Name: Amber Khurshid

Section: BAI-4A

Roll No: 22P-9295

COAL Lab Task #10

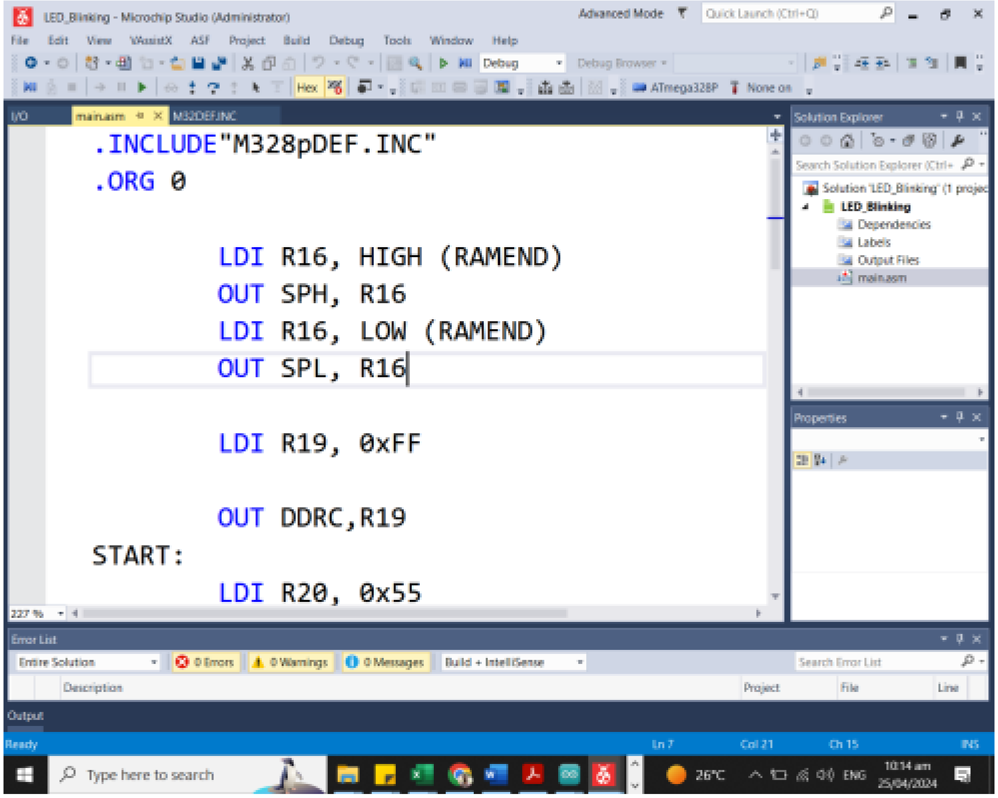
Microprocessor Basics

A microprocessor is the central processing unit (CPU) of a computer that performs calculations and executes software instructions. It is a single integrated circuit (IC) that encompasses the entire processing system. It is like the brain of a computer, controlling the flow of information and enabling a computer to perform various tasks. The microprocessor fetches instructions from memory, decodes them, and executes tasks such as running applications, performing calculations, and managing input and output devices.

Programming microprocessors

Programming microprocessors involves writing code in Assembly Language, which is tailored to the specific architecture of the processor, and then assembling it into machine code that the processor can execute directly.

Example: Creating a Simple "Turn On LED" Program in Assembly Language for Microprocessors:



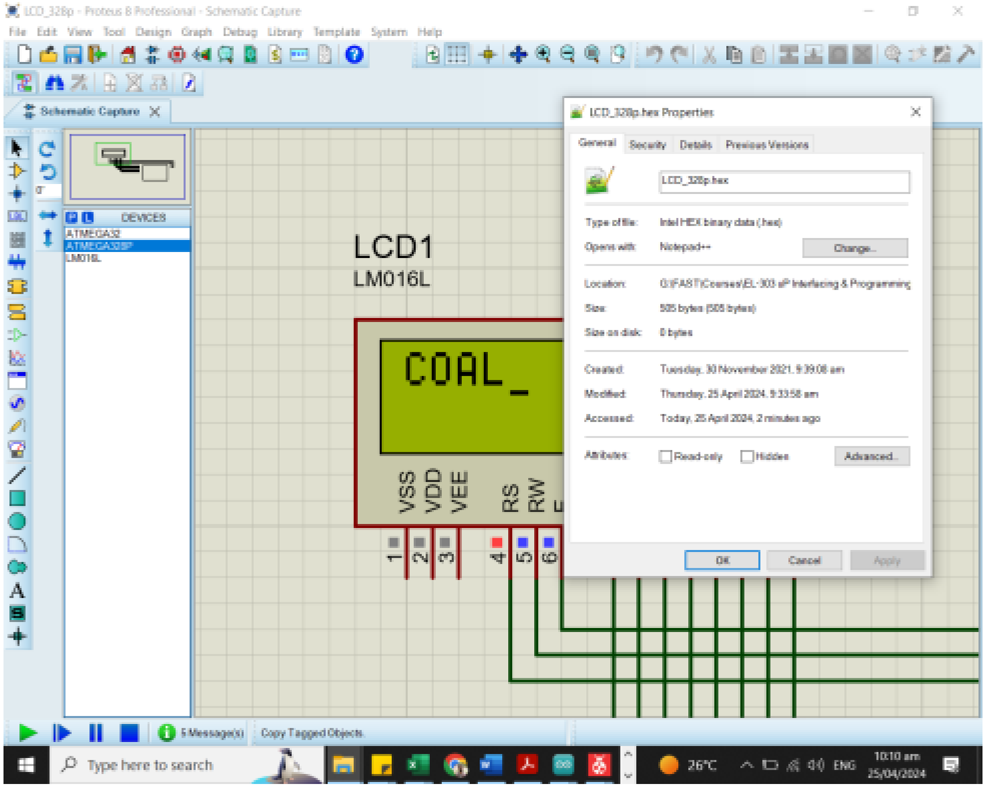
```
.INCLUDE"M328pDEF.INC"
.ORG 0

        LDI R16, HIGH (RAMEND)
        OUT SPH, R16
        LDI R16, LOW (RAMEND)
        OUT SPL, R16

        LDI R19, 0xFF

        OUT DDRC,R19
START:
        LDI R20, 0x55
```

This code takes 505 bytes of memory.



LCD_328p.hex Properties

General | Security | Details | Previous Versions

LCD_328p.hex

Type of file: Intel HEX binary data (.hex)

Opens with: Notepad++ Change...

Location: G:\FAST\Courses\EL-303 uP Interfacing & Programming

Size: 505 bytes (505 bytes)

Size on disk: 0 bytes

Created: Tuesday, 30 November 2021, 9:39:08 am

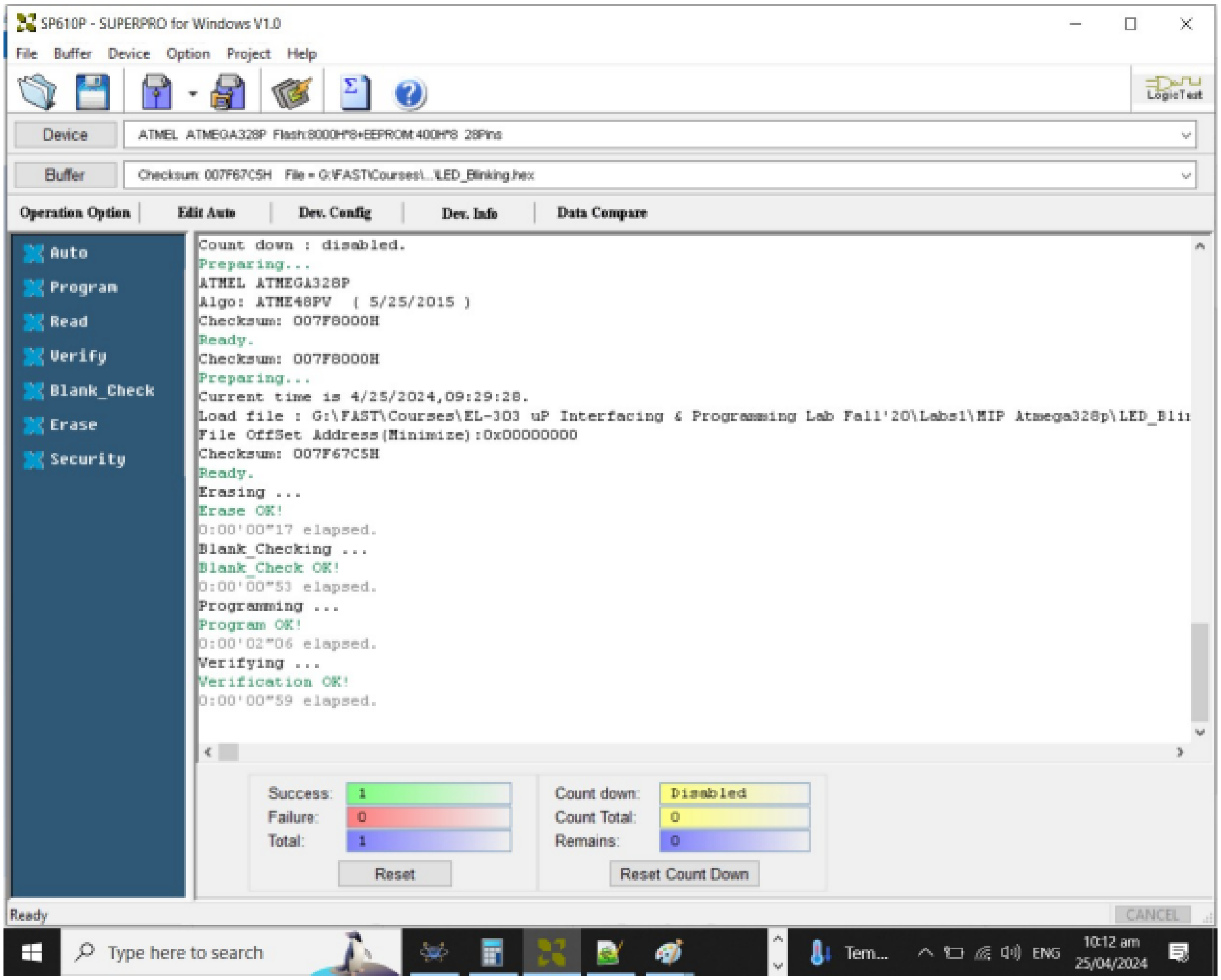
Modified: Thursday, 25 April 2024, 9:33:58 am

Accessed: Today, 25 April 2024, 2 minutes ago

Attributes: □ Read-only □ Hidden Advanced...

OK Cancel Apply

1. Assemble the Code: Utilize an assembler to convert the Assembly Language code into machine code, which the microprocessor can understand.
2. Connect the Microprocessor to a Programmer Device: Link the microprocessor to a suitable programmer device, such as an Intel 8085 Programmer, via appropriate connectors or interfaces.
3. Burn the Machine Code into the Microprocessor: Use the programmer device to transfer the assembled machine code into the microprocessor's memory.
4. Connect the Microprocessor to an LED Display Circuit: Establish connections between the microprocessor and the LED display circuit, ensuring compatibility with the microprocessor's output signals.
5. Power On the Microprocessor: Once all connections are secure, power on both the microprocessor and the LED display circuit. The LED display should now illuminate, reflecting the output generated by the microprocessor based on the burned code.



Device: ATMEL ATMEGA328P Flash:8000H*8+EEPROM:400H*8 28Pins

Buffer: Checksum: 007F67C5H File = G:\FAST\Courses\...\LED_Blinking.hex

```
Count down : disabled.
Preparing...
ATMEL ATMEGA328P
Algo: ATME48PV  ( 5/25/2015 )
Checksum: 007F8000H
Ready.
Checksum: 007F8000H
Preparing...
Current time is 4/25/2024,09:29:28.
Load file : G:\FAST\Courses\EL-303 uP Interfacing & Programming Lab Fall'20\Labs1\MIP Atmega328p\LED_Bli
File OffSet Address(Minimize):0x00000000
Checksum: 007F67C5H
Ready.
Erasing ...
Erase OK!
0:00'00"17 elapsed.
Blank_Checking ...
Blank_Check OK!
0:00'00"53 elapsed.
Programming ...
Program OK!
0:00'02"06 elapsed.
Verifying ...
Verification OK!
0:00'00"59 elapsed.
```

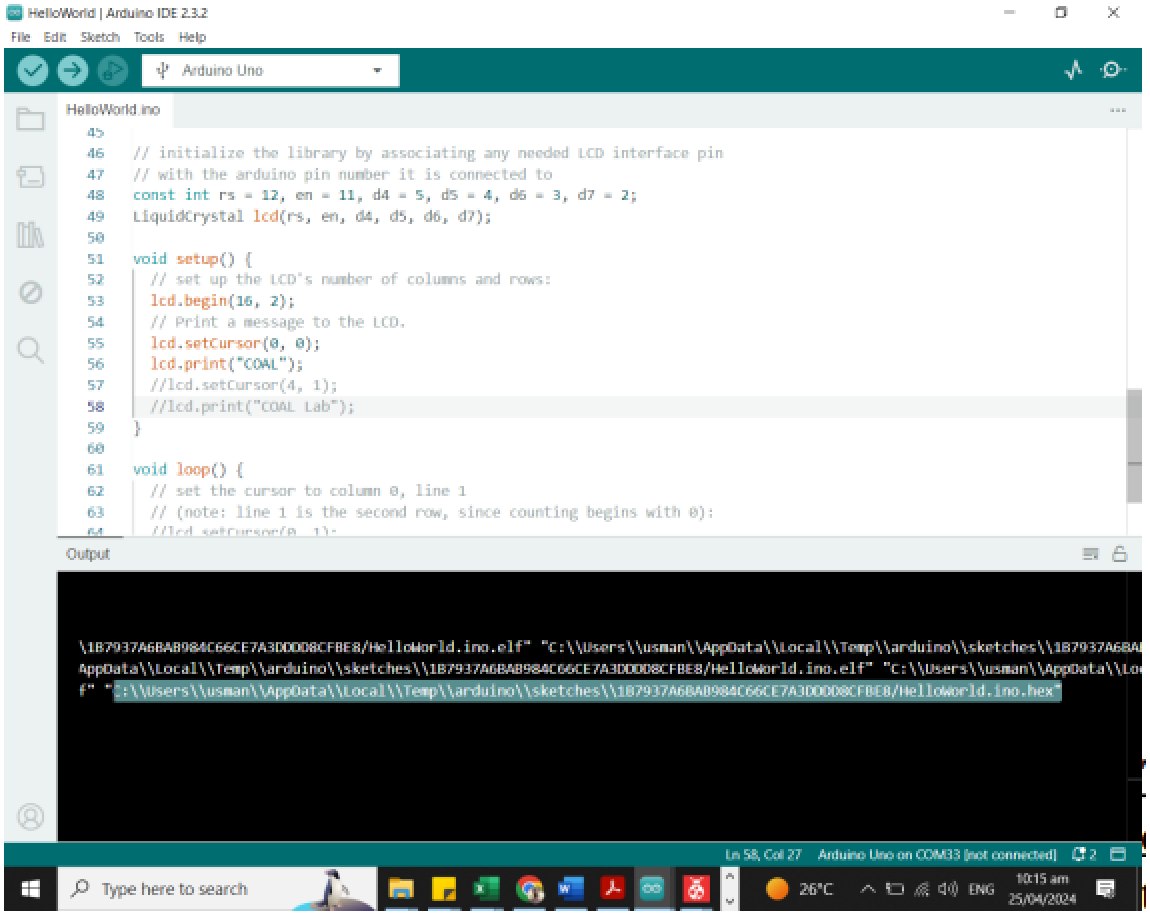
Success: 1
Failure: 0
Total: 1

Count down: Disabled
Count Total: 0
Remains: 0

Microprocessors in Arduino:

Arduino boards employ micro controllers such as the ATmega328P to execute code. The programming of these micro controllers is facilitated through the Arduino Integrated Development Environment (IDE), which compiles and uploads the code onto the micro controller.

When programming a micro controller like Arduino, we utilize a high-level language such as C++ Example: Creating a Simple "Turn On LED" Program in C++ for Arduino:



```

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.setCursor(0, 0);
  lcd.print("COAL");
  //lcd.setCursor(4, 1);
  //lcd.print("COAL Lab");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  //lcd.setCursor(0, 1);
```

```
\1B7937A6BAB984C66CE7A3DDDD8CFBE8/HelloWorld.ino.elf" "C:\\Users\\usman\\AppData\\Local\\Temp\\arduino\\sketches\\1B7937A6BA
AppData\\Local\\Temp\\arduino\\sketches\\1B7937A6BAB984C66CE7A3DDDD8CFBE8/HelloWorld.ino.elf" "C:\\Users\\usman\\AppData\\Lo
f" "C:\\Users\\usman\\AppData\\Local\\Temp\\arduino\\sketches\\1B7937A6BAB984C66CE7A3DDDD8CFBE8/HelloWorld.ino.hex"
```

Ln 58, Col 27   Arduino Uno on COM33 [not connected]

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