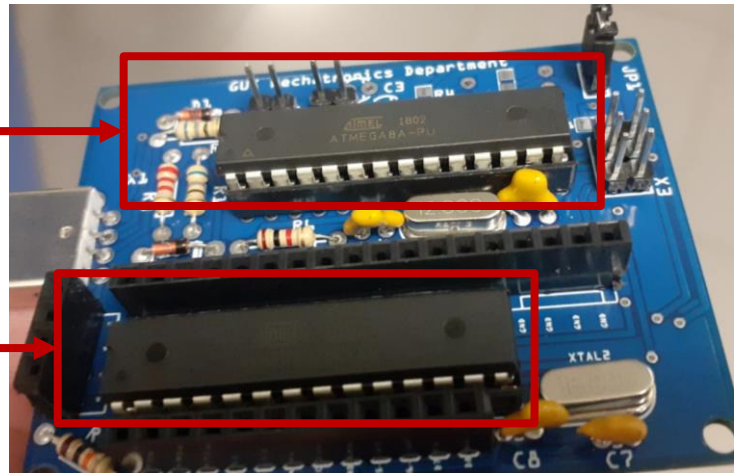


Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science



ATmega8A
(Ext. Programmer)

ATmega328P
microcontroller



Tutorial Contents

- Introduction to Embedded Systems
- Introduction to Micro Controllers
  - ATmega328P
- **Introduction to Microchip Studio**

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

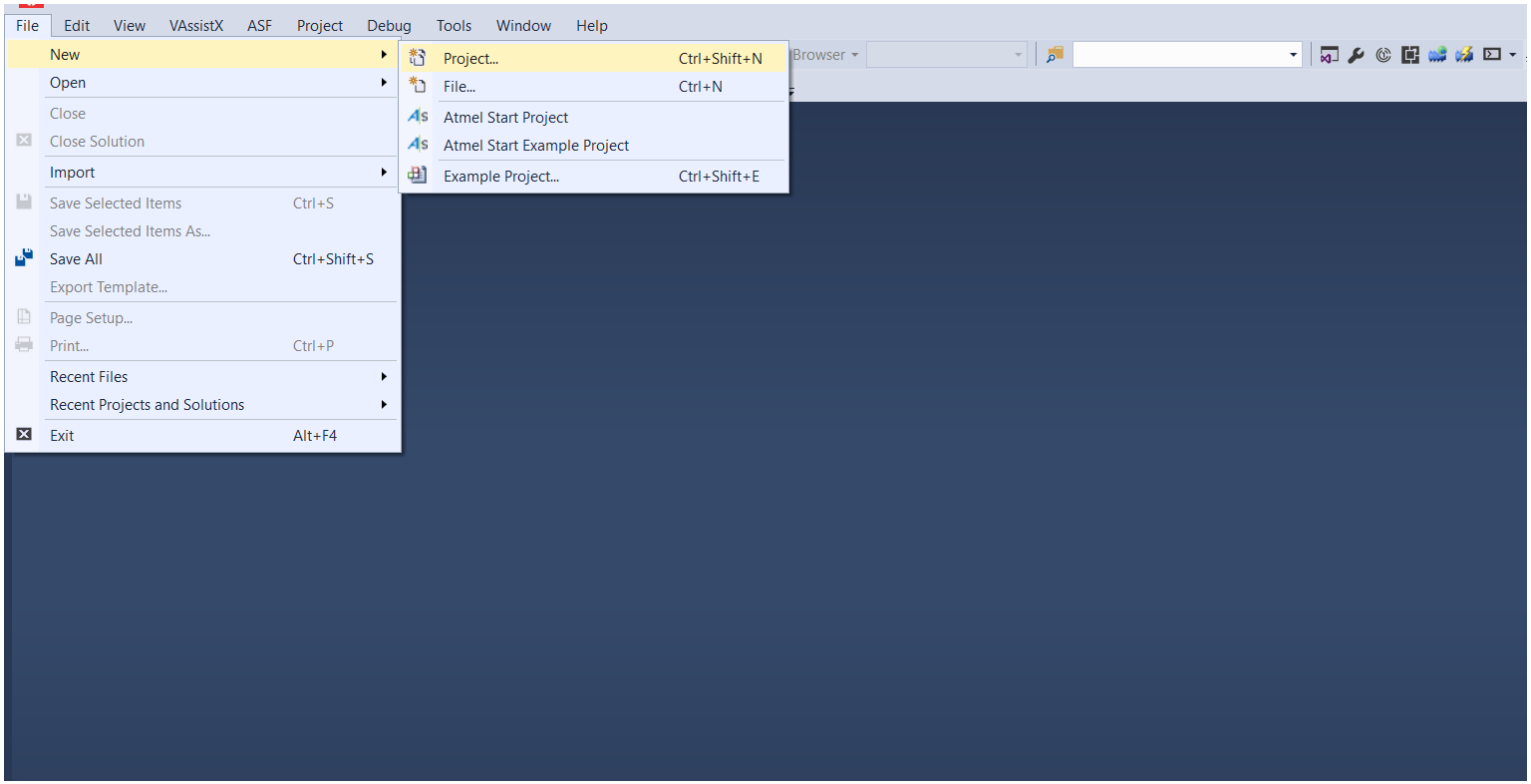
Microchip Studio Installation

- Microchip studio is the integrated development platform (IDP) for developing and debugging all AVR® and SAM microcontroller applications.

