

**Namal University**

**Mianwali**

**Department of Electrical Engineering**

EE-252L: Introduction to Embedded Systems

**Lab Manual: 01**

**Introduction to Microchip Studio and Assembly Language**

|  |  |
| --- | --- |
| **Students Name** | **Fahim Ur Rehman Shah** |
| **Roll Number** | **NIM-BSEE-2021-24** |
| **Submission Date** | **28/03/2023** |
| **Marks Obtained** |  |

**Instructors: Engr. Faizan e Mustafa / Dr. Hamza Zad Gul**

**Introduction**

An embedded system is a combination of computer hardware and software designed for a specific function. Embedded systems may also function within a larger system. The systems can be programmable or have a fixed functionality. Industrial machines, consumer electronics, agricultural and processing industry devices, automobiles, medical equipment, cameras, digital watches, household appliances, airplanes, vending machines and toys, as well as mobile devices, are possible locations for an embedded system.

**What are Embedded Systems?**

An embedded system is a small or large non-computer device with integrated software based on microcontrollers and microprocessors for performing a dedicated function or a limited set of functions. It may or may not have a screen and a keyboard, be either programmable or non-programmable, perform a single function in isolation, or work as a part of a large system.

A TV remote control, a microwave oven, a network of sensors and control systems in automobiles and complicated manufacturing robotic equipment — all these devices and electronic systems operate due to embedded software.

Diagram, schematic

Description automatically generated

Figure 1.1: Block diagram of an embedded system

Microcontroller Processor RAM ROM ADC I/O Ports Bus DAC Analog Signals Chemical Electrical, Mechanical, or Other devices embedded programming skills. We follow the bottom up approach by starting with simpler tasks and gradually building on that to develop a complete embedded system.

**Lab Objectives**

This lab provides an introduction aimed at teaching basic usage of simulator and its functionality

**Prerequisites for Lab**

This lab is designed for the students having some experience in programming, but no prior experience with embedded systems. In this lab, we assume that you understand the basics of digital logic design and analog electronics.

**Hardware Required**

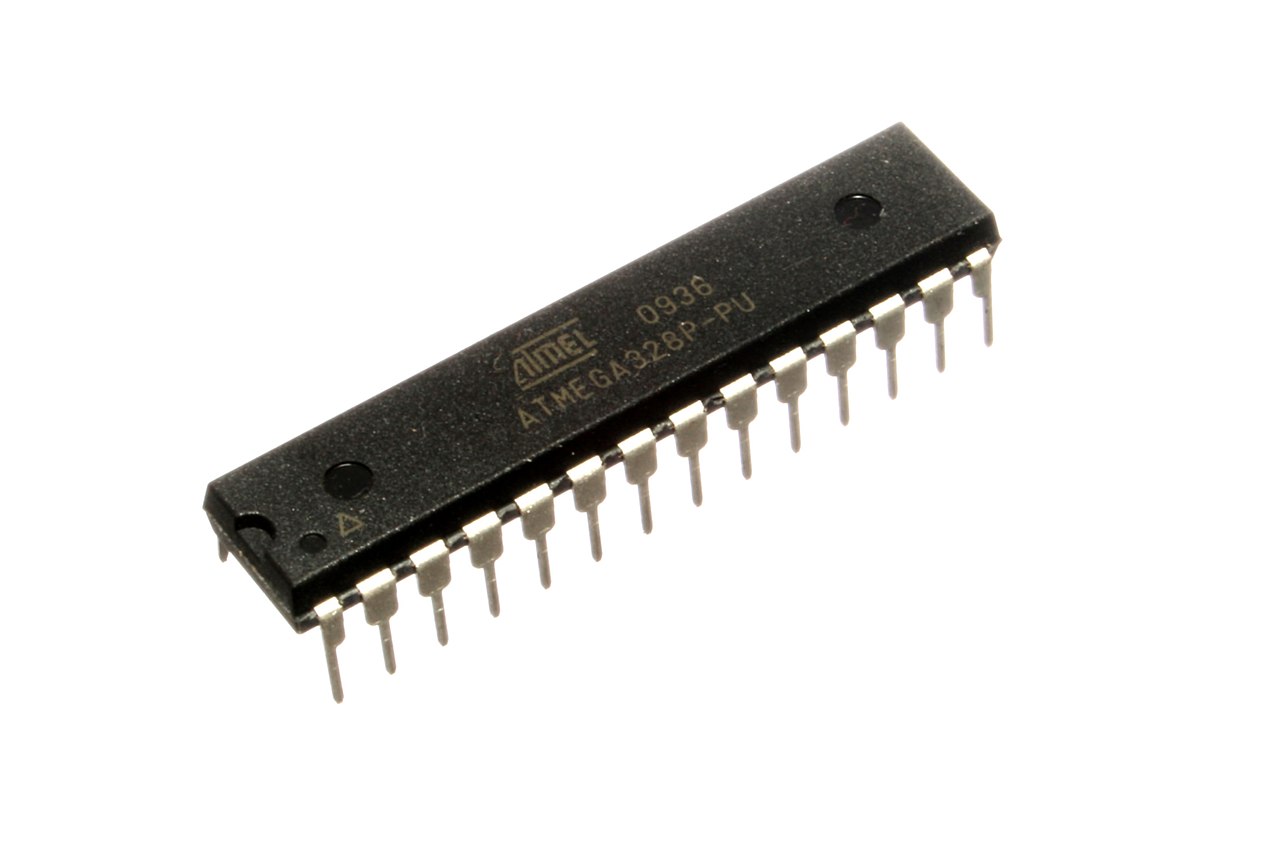
This lab does not require any hardware components

