ZERO PROJECT REPORT

Muhammad Shehroz Khan
211224
BEMTS – 8– A

PROJECT SUPERVISOR

SIR UMER FAROOQ

MAAM Sadia Saeed

DEPARTMENT OF MECHATRONICS ENGINEERING

FACULTY OF ENGINEERING

AIR UNIVERSITY, ISLAMABAD

Table of Contents

Objectives:

- To learn the basic infrastructure of a microcontroller i.e., Arduino UNO.
- To learn Implementation of Generic board on proteus.
- To Learn Implementation of Generic Board on PCB Board.
- To learn troubleshooting of a circuit on proteus as well as on bread board and PCB board.

Component Used:

S No.	Name	Description	Quantity
1	ATMEGA328P	IC	1
2	LM7805	Voltage Regulator	1
3	Capacitor	22pF, 10uF, 0.1uF	2+2+2
4	Resistor	330Ohms, 10k Ohms	2+1
5	Diode	IN4001	1
6	Crystal Oscillator	6MHz	1
7	Female Header		-
8	Push Buttons		1
9	LEDs		2
10	PCB board		1
11	$FeCl_3$		250g
12	IC Base	28 Pins	1
13	Jumper Wires		-
14	Arduino UNO	DIP	1
15	Arduino UNO cable		1

Tools Used:

- Soldering device
- Drill Machine

- Iron
- Sticky paper / photo paper
- HP LaserJet Printer
- Arduino IDE software
- Bucket for etching

Introduction:

This Project is all about Understanding the basic layout of generic Board (Arduino UNO) and its implementation on PCB.

Generic Board:

This is generally a board which have 2 main components. The 1st component is a circuit which contain a programmable Chip. The 2nd component is some sort of programmable Software. This software is used to write the code according to the need and then by some data cable, it is transferred to the Chip. This kinda work is done because they are efficient for small work, i.e., we always can't use a big PC. This work doesn't require any pro programmer. It just uses some basic logic which is know as atmega 328p.

Project Description:

This project is based on understanding Microcontrollers. Microcontrollers are specified devices that we can say a part of embedded systems that works for a specific programme, so we can say that they are minicomputers.

The selected Microcontroller for this project is Arduino Uno. This is a microcontroller board based on the ATmega328P. It has 20 digital input/output pins.

The input/Output Ports are named as B, C, and D. Port B and D are digital while Port is C is Analogue. Port D has 8 pins while port C has 6 pins.

There are 2 ports for a 16 MHz resonator (Crystal), a USB connection, a power jack, an in-circuit system programming, 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 an AC-to-DC adapter or battery to get started.

The code can be uploaded first time to the IC by burn Bootloader. There are two ways to do that, i.e., by using Arduino DIP and Uploading code using TTL.

The 16MHz crystal oscillator is used to provides an external clock of 16MHz to Atmega328p as with its default frequency.

The Voltage regulator, LM7805, regulate the Voltage to 5V, that is enough voltage for the IC.

The diode is used for reverse voltage protection.

Two LEDs will be used. On for power supply and other for the Arduino IC.

The push will be used as a reset that brings the microcontroller to its initial condition.

Component Description:

IC (ATEMGA328P):

AtMega328P are basically manufactured by Atmel Microchip, and they follow the RISC Architecture. This is an 8-bit AVR microcontroller chip and has a flash-type program memory of 32KB. They are single chip, highly efficient and low power dissipation. They can operate on voltage from 1.8 to 5.5V and temperature -40 to 85 Degrees in Celsius. It has an EEPROM memory of 1KB and its SRAM memory is 2KB.

This Chip contain 28 Pin. It has got 20 input/output ports, the analogue port is the port C while the digital is port B and D. it has 1 Vcc, 2 grounds and reset pin. The ports and pins can be seen in the figure below.