- Microchip Studio Download Link:
https://www.microchip.com/en-us/development-tools-tools-and-software/microchip-studio-for-avr-and-sam-devices

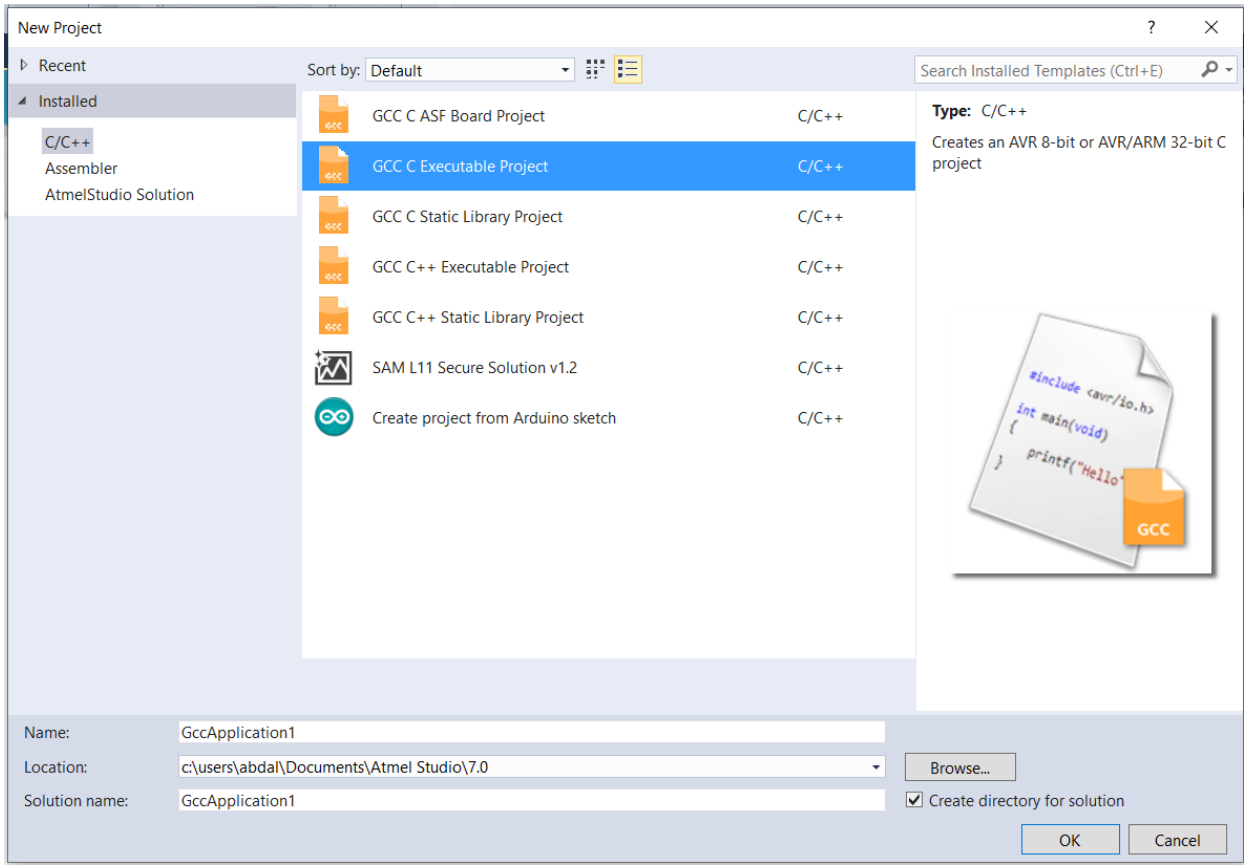
Creating a New Project in Microchip Studio

- To create a new project, click on **File>New>Project**



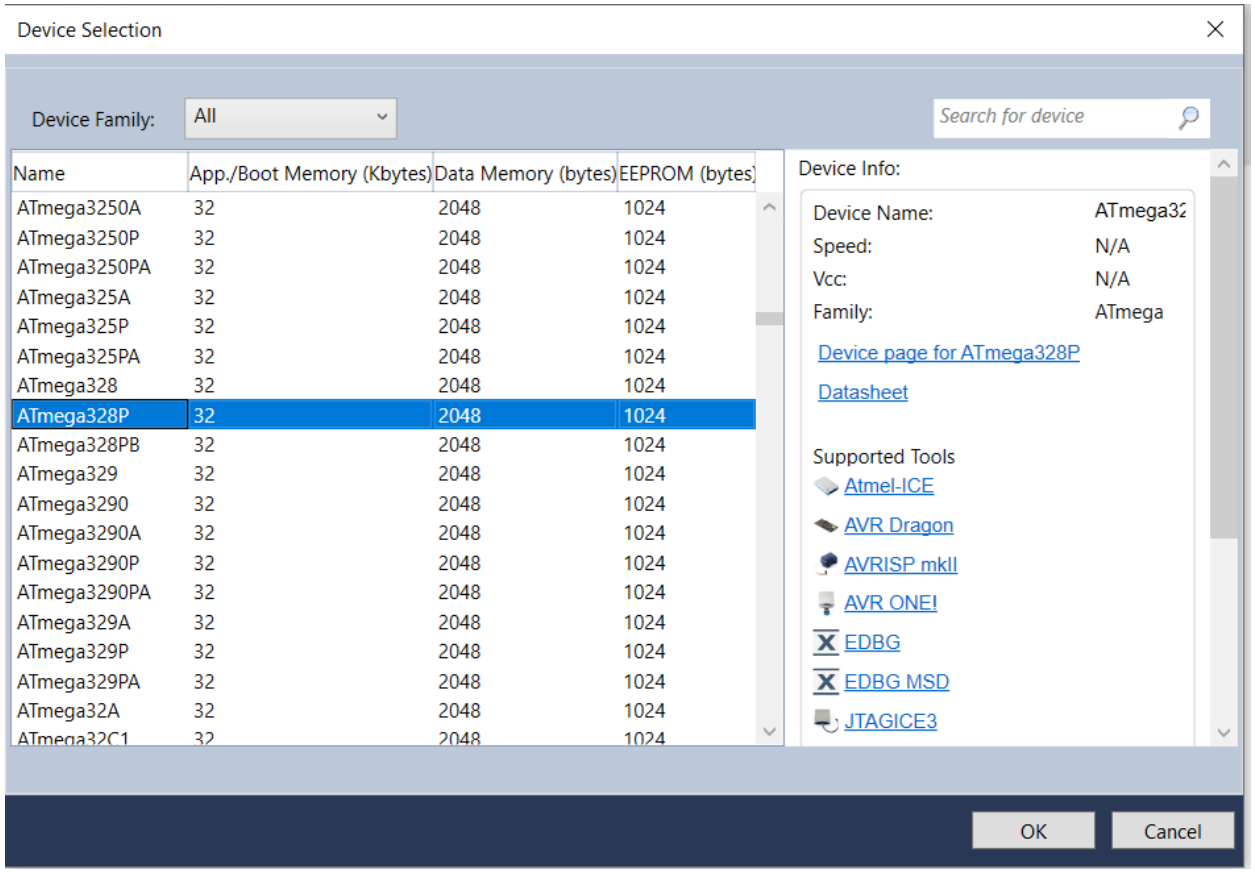
Creating a New Project in Atmel Studio 7

- Choose **GCC C Executable Project**



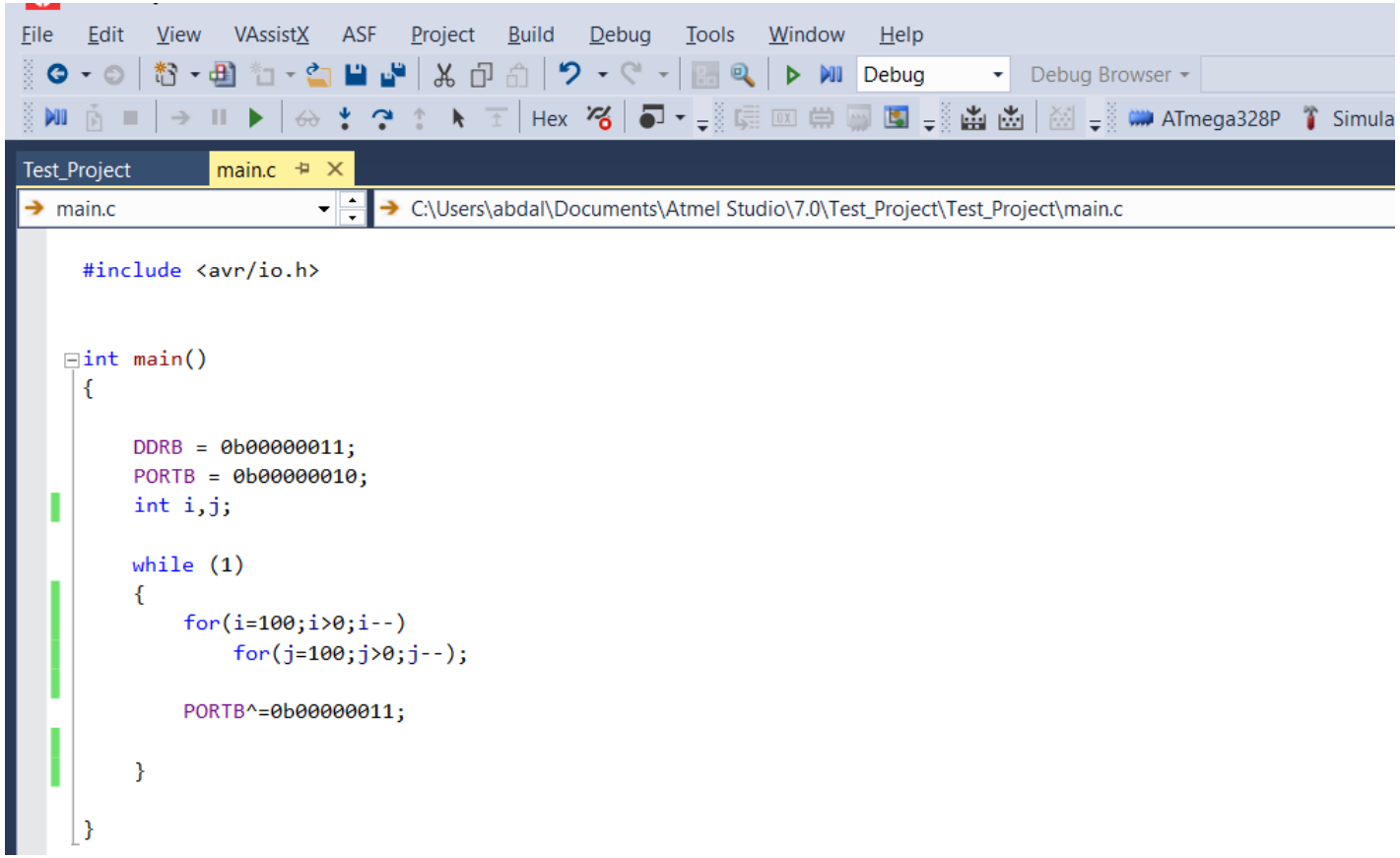
Creating a New Project in Atmel Studio 7

- Select **ATmega328P** from the list of devices:



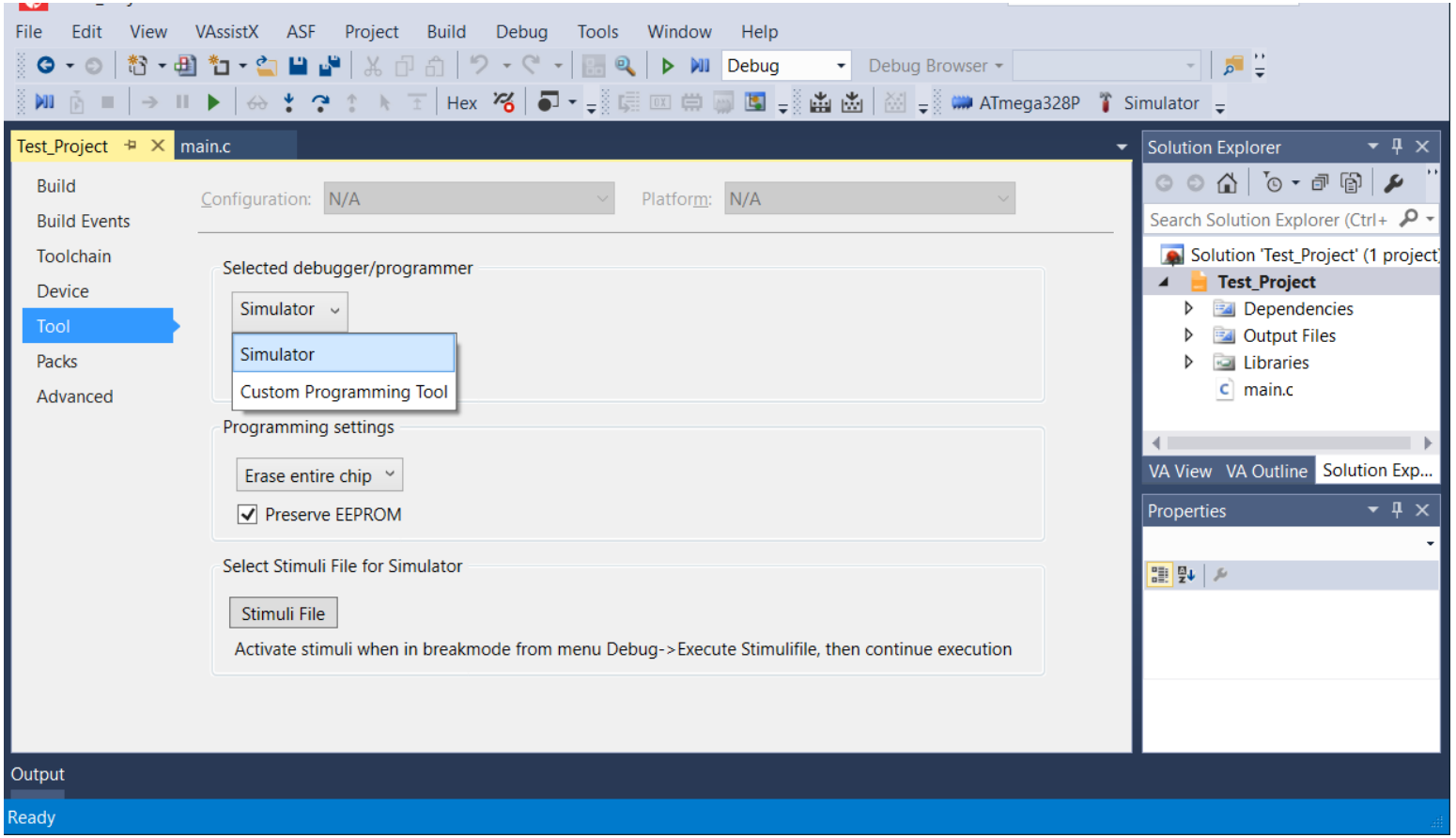
Build Project

- Write your code in main.c



Select Debug Tool

- Click on **Project > [Project name] Properties**
- From **Tool** tab , Select **debugger/programmer** to be **Simulator**

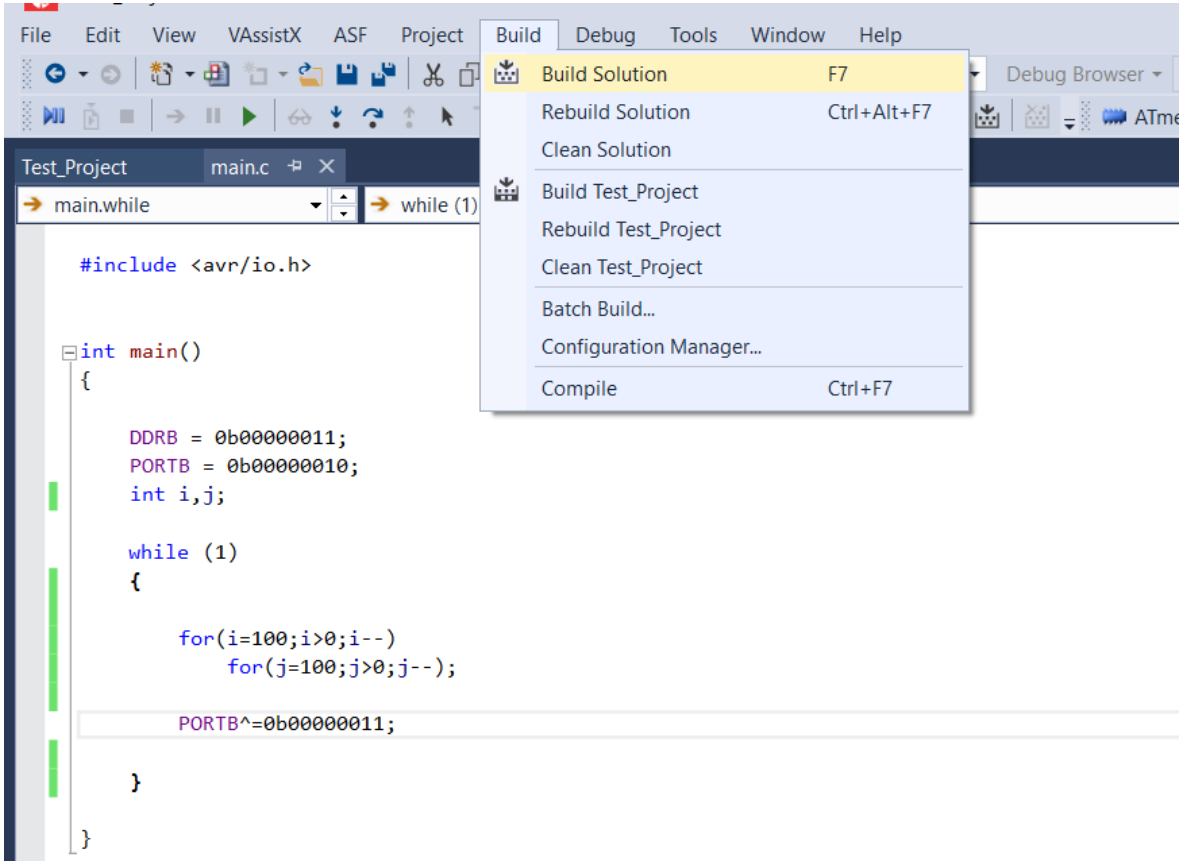


GUC German University in Cairo

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

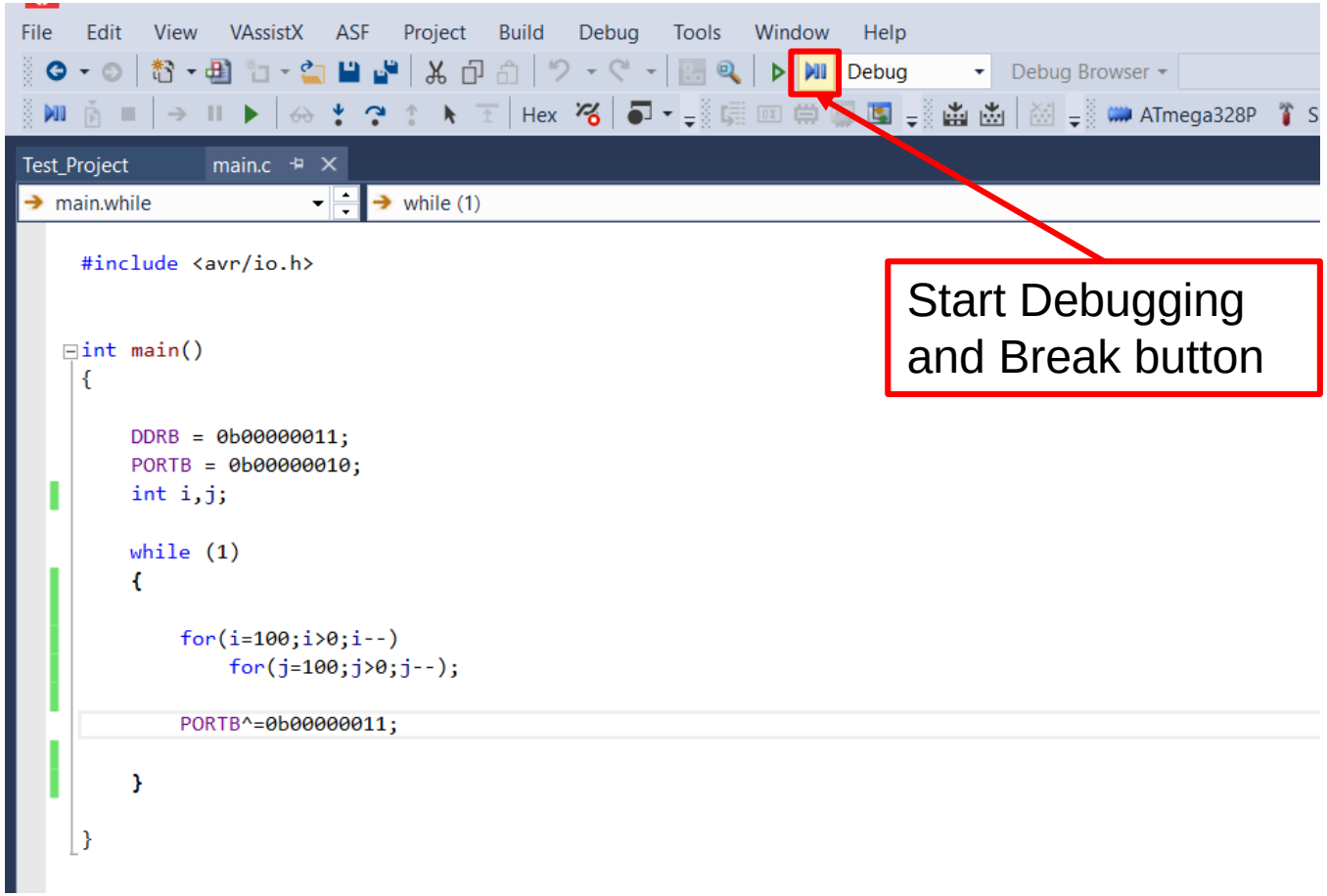
Build Project

- Click on **Build> Build Solution**



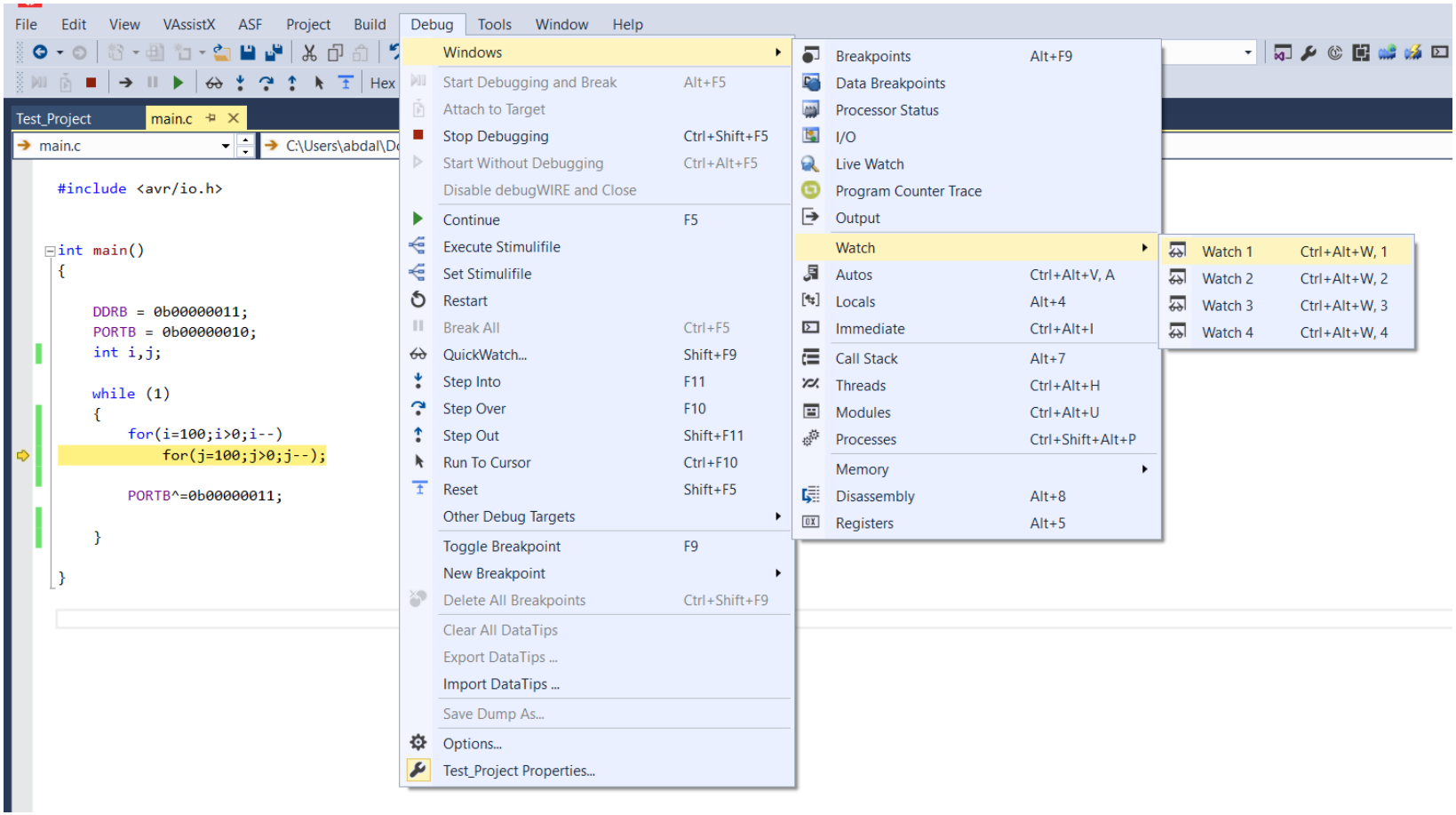
Debug Project

- Click on **Start Debugging and Break** button



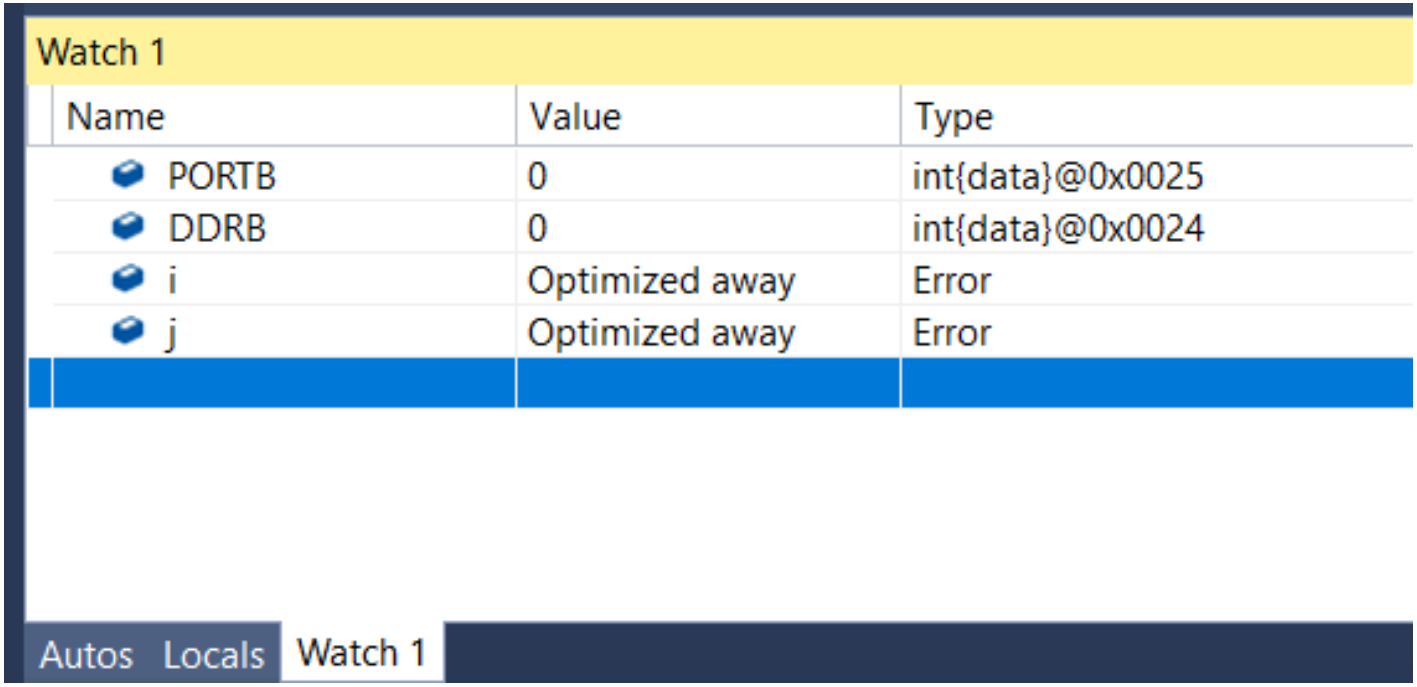
Debug Project

- To watch how variables change during the debugging session, click on **Debug>Windows>Watch>Watch 1**



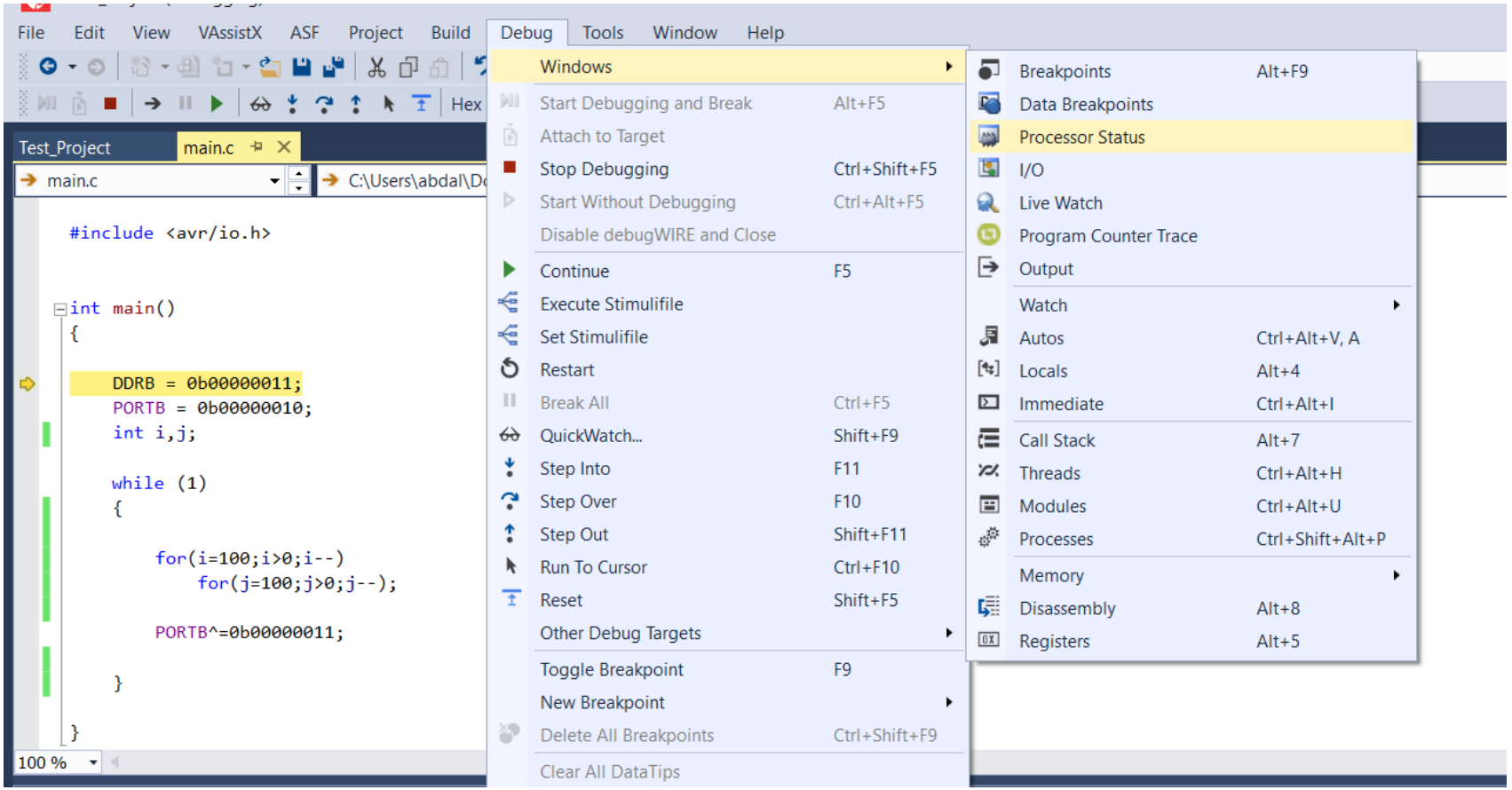
Debug Project

- In the watch window, type the name of the variable you want to watch during debugging then press enter to add it to the list.
- In this example the variables of interest are: **PORTB**, **DDRB**, **i** and **j**



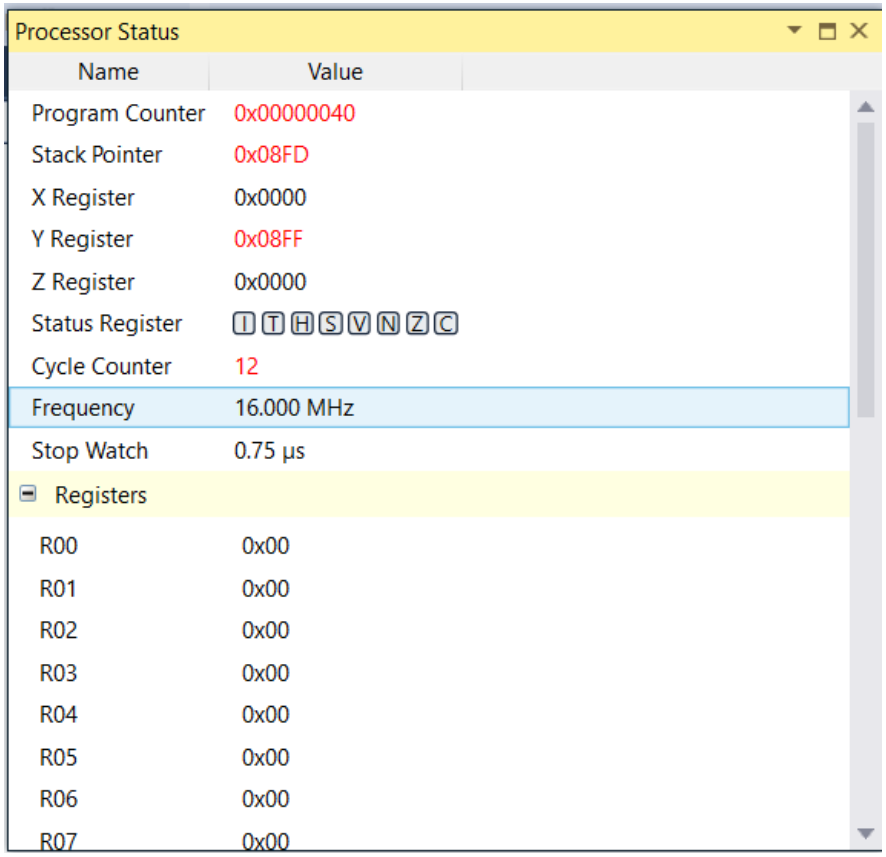
Debug Project

- Click on **Debug>Windows>Processor Status**



Debug Project

- Set processor clock frequency. In this case it is 16 MHz
- The **Cycle Counter** is the number of instruction cycles elapsed since the start of the simulation and the **Stop Watch** shows the time elapsed since the start of the simulation.

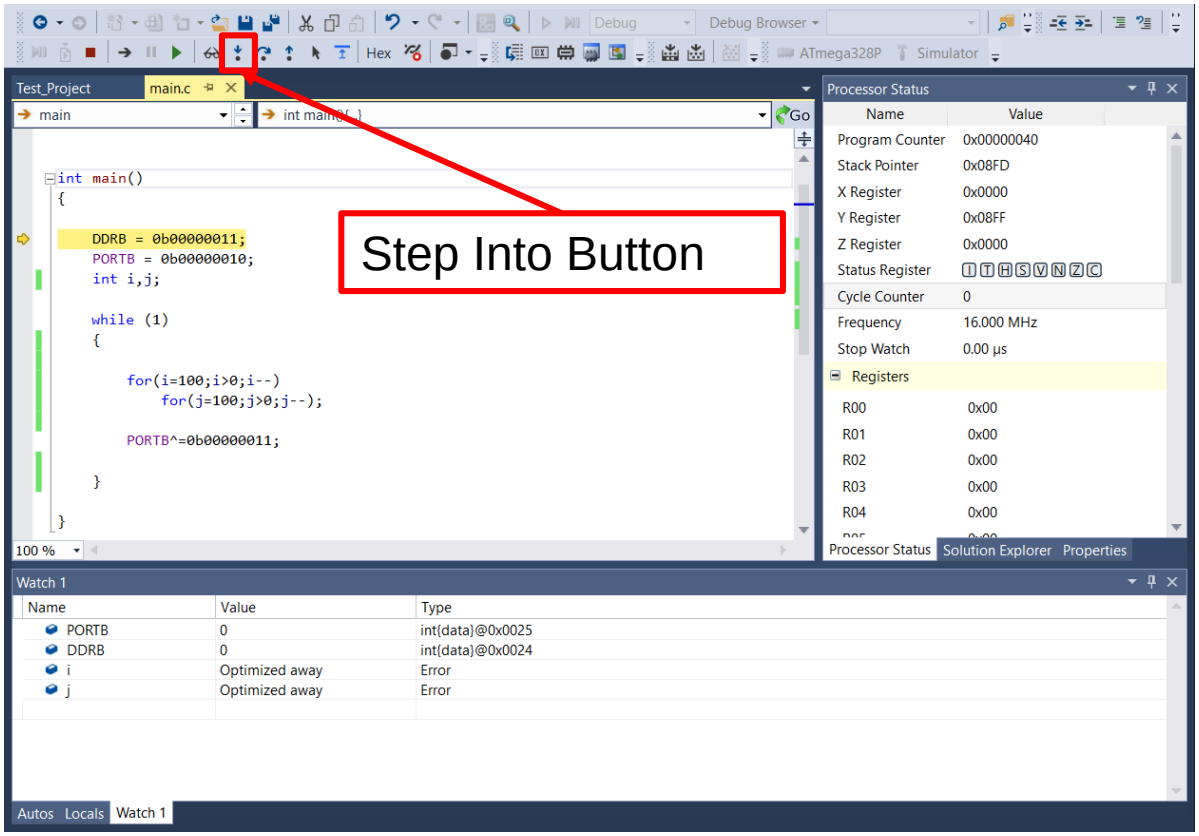


GUC German University in Cairo

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Debug Project

- Click on **Step Into** button to execute a single step in the simulation
- Watch how variables change as the program executes
- To execute the code until it reaches a certain line right click on the line and click on **Run to Cursor**



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

To be prepared for the next week lab :

- Purchase the components shown in the components list pdf on the CMS
- Solder the components to build the AVR embedded
- Download the following softwares:
    - Zadig: from https://zadig.akeo.ie/, tested version is v2.5
    - Avrdude: from http://download.savannah.gnu.org/releases/avrdude/. Tested version is v6.3
- Bring your laptop !!!