**AVR ATMega328P**

The high-performance Microchip picoPower® 8-bit AVR® RISC-based microcontroller combines 32 KB ISP Flash memory with read-while-write capabilities, 1024B EEPROM, 2 KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented Two-Wire serial interface, SPI serial port, a 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts.

**AVR Development Tool**

To develop an application and run it on AVR microcontroller, a software is required to write our code, debug and download it to the device. Fortunately, many IDEs are available for the application development of AVR. In this lab, we will use AVR Studio by Microchip as our development tool.



*Figure 1.2: ATMega328P*

**Downloading and Installing AVR Studio**

Follow the step below to download and Install AVR Studio:

1. Go to website <https://www.microchip.com/en-us/development-tools-tools-and-software/microchip-studio-for-avr-and-sam-devices#Downloads>
2. Select Product download option.
3. Select Web Installer or Offline Installer (as per own preference).
4. Download installer file.
5. Install the AVR studio using the installer file.

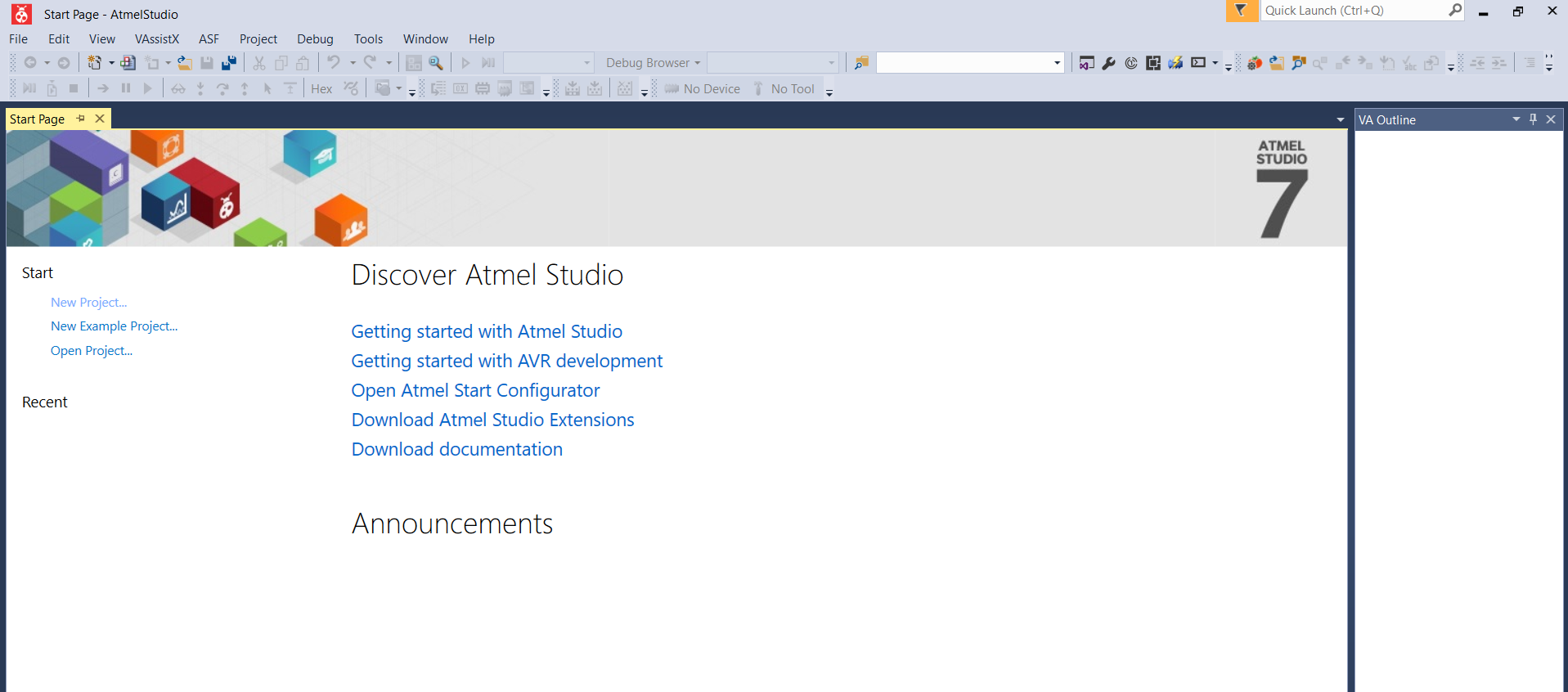
**NOTE:**

Web Installer: Requires internet connection during download and install process

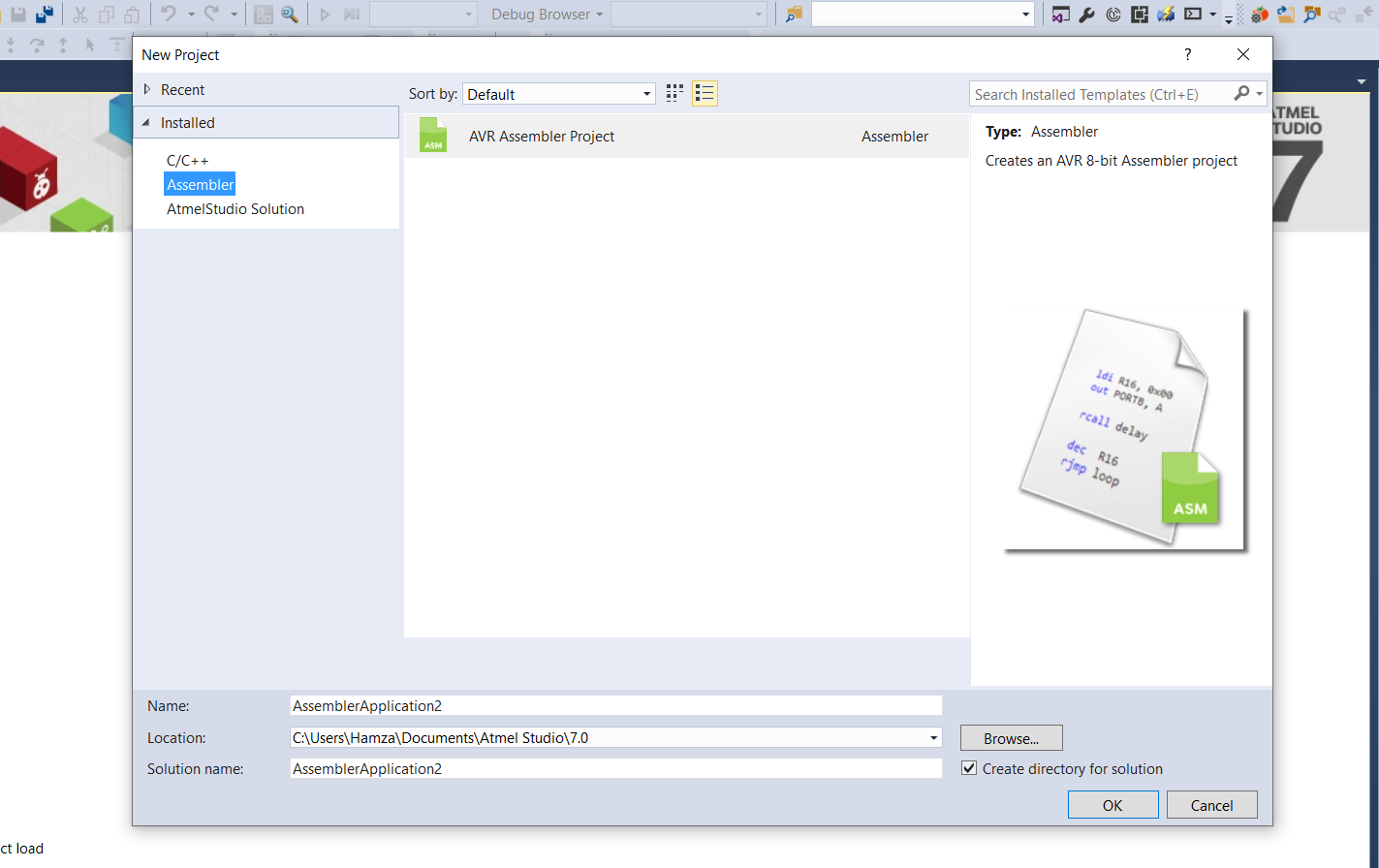
Offline Installer: Requires internet connection only during the download and not during the install process

**Setup Atmel Studio to Write Code**

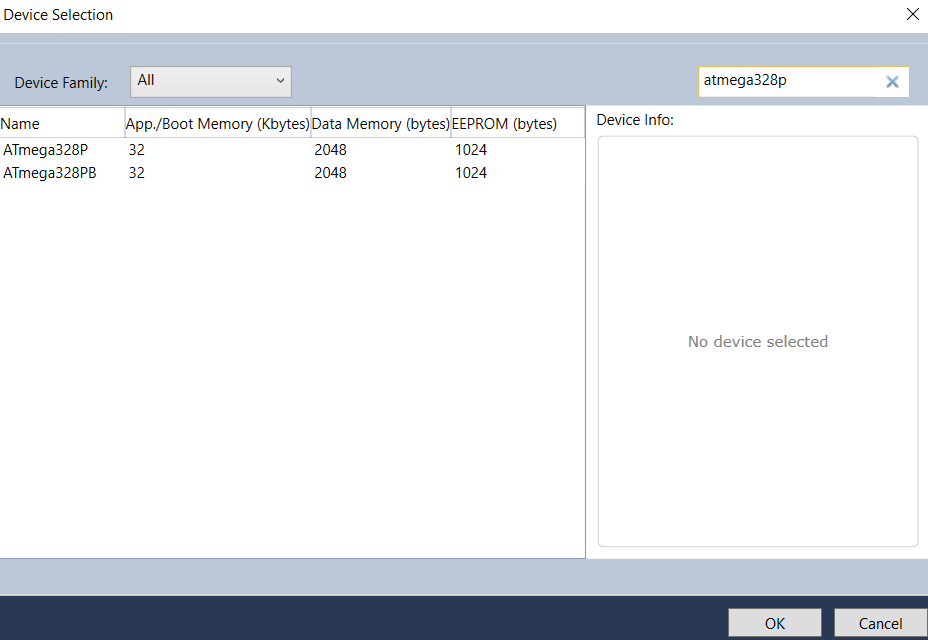
1. Run the software by clicking the icon on desktop, if available, or by clicking on Start → All Programs → **Microchip (or Atmel Studio)**. An interface similar to one shown below will open.



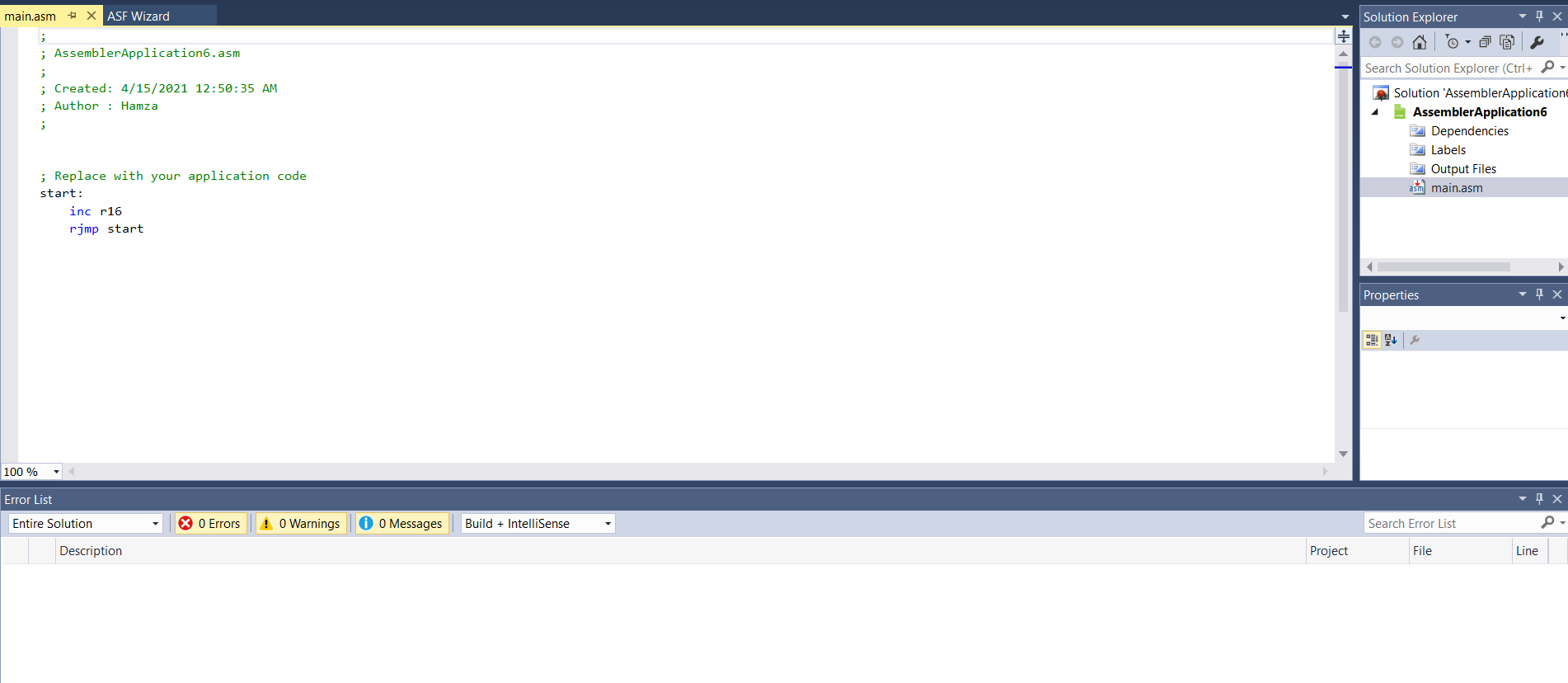
1. Click on **New Project** on the left hand side and choose **Assembler** followed by **AVR Assembler Project** from the menu as shown in below



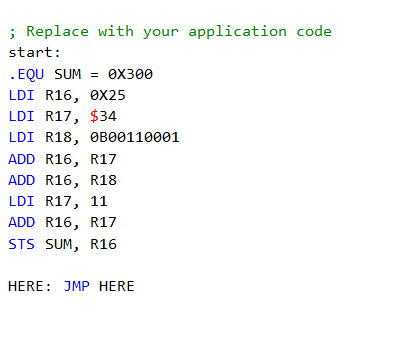
1. Select your chip, use the search area and type **ATMega328P.** Select the ATmega328P device and click on **OK**



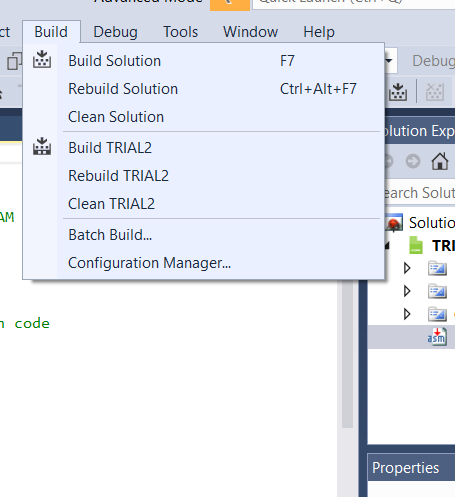
1. Your project window will open



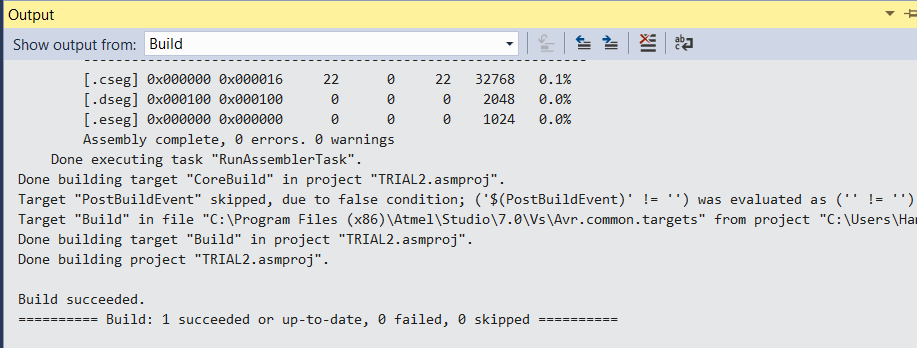
1. Remove the lines below “**Start:**” and type your code there (type the following code)



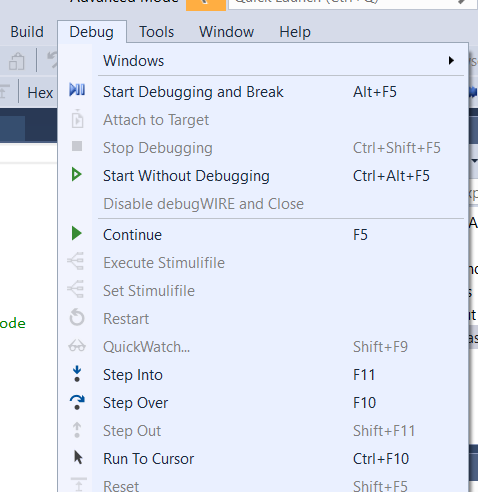
1. After writing the code, Open the **Build** menu and select **Build “Filename”**



After build is complete, check in error list to make sure there are no errors in your code



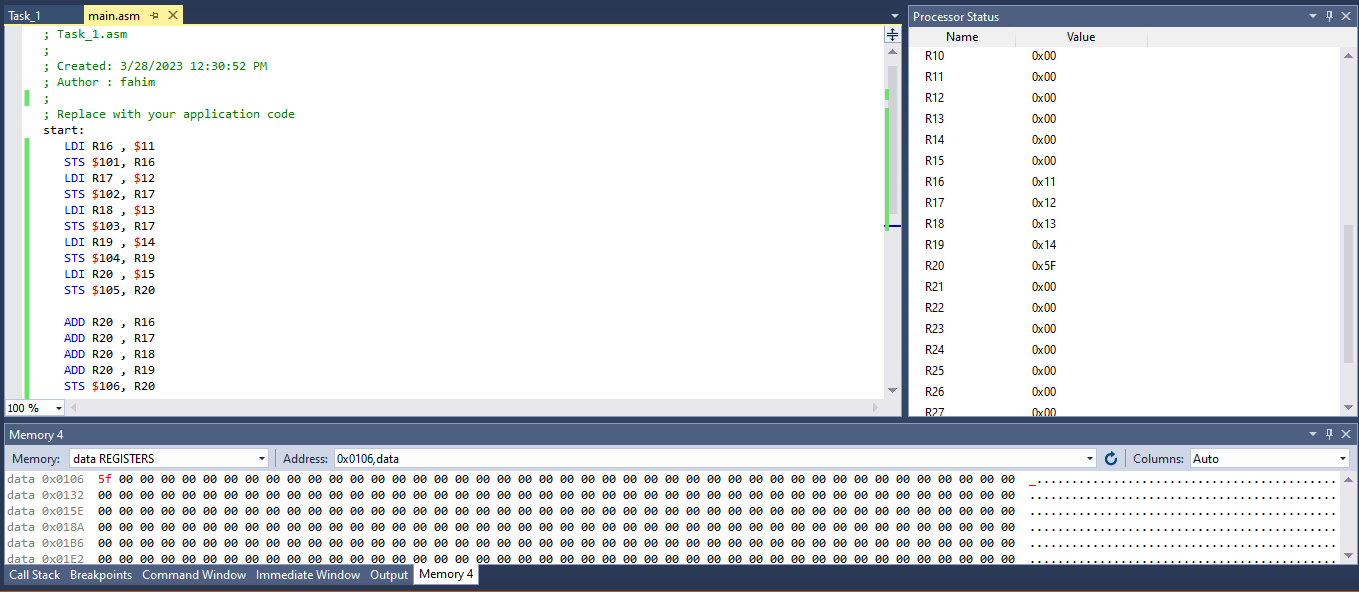
1. After this go back to **Build** menu and select **Rebuild “Filename”**. Following this set go to the **Debug** Menu and select **Start Debugging and Break**



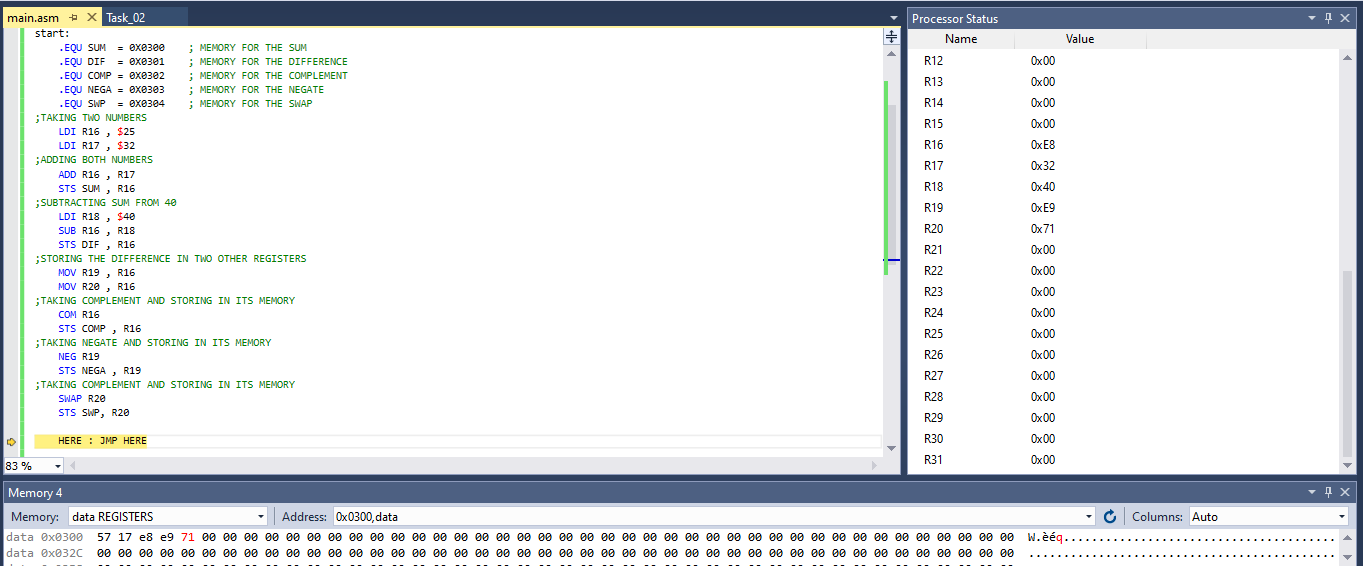
1. After starting Debugging mode is activated, execute your program one step at a time and view the contents and changes in the registers and memory.

**Lab Tasks:**

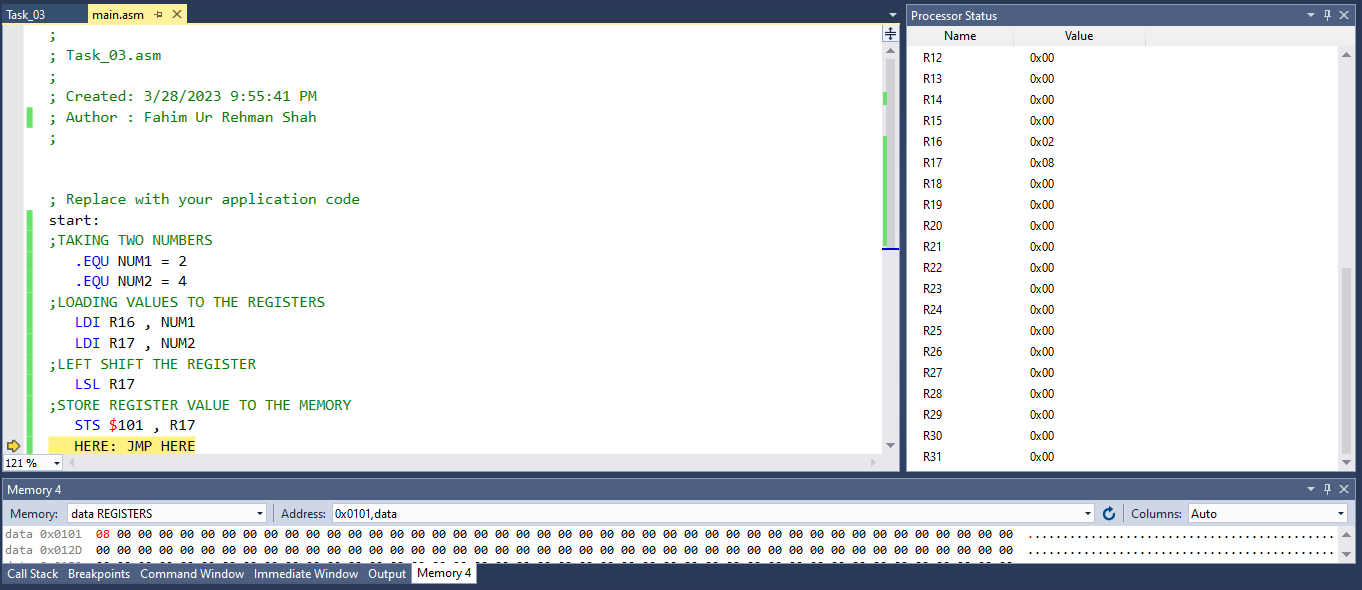
1. Write a simple code to (a) load the values $11, $12, $13, $14 and $15 into locations $100 to $105 and (b) add the values together and place the result in $106 after addition is done.

T**ask 1 . Code and Out Put**

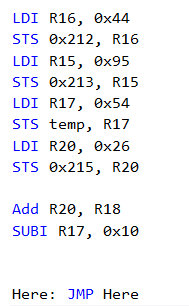
1. Write a simple code in which you will add any two numbers then subtract from 40. The result should be complement, negate and swap and store the results of this in separate memory locations of your choosing. (You can consider any two numbers of your choosing)

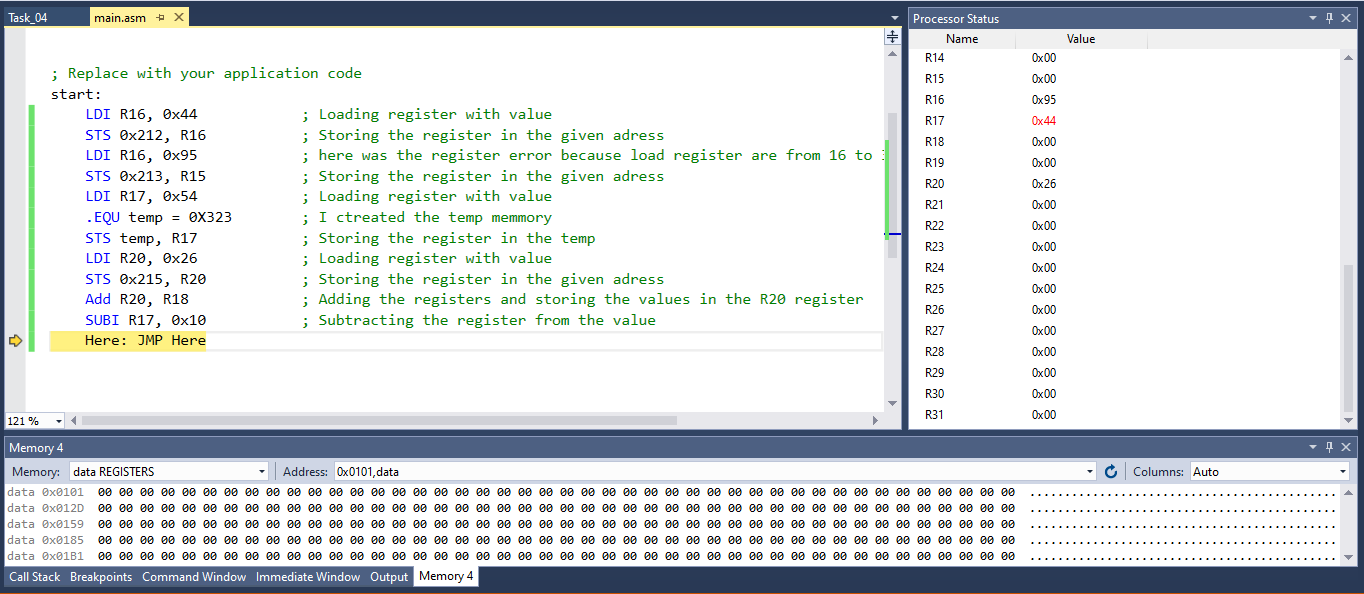
**Task 2 . Code and Out Put**

1. Write a code to multiply any two numbers. (You can choose any two numbers)



1. Write the following program in your simulator and run the code. If you get any errors then fix the code and rerun it. Provide comments in front of each instruction which tell what the instruction does





**Conclusion:**

In this lab we learned about different commands of assembly language for ATMEGA328P multicontroller processor. We performed different task for building the basics logics for the controller which can be used for controller special tasks operations . We have learned that hoe to use differenty commands like LDI , ADD , SUB and STS etc , we stored some values in the registers and perform different operations on it. We performed tasks such as adding and multiplying numbers, storing values in memory, and manipulating data using logical and arithmetic commands. This experience helped us develop fundamental programm ing skills and appreciate the importance of efficient and error-free code.

**Introduction to Embedded System Lab Rubrics**

* **Method of Evaluation** Viva Conducted during lab and lab reports submitted by students

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Assessment tool/ weightage/**  **(CLO, PLO)** | **Excellent**  **(10 - 9)** | **Good**  **(8 – 7)** | **Satisfactory**  **(6 – 4)** | **Unsatisfactory**  **(3 – 1)** | **Poor**  **0** | **Marks Obtained** |
| **Programming**  **(CLO1, PLO5)** | Correct Code. Easy to understand with proper comments | Correct Code but without proper indentation or comments | Slightly incorrect code with proper comments | Incorrect code with improper format and no comments | Code not submitted |  |
| **Circuit Design**  **(CLO2: PLO3)** | Circuit is simulated/implemented correctly without any errors | Circuit is simulated but implemented with minor errors | Circuit is simulated & implementation both have errors | Circuit is simulated & implemented however some components are missing/incorrect value | Circuit is simulated/implemented does not work |  |
| **Individual/ Teamwork**  **(CLO3:PLO9)** | The student/s worked effectively throughout lab to perform the assigned tasks | The student/s performed all the assigned lab tasks however one member took lead | The student/s completed all tasks however failed to work effectively | The student/s attempted all the tasks however the one member did most of the work | The student/s did not work together/at all |  |
| **Lab Report**  **(CLO4:PLO10)** | The student was able to effectively answer all questions regarding performed tasks and report provides all information without mistakes | The student was able to effectively answer all questions regarding performed tasks however the report has minor mistakes | The student was able to answer most questions regarding performed tasks and information in report is not communicated effectively | The student was able to answer some questions regarding performed tasks and report is confusing and misleading | The student was not able to answer questions regarding performed tasks and report information is incorrect/irrelevant |  |
| Total | | | | | |  |