**Designing and building an Arduino UNO R3**

Designing Arduino UNO R3 Board (To be shown in 1 weeks’ time):  
You are to assemble your version of Arduino UNO R3 Single Board. The following items are to be physically inspected by the instructor and then incorporated into the report:  
1. The Designed schematic Diagram in Proteus / or any other popular schematic software e.g. Eagle cad etc. [5 marks]  
2. The Bread Board implementation of regulated 5 and 3.3V voltage circuits [5 marks]  
  
Building Arduino UNO r3 Board (to be shown in 2 weeks’ time:  
1. The Bread Board implementation of the complete circuit (regulated power +Microcontroller) [5 marks]  
2. PCB Simulation, containing proper Layer Layouts and on the Silk Screen Layer, your name and registration Number is written [5 marks]   
3. Code running on the above breadboard implemented complete circuit. [5 Marks]

**Introduction**

A popular microcontroller development board in the electronics sector is the Arduino Uno R3. The board's ATmega328p microcontroller serves as its foundation, and it includes a number of electronic components, including voltage regulators, a crystal oscillator, a reset button, and a built-in LED, to make programming and connecting to other electronics simple.

The Arduino Uno R3 has an easy to implement design. The board's compact size and easy-to-use development environment have made it a popular choice for various projects, ranging from robotics to home automation.

In this project, an Arduino Uno R3 board is made using an ATmega328p microcontroller, a variety of electronic components, and Proteus software for schematic and PCB design. The board has a 5V regulator circuit, a 3.3V regulator circuit, a reset button, a crystal oscillator, a built-in LED, and other components required for interacting with other electronic devices with microcontrollers.

**Methodology**

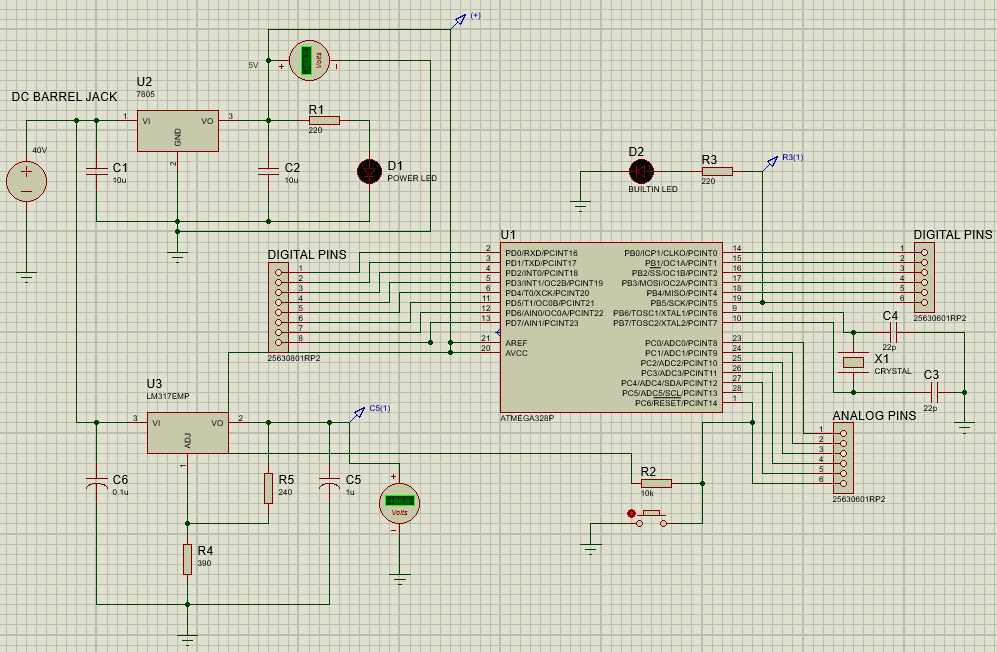
Selection of hardware components: The first step include choosing and getting the hardware components. The list of components is shown below:

|  |  |  |
| --- | --- | --- |
| **Component** | **Quantity** | **Purpose** |
| ATmega328p | 1 | Microcontroller that serves as the brain of the project |
| Capacitors | 2 (ceramic 22pF)  2 (ceramic 10uF)  2 (polar 0.1uF and 1uF) | Used to stabilize the voltage and filter out noise |
| 7805 voltage regulator | 1 | Converts the input voltage to a stable 5V supply voltage |
| Resistors | 2 (220ohm)  1 (10k)  1 (390ohm)  1 (240ohm) | Used to limit current and set the values of various components |
| Push button | 1 | Used for the reset feature of the microcontroller |
| LED | 2 (Red) | Used for indication and feedback in the project |
| Crystal oscillator | 1 (16M Hz) | Provides a stable clock signal for the microcontroller |
| Female Header Pins | 1×14 | For connections of digital pins |
| Female Header Pins | 1×6 | For connections of analog pins |
| Atmega328p  DIP socket | 1 | For placing Atmega328p when soldering in PCB or Perf board |
| LM317 voltage regulator | 1 | Converts the input voltage to a stable 3.3V supply voltage |
| Connecting wires | - | For connections |

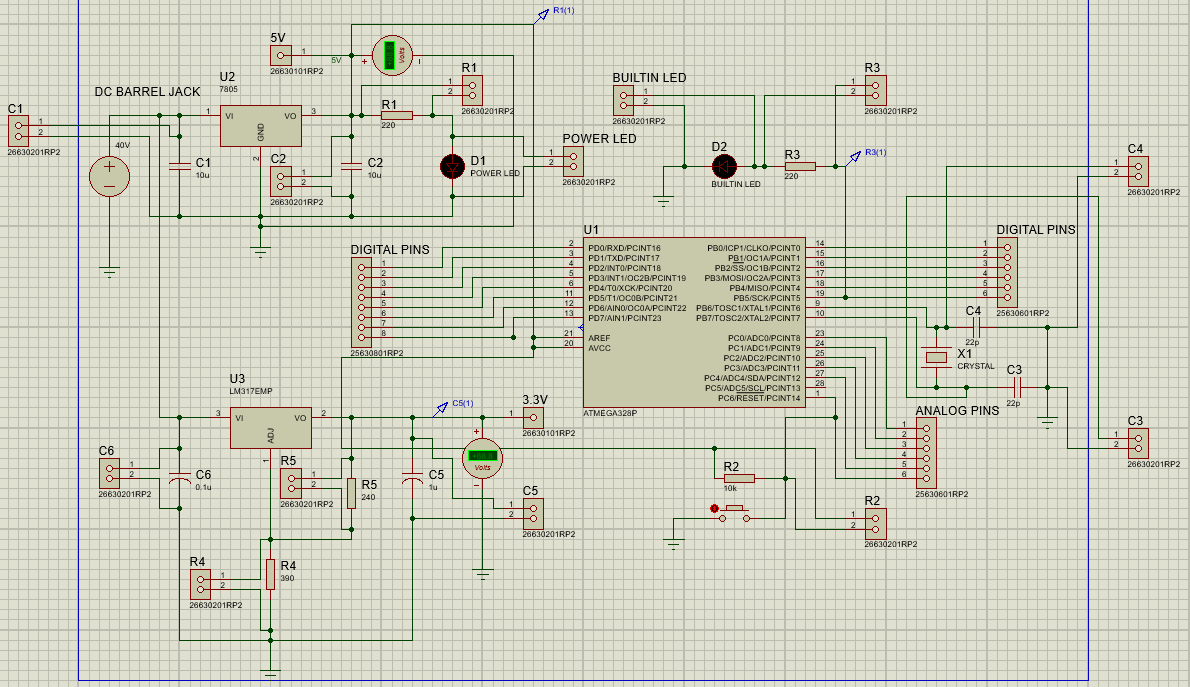
Working of each component: The working of each component is as follows:

* ATmega32: The microcontroller acts as the project's central processing unit. It is in responsible for executing the program code and controlling many inputs and outputs.
* 7805 and LM317 voltage regulator: The 7805 and LM317 voltage regulator is used to convert the input voltage to a stable 5V and 3.3V supply voltage respectively. This ensures that the microcontroller and other components receive constant and reliable power.
* Capacitors: Capacitors are used in power supplies to stabilize voltage and filter out noise. They are also used to filter out undesirable signals in various circuit components.
* Resistors: Resistors are used to limit current and set the values of various circuit components. They are also used to pull up and pull down the microcontroller's input pins.
* LED: The LED is used for indication and feedback. It is used to identify the project's status or to provide visual feedback to the user.
* Crystal oscillator: The crystal oscillator gives a steady clock signal to the microcontroller. This ensures that the microcontroller runs at a steady speed and that the program code runs smoothly.
* Push button: The push button is used to access the microcontroller's reset capability. When pressed, it resets the microcontroller and restarts the program code from the beginning.

Circuit Design: The second step in designing the Arduino Uno R3 board was to create the schematic in Proteus. The Arduino Uno R3 is a microcontroller board based on the ATmega328p chip. The board is intended to be simple to use, with a straightforward interface and a large range of input and output choices. The schematic shows the connections between all of the electronic components and provides a clear picture of how the circuit is structured. The schematic includes ATmega32 microcontroller, 16 MHz crystal oscillator, Two 22 pF capacitors, 5V regulator circuit (7805 regulator, two 10 µF capacitors, and a 220 ohm resistor), 3.3V regulator circuit (LM317, two capacitors of 0.1 and 1 µF and one 240 and 390 ohm resistors), 10K resistor and push button for reset, Built-in LED with 220 ohm resistor and various other passive components such as resistors, capacitors. The schematic also includes the female headers for analog and digital pin connections of the microcontroller.



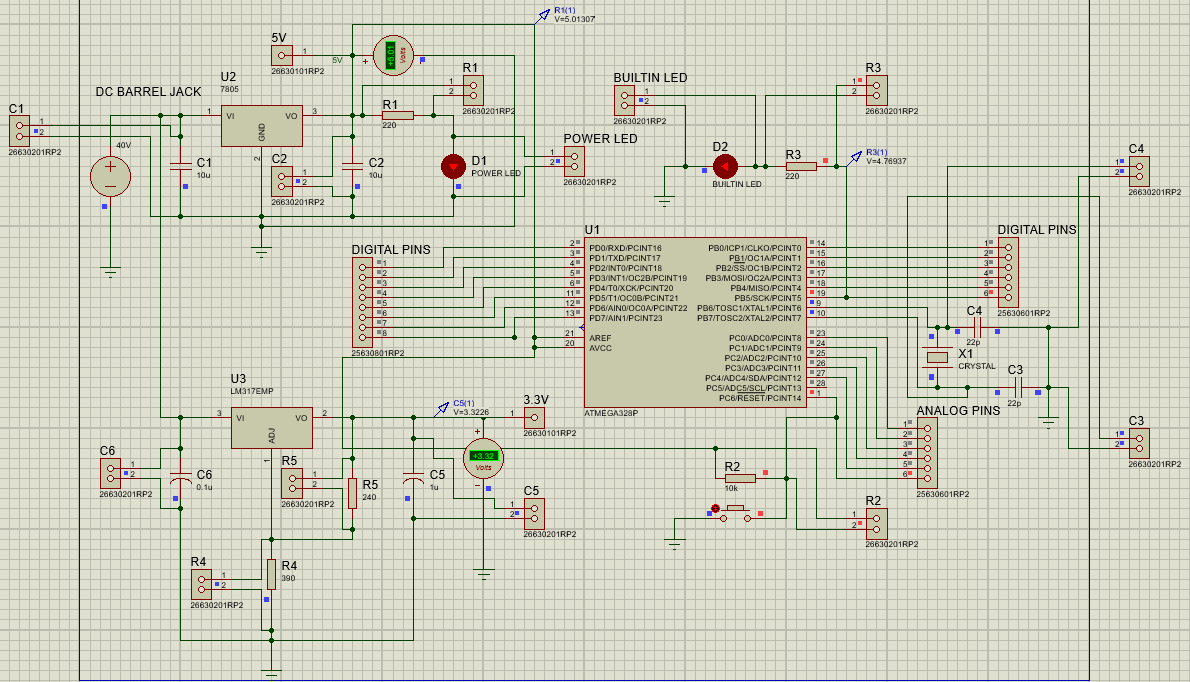
For components having no foot print, a one or two pin female header is used. This can be shown in figure below:

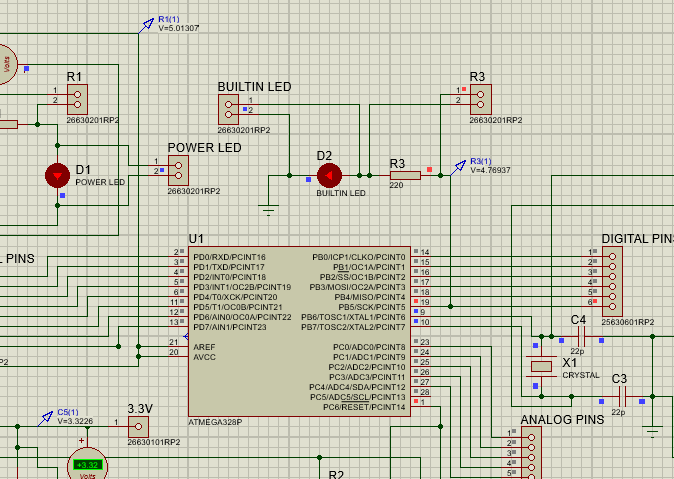


Working and Testing: After the schematic was completed, it was tested to ensure it functions correctly. Testing involves checking the voltage output of the voltage regulators, testing the crystal oscillator's frequency, and testing the built-in LED by uploading a simple program that blinks the LED.

LED blinking:

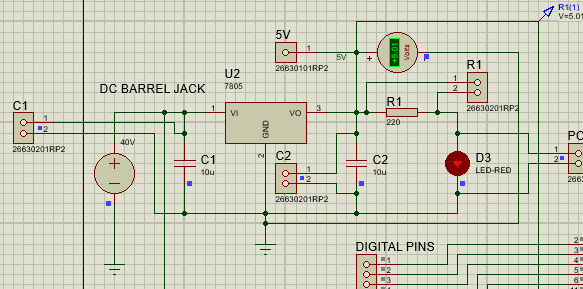
A built-in LED on the board is also attached to one of the microcontroller's digital pin 13. A 220 ohm resistor connects the LED to the microcontroller, limiting the current through the LED.





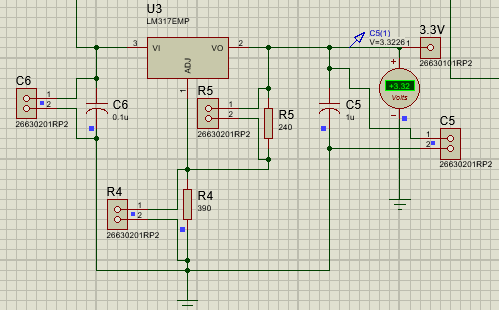
5V regulator circuit:

The 7805 is a popular linear voltage regulator that produces a constant 5 volts DC from a greater input voltage. There are three pins on the 7805 regulator: input, ground, and output. The input voltage is connected to the input pin, while the ground pin is connected to the circuit ground. The output pin provides the output voltage. The two 10uF capacitors are linked between the input and ground pins, as well as the output and ground pins. These capacitors filter out any noise or ripple in the input or output voltage and keep the regulator stable. To limit the current flowing through the regulator, a 220 ohm resistor is connected in series with the input voltage. This resistor ensures that the regulator does not get damaged due to overcurrent.



3.3V regulator circuit:

The LM317 is a popular adjustable voltage regulator that can be adjusted to produce a range of voltages by connecting external resistors and capacitors to its pins. Connect a 240 ohm resistor between the LM317's output pin and it’s adjust pin, and a 390 ohm resistor between the adjust pin and ground to build a 3.3V regulator circuit. A 0.1 microfarad capacitor can be connected between the LM317's output and ground pins, and a 1 microfarad capacitor can be connected between the input and ground lines. When the LM317 is powered up, the 1 microfarad capacitor charges up, giving initial stability and preventing oscillations. By regulating the current flow through the 240 ohm and 390 ohm resistors, which establish the feedback voltage at the LM317's adjust pin, the output voltage is regulated to 3.3V. The 0.1 microfarad capacitor aids in the reduction of any high frequency noise on the output voltage.



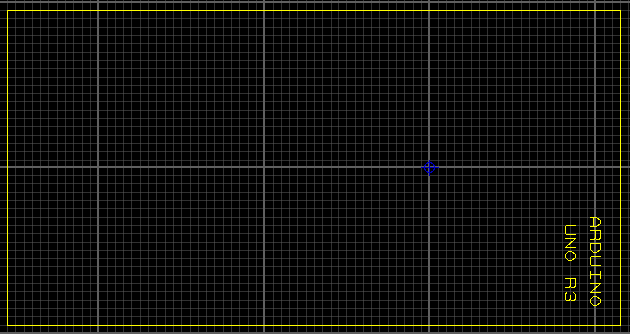
PCB Layout Design: After creating the schematic and simulating it successfully, the next step was to design the PCB layout as well in proteus. This involves arranging the components on the board and routing the connections between them.

First, the board edge was sketched. The components were then placed by using the auto place command. The components were arranged so that the routing/wires did not overlap.

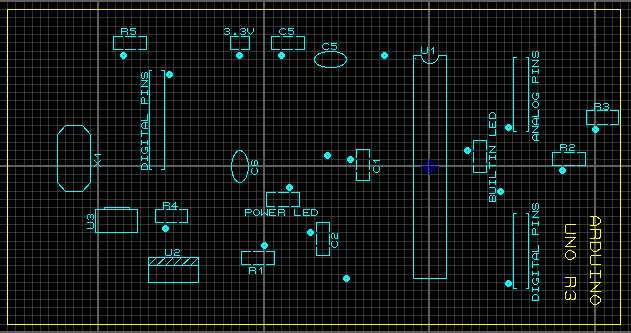
The connections were routed using Proteus' auto-routing tool, which automatically routes the connections based on the schematic and also using manual routing. After routing, the board's connections were reviewed to ensure there are no short circuits or issues with the placement of components. The routing trace style T20 is used.

Following are the pictures of PCB designing:

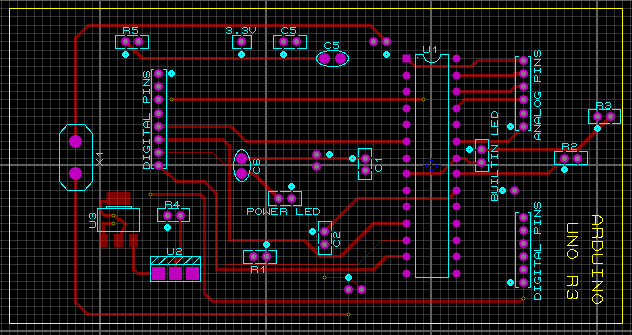
PCB board edge:



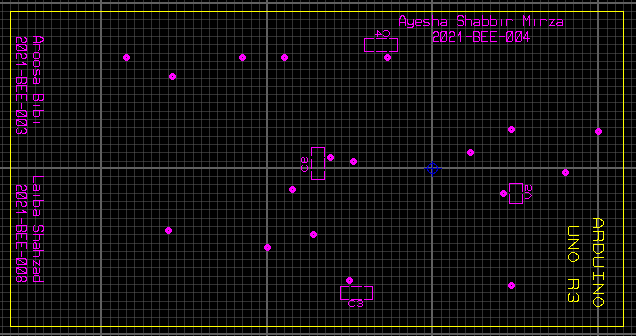
Top silk layer:



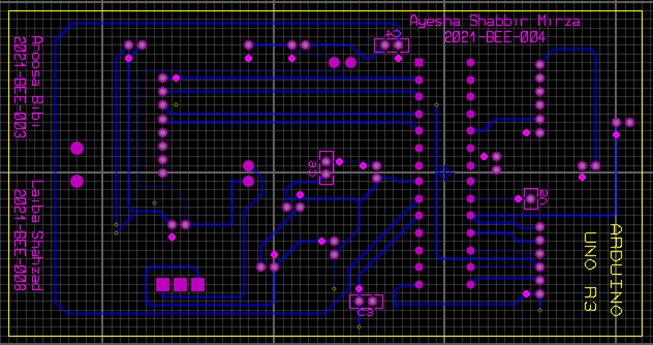
Top copper layer with top silk:



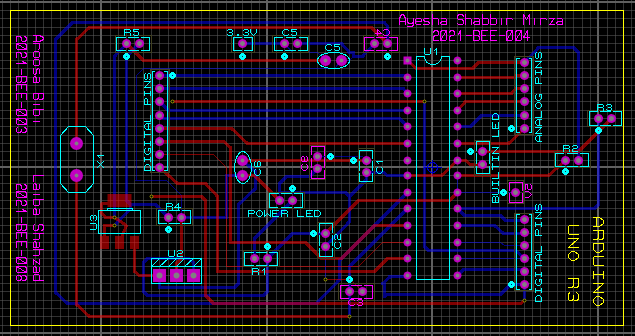
Bottom silk layer:



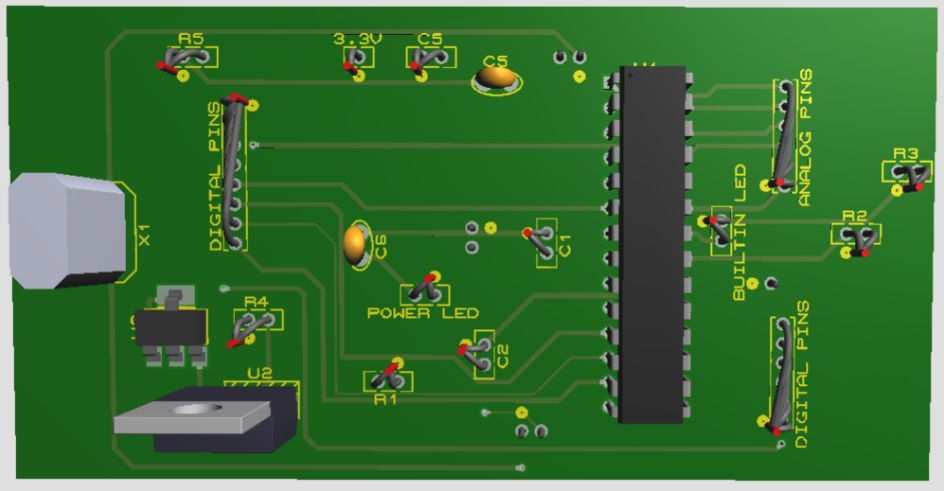
Bottom copper layer with bottom silk:



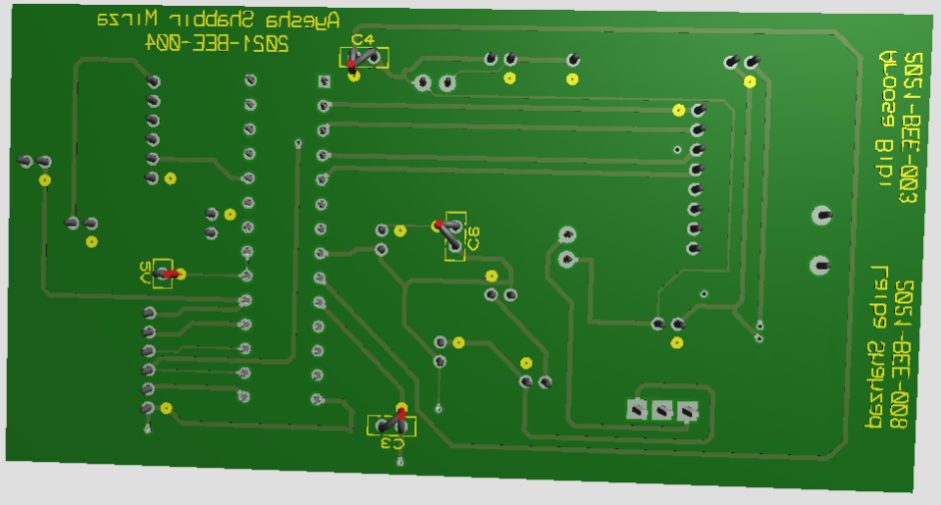
Complete PCB:



3D Top view:

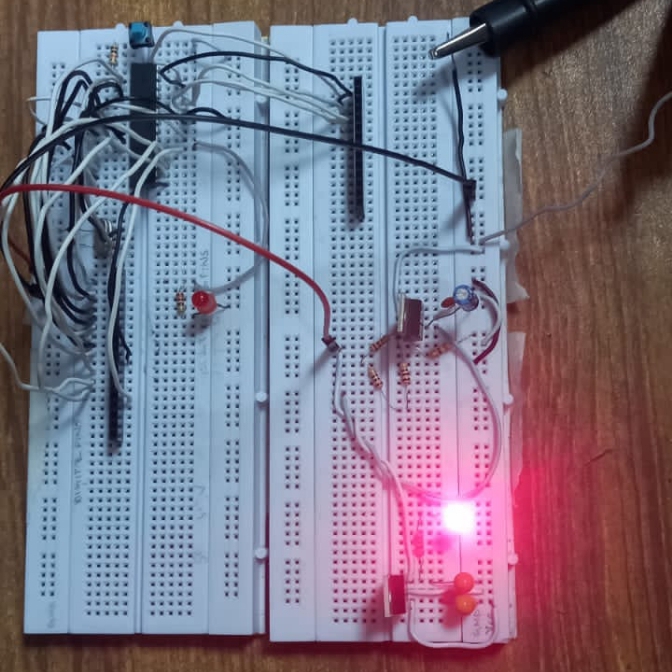


3D Bottom view:



Breadboard implementation: After the schematic is finalized and simulated successfully, the circuit is implemented on breadboard and IC Atmega328p is programmed. And then, converted into PCB, the circuit is soldered on perf/Vero board.

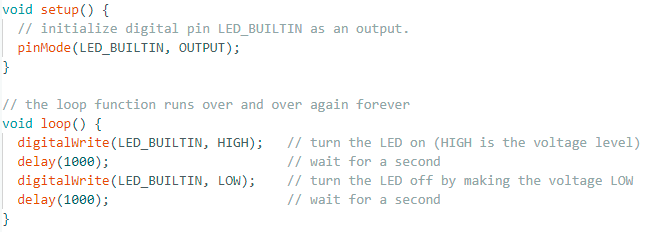
1. The components are then placed on the breadboard, and connections were made same as shown in schematic and software simulation.



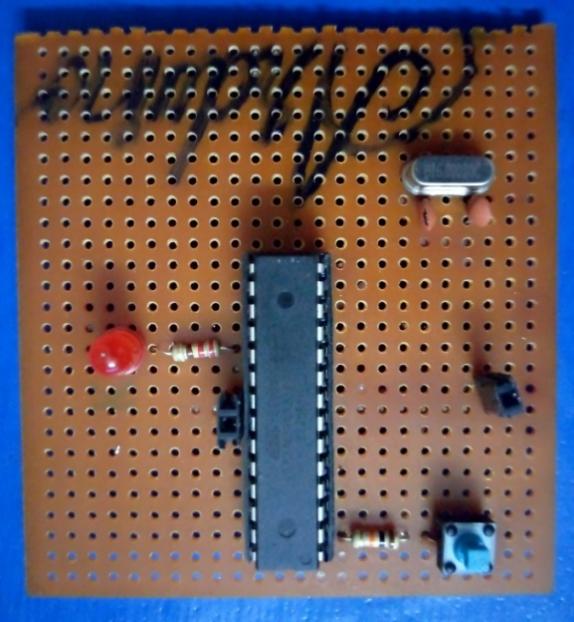
1. The circuit is programmed through following steps:
2. Connect Arduino with your Laptop or your PC.
3. Now, open Arduino IDE.
4. Go to the File tab then click on examples and click on ArduinoISP.
5. Upload this ArduinoISP code to your Arduino UNO Board.
6. After the ArduinoISP code is uploaded, go to the Tools tab and then go to the Programmer option and select Arduino as ISP.
7. Connect the ATmega328P IC with the Arduino UNO board as described below:

* Connect pin RESET, D13, D12, D11 and D10 of Arduino with pin 16, 19, 18, 17 and pin 1 of Atmega328p respectively.
* Now, connect 5V of Arduino with pin 7(VCC) & 20(AVCC) of Atmega328p. And connect GND of Arduino with pin 8(GND) & 22(GND) of Atmega328p.

1. Go to the Tools tab and select port & Arduino Pro or Pro Mini Board.
2. After that, there will be an option of processor in Tools tab. Select Atmega328p (5V, 16 MHz).
3. Go to the Tools tab again and then click on burn bootloader.
4. The bootloader is burned in the IC.
5. When, the bootloader is burned in IC successfully, the led at pin 19 of Atmega328p i.e. BUILTIN\_LED and the BUILTIN\_LED of Arduino will blink in sync. The video of circuit while burning bootloader is attached in submission tab.
6. As the bootloader is burned, now, we can program our IC Atmega328p.
7. For, programming, remove all connections except 5V and GND.
8. Connect pin D10 of Arduino with pin 1 of Atmega328p and pin D0 & D1 with pin 2 & 3 of Atmega328p.
9. Now, upload the blinking led code (that is available in File>Examples>Basics>Blink) by clicking upload using programmer in Tools tab.



1. Finally, Your IC is programmed. ☺
2. The circuit is converted into PCB using Vero board by soldering components on it.



**Conclusion:**

The Arduino Uno R3 development board is a great example of a microcontroller development board that is designed for easy prototyping and project implementation. The design of the board comprises almost all of the components required to program and interface with other electronic devices. This assignment report describes the procedures involved in building an Arduino Uno R3 board with Proteus, from schematic design to breadboard implementation and testing.

Note:

Guidance & information for this assignment is taken from various sources.