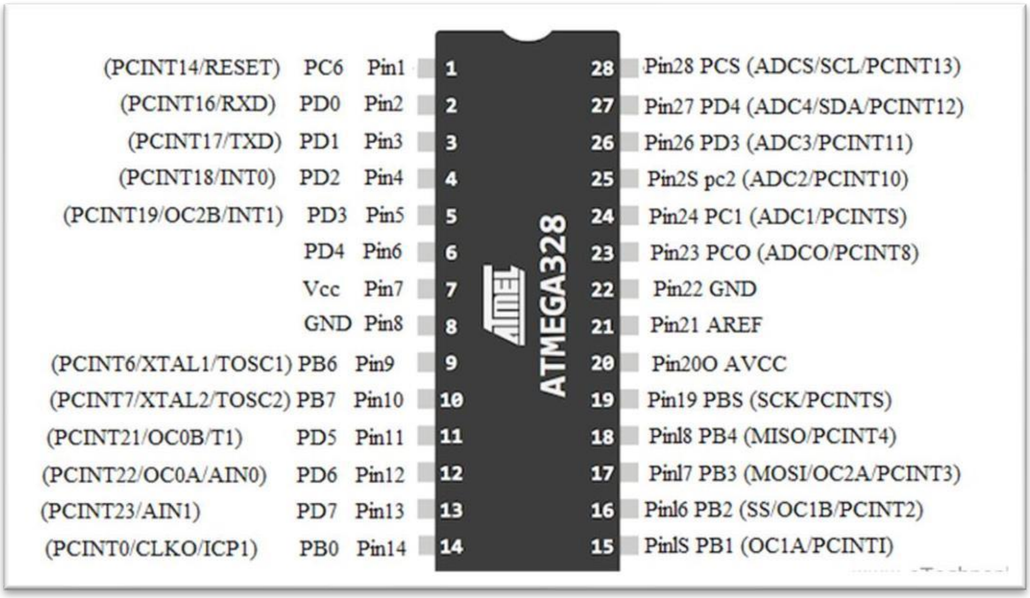


Figure 1: AtMega328P Pins

Voltage Regulator (LM7805):

It is a member of the 78XX series of voltage regulator ICs. The XX present in 78XX represents the value of the fixed output voltage that the IC provides. So, 7805 IC, has +5V DC regulated power supply. It is a fixed linear voltage regulator. The IC will give 5V output and it can bear voltage level up to 35V. it has got a heat sink because voltage more then 5V is converted to heat which is then exhausted.

This IC has 3 pins.

Pin 1:

This is the **Input pin**. It gives the input voltage. Its value may vary from 7V to 35V. We apply an unregulated voltage to this pin for regulation. For 7.2V input, the PIN achieves its maximum efficiency.

Pin 2:

This is the **Ground pin.** Ground of power supply is connected to this pin.

Pin 3:

This is the **Output Pin.** This pin is used to take the regulated output.

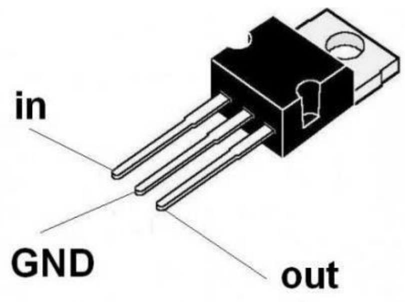


Figure 2: LM7805

Capacitors:

It is a device that stores electrical energy in an electric field. It is a passive electronic component. It has two terminals. It has an effect called capacitance. We have used to 3 capacitors.

22uF:

It is used with the crystal circuit.

10uF & 0.1uF:

These two are used with voltage regulator circuit.

Figure 3: Capacitors

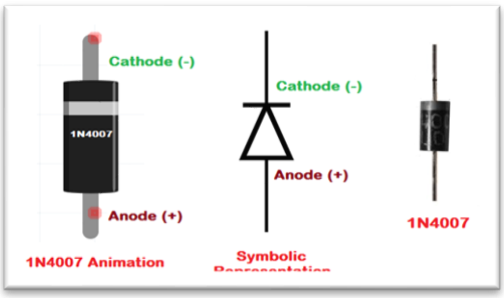
Resistor:

Resistor basically provides resistance to the flow of current. It is a passive device having two terminals. We have used two resistors in our circuit.

Diode (IN4001):

It is a semiconductor device that is mostly used as a switch for current. It allows current to flow easily in one direction and restricts current from flowing in the opposite direction. They are also known as rectifiers because they change alternating current (ac) into pulsating direct

current (dc). They have polarity, determined by an anode (positive lead) and cathode (negative lead). Most diodes allow current to flow only when positive voltage is applied to the anode.



Crystal Oscillator (16MHz):

It provides a clock signal to microcontroller Atmega328P used in the circuit. This provides a square wave signal which determine the time required for each T state. As in general Arduino board has 16Mhz frequency crystal hence takes 1/16 uses to run 1 T state.

Figure 6: 16MHz Crystal

Female Header:

They are used to help us patch the circuit easily. A female header is one, into which a wire can be put inside. It is used for the C, D, B ports in the circuit.

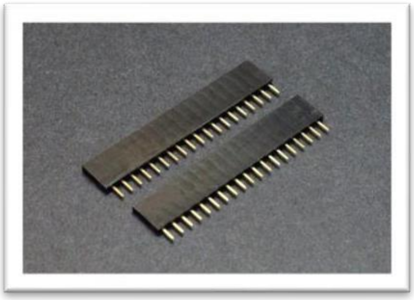


Figure 7: Female Headers

Push Button:

It is a button that only work when it is pressed. It is used instead of a reset female header.

Figure 8: Push Button

LEDs:

2 LEDs are used, one for power supply and the other is integrated with IC of AtMega328P. They will let us know if the circuit is working. The long pin of LEDs is High while the small pin is ground.

Figure 8: LED

PCB board:

PCB stands for Printed Circuit Board. It has got conducting of copper and insulated layer.it mechanically and electrically support a circuit. It is used for permanent work after bread board implementation.

$FeCl_3$:

It is a chemical used for etching. It engraves the circuit boards.

Figure 12: $FeCl_3$

IC Base:

The IC Base can be use for different purposes. Sometimes we need to replace the IC with another due to damage of the IC, sometimes we use another type of component. The below IC is of 28 pins

Figure 13: IC Base

Jumper Wires:

They will be used in patching the circuit on bread board. Male to male are required.

Figure 14: Wires

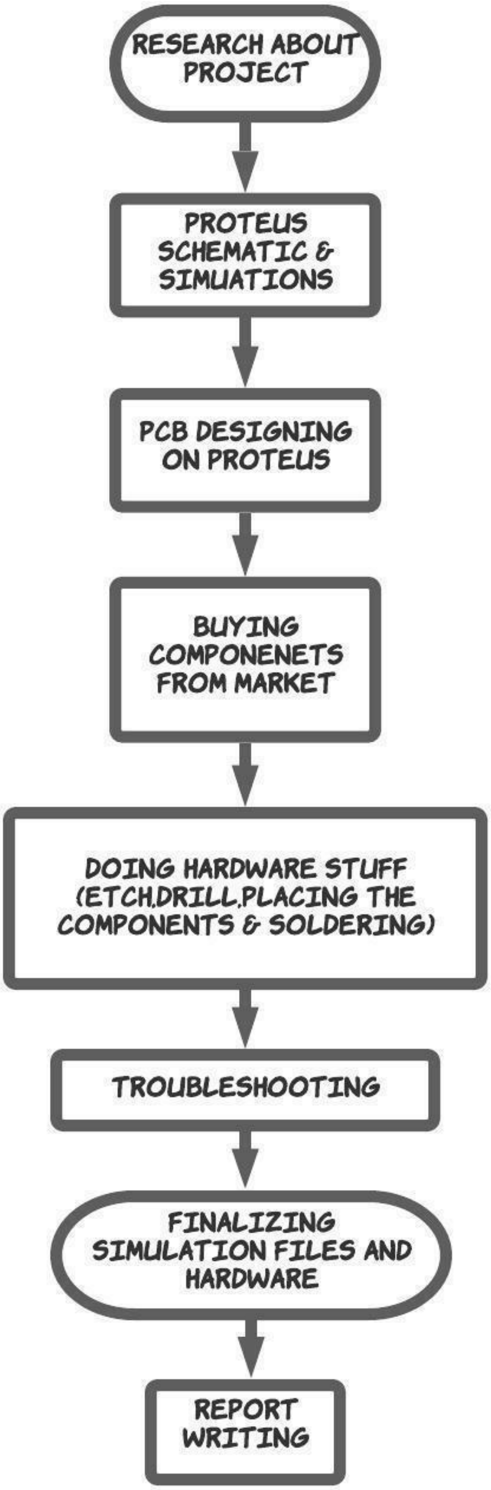
Arduino UNO & Cable:

This will be specifically used for the burn boot loading of new IC Atmega328P along with its Data cable.

Figure 15: Arduino UNO & Cable

Work Breakdown:

The plan which is followed is shown here in the form of flow chart.



Above flow chart explains the processes involved to make project from zero to working hardware. Here are steps explaining above mention processes:

1. First step was to understand project requirements and brainstorm and do research about the project. We did the rough work then.
2. As we have a rough work done, it’s time to go through software work. First, we made schematic diagram on proteus placing components and wiring them, simulating, and then designing the PCB layout and made a single layer PCB in software.
3. After software, it’s time for hardware which involves etching, drilling, placing components and soldering them.
4. Boot load run in hardware and troubleshoot if there is problem.
5. Final step is to make a report explaining every aspect of project.

Software Designing:

Schematic Circuitry:

After researching online for circuit according to requirements of project, we have designed schematic circuit having following parts and components.

First, we have a power supply circuit which provides our Arduino328p chip constant 5Volts. In it we used 7805 IC which converts any in put into 5V. we used some capacitors at both sides of 7805 which are here to stabilize and fluctuations in input voltage and we have a constant 5V at the right end. LED is placed to verify 5V output.

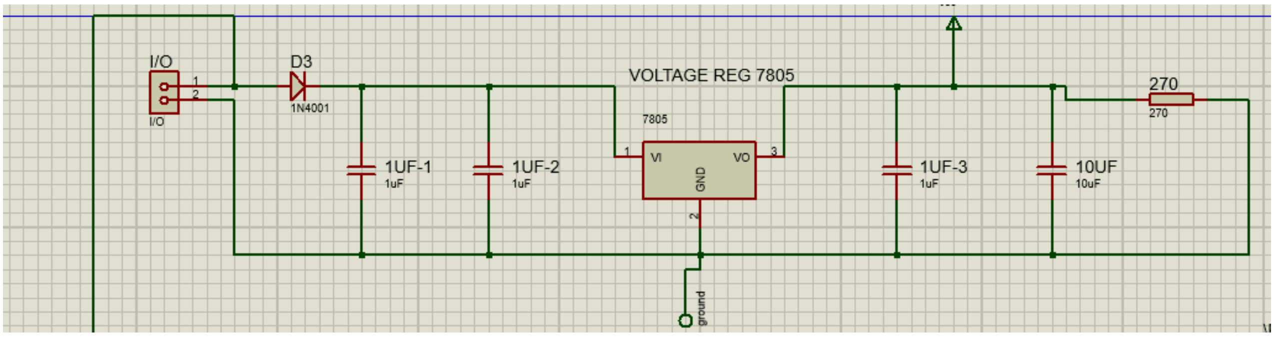


Figure 16: 5V Power supply

Then search components Arduino328p and place it in schematic. Place Sil-headers for PINB, PINC and PIND. At pin19 connect LED which will indicate code burning and can also be used for blinking. At pin20 we applied 5Vsupply. At pin21 we connected reset button to reset code if needed. At bottom right corner, we have oscillator or crystal of 16MHz with two capacitors connected (in series to each other and) parallel to crystal. They help and provide crystal constant power supply when there are voltage fluctuations.

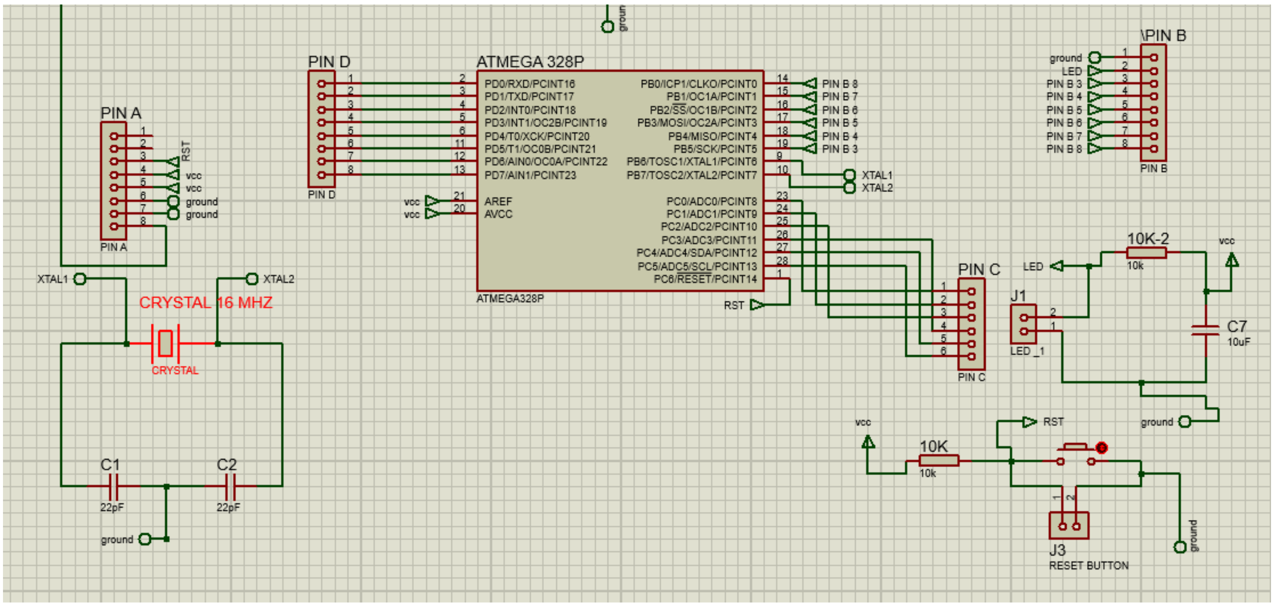


Figure 17: Sil-Headers connections to 328P pins

Circuit below is our final and complete schematic diagram.

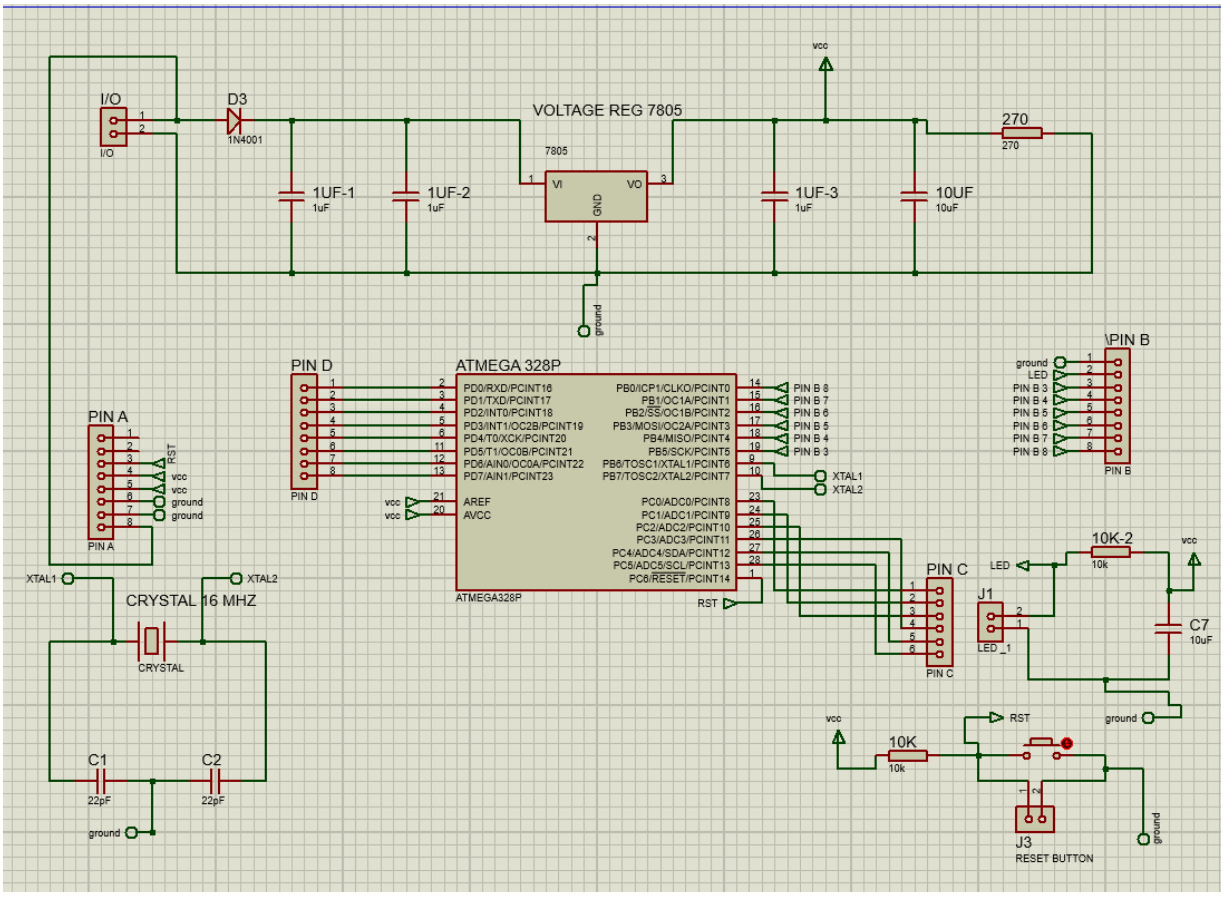


Figure 18: Complete Schematic

PCB Designing:

To convert a circuit in schematic of proteus to real life product, we design PCB layout to later print, etch etc. and have a working hardware. First, we have a blank sheet on which we place components given one by one on left side shownon left side of layout like following:

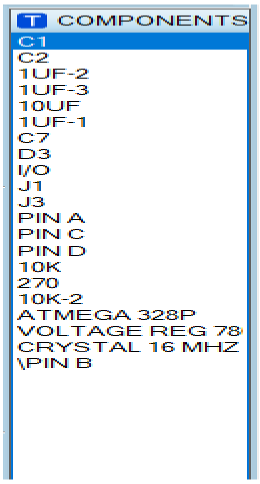


Figure 19: Components Placed

After placing the components, we started connecting them to each other as directed by proteus. As every connection has been done, we rearrange them to get the best possible thing in hardware later. Now set the path thickness connecting components and apply T40 but in some cases due to complexity we apply T30 or T15.

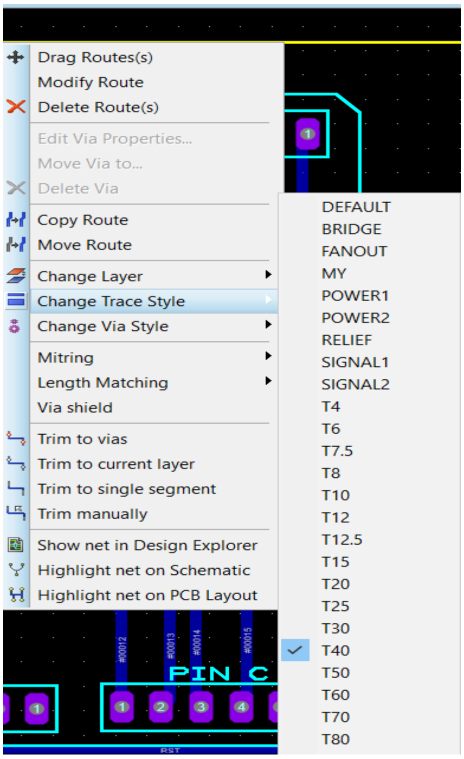


Figure 20: Path thickness

After that make sure there is no jumpers involved in whole PCB layout. So, we have following layout with name mentