Name : Tazmeen Afroz
Roll No : 22p-9252
COAL-LAB-TASK-10

This manual will guide you through the basics of microprocessors and programming them using Assembly Language. We will also explore how microprocessors are used in Arduino board.

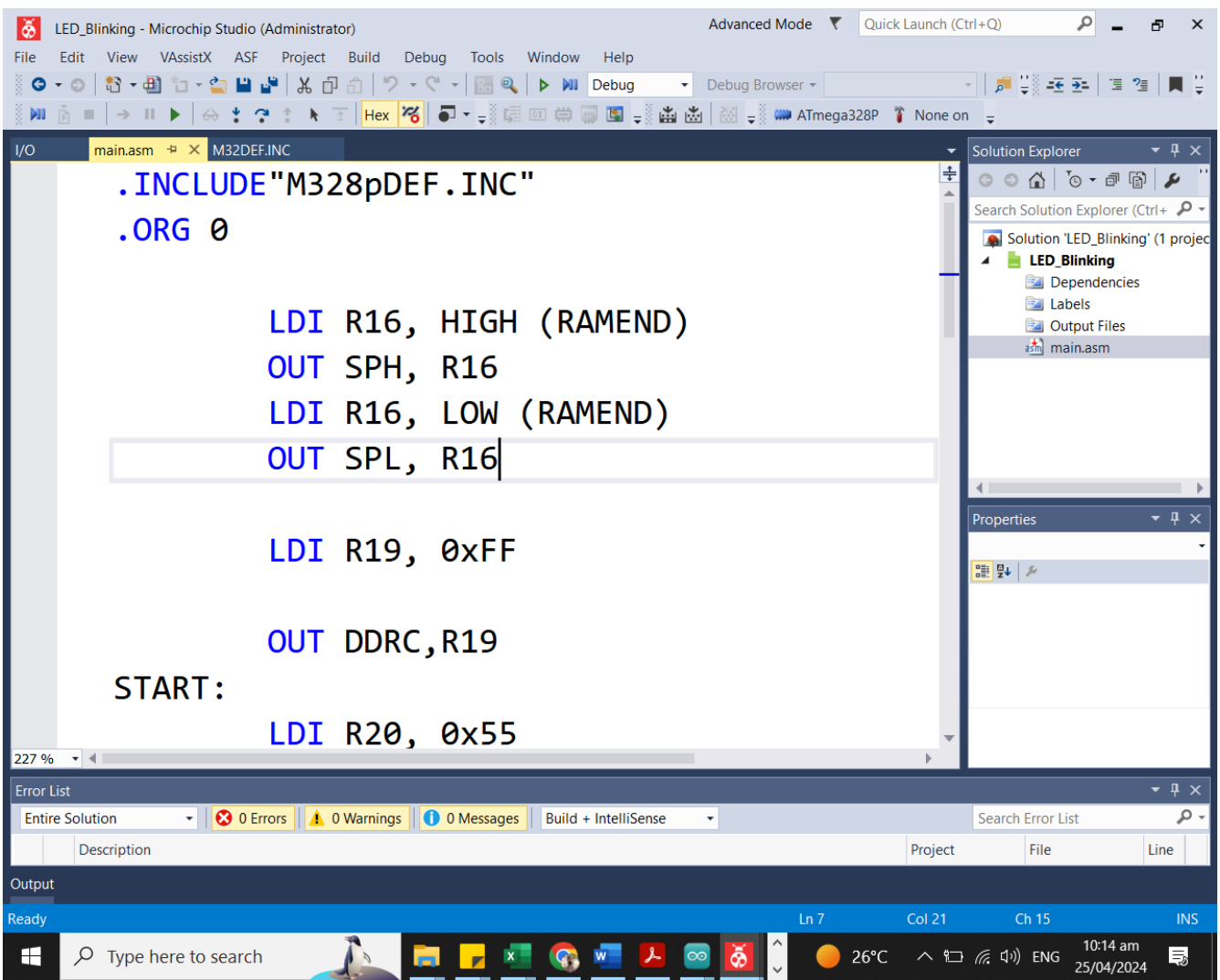
Microprocessor:

A microprocessor is a central processing unit (CPU) that contains the entire processing system of a computer on a single integrated circuit (IC). It executes software instructions and performs calculations.

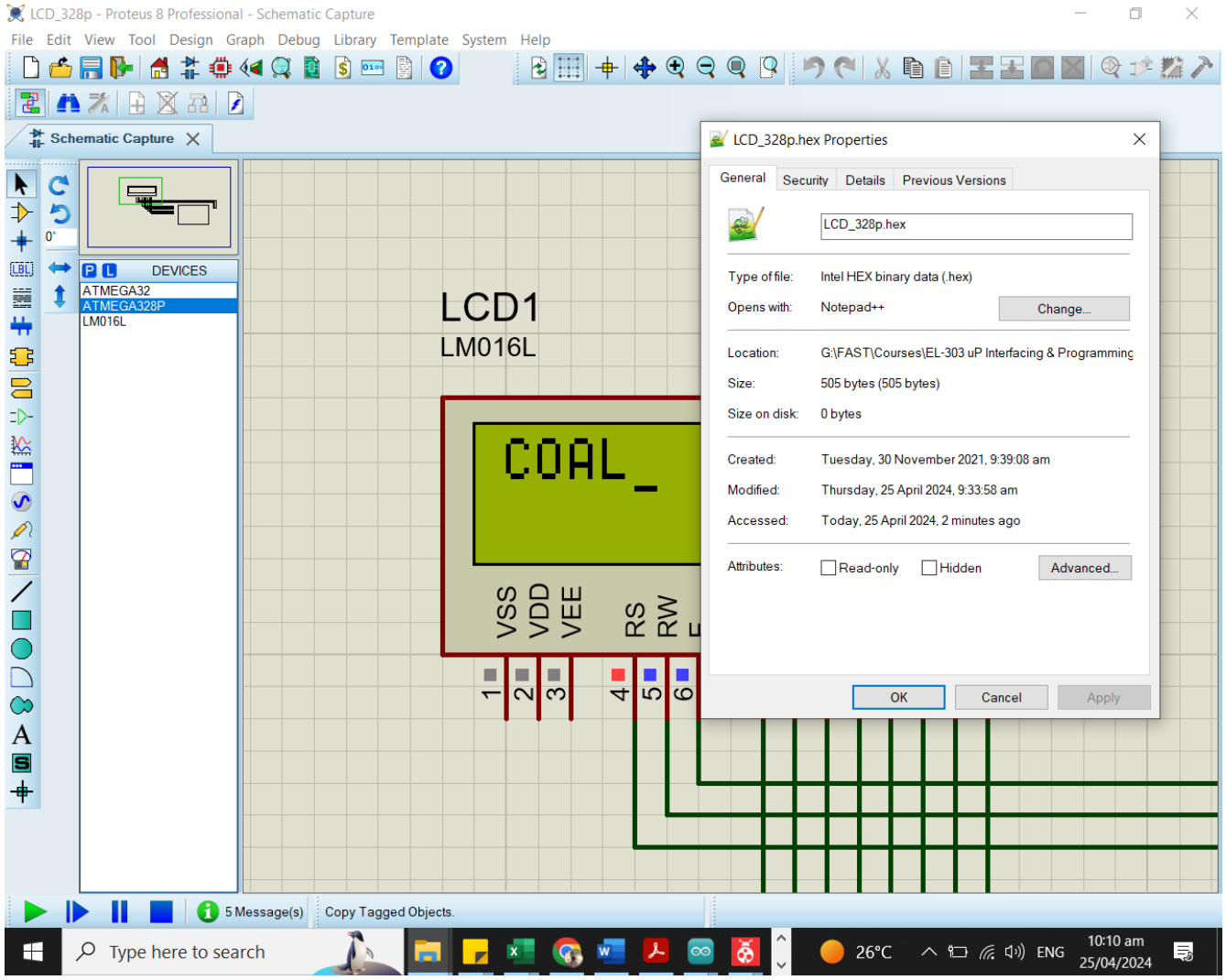
Programming Microprocessor:

To program a microprocessor, we write code in Assembly Language, which is specific to the processor's architecture. The code is then assembled into machine code that the processor can execute directly.

Example: Writing a simple "COAL" on LED program in Assembly Language for the Microprocessor:



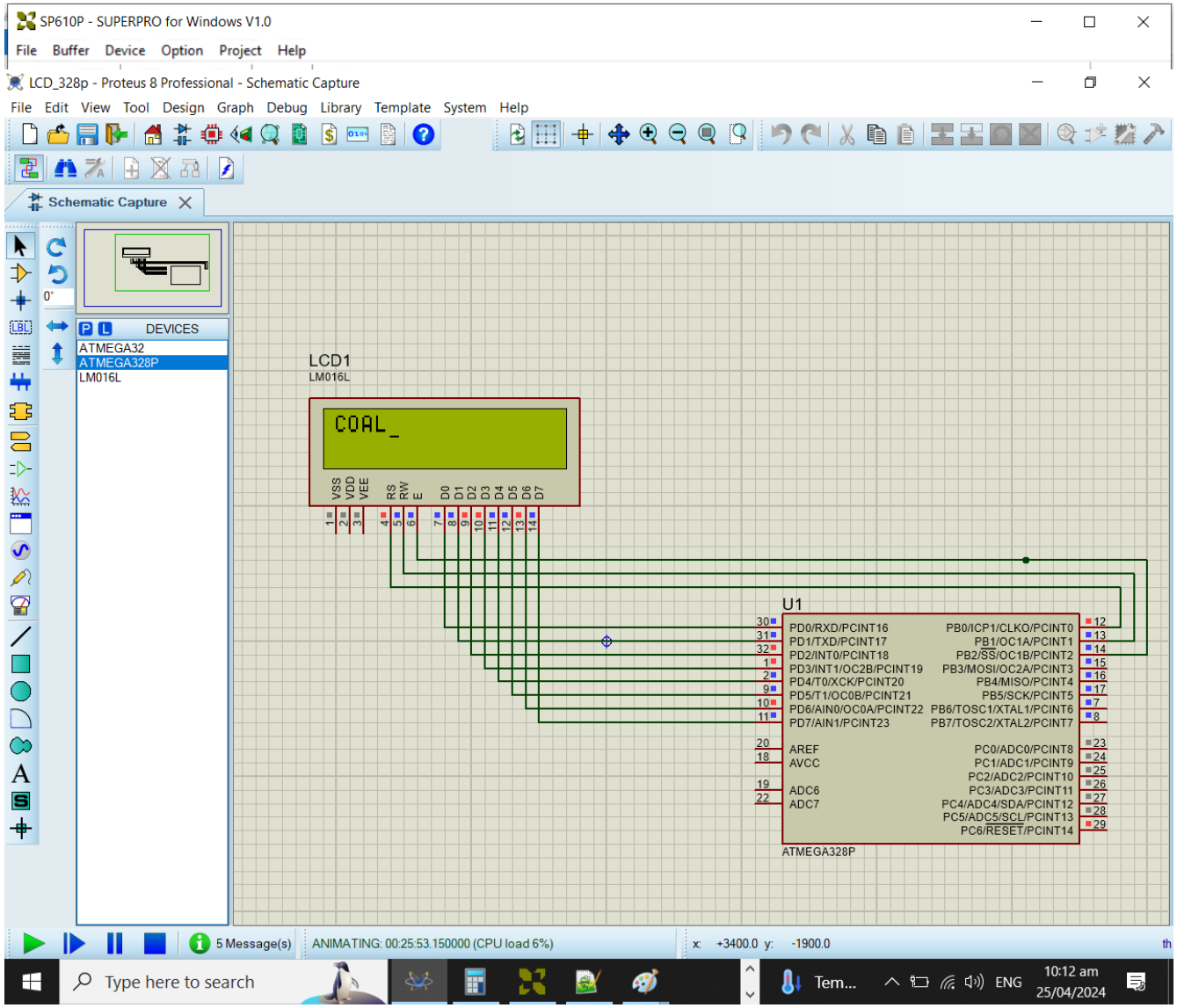
This code takes 505 bytes of memory.



Burning Code into Microprocessor:

To display the code on an LED using a microprocessor, we need to burn the code into the microprocessor using a programmer device. Here are the steps:

1. Write the Assembly Language code for the microprocessor.
2. Assemble the code into machine code using an assembler.
3. Connect the microprocessor to a programmer device (e.g., Intel 8085 Programmer).
4. Burn the machine code into the microprocessor using the programmer device.
5. Connect the microprocessor to an LED display circuit.
6. Power on the microprocessor and the LED display will show the output.

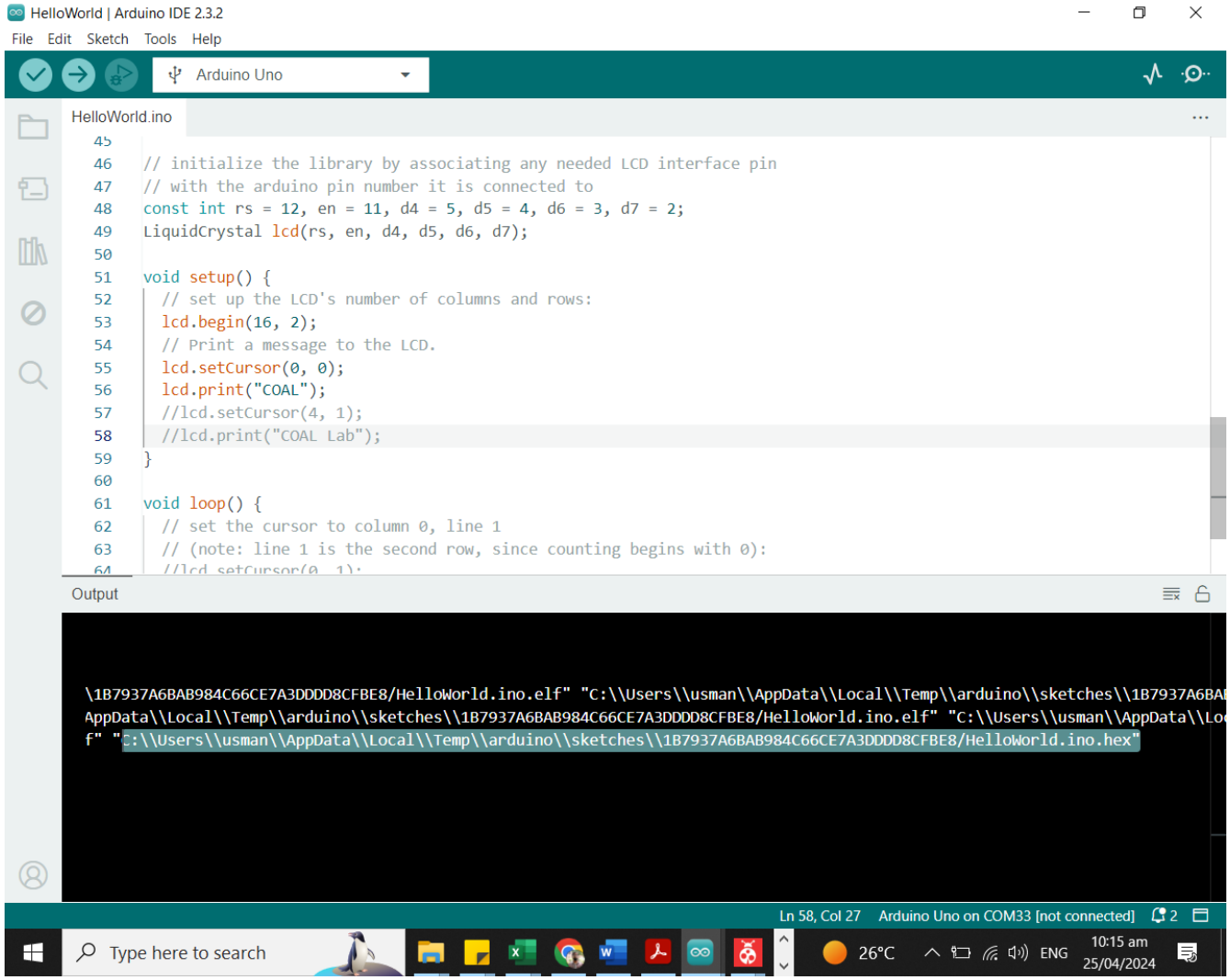


Microprocessor in Arduino:

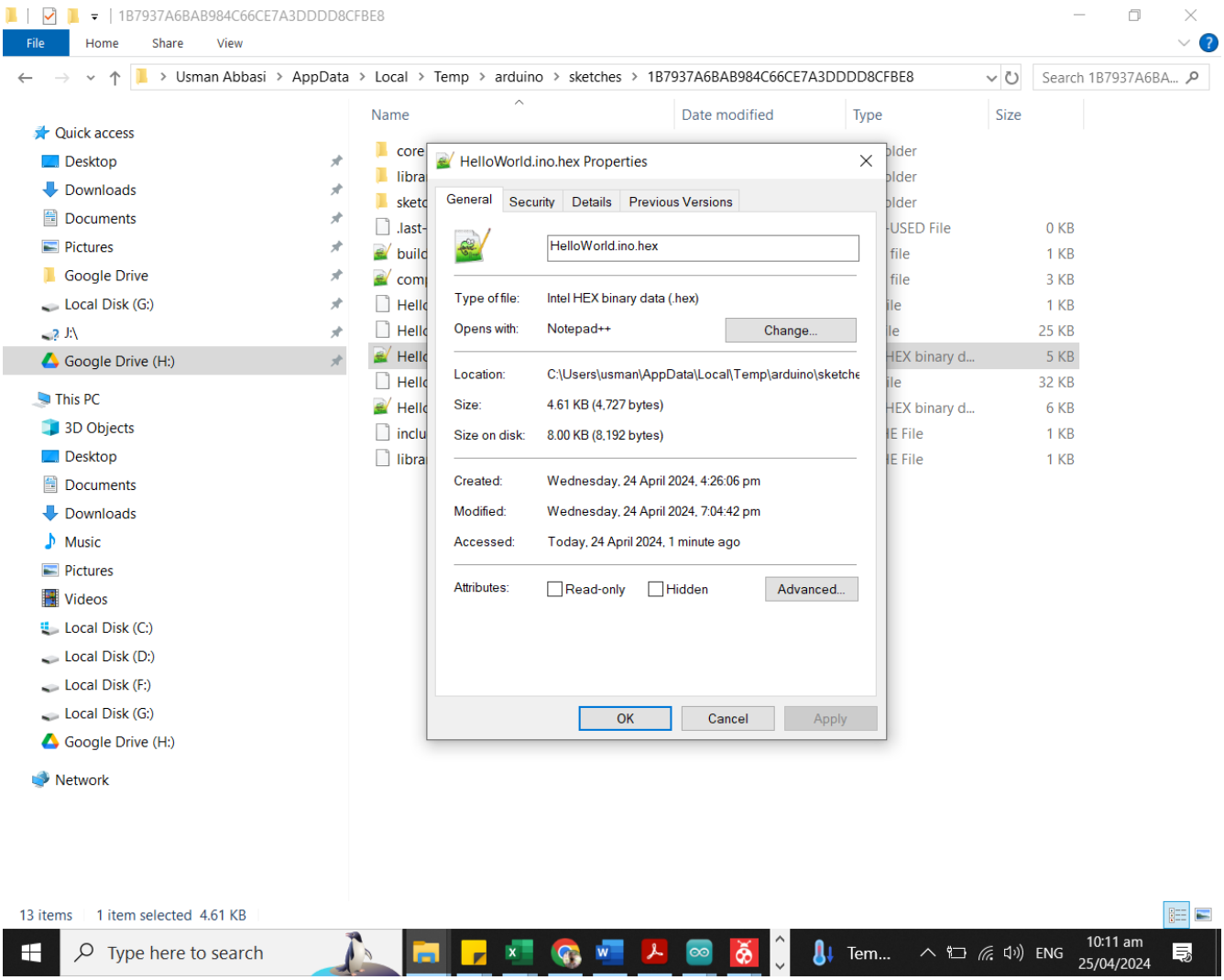
Arduino boards use microprocessors like the ATmega328P to execute code. The microprocessor is programmed using the Arduino IDE, which compiles and uploads the code to the microprocessor.

To program a microcontroller like Arduino, we write code in a high-level language like C++.

Example: Writing a simple "COAL" on LED program in C++ for Arduino:



This code takes 8.00 KB (8,192 bytes) of memory.



Comparison:

As you can see, the Assembly Language code for the microprocessor takes much less memory (505 bytes) compared to the C++ code for the microcontroller (8.00 KB). This is because Assembly Language is specific to the processor's architecture and requires manual management of memory and resources, whereas C++ is a high-level language that abstracts away many details, resulting in larger code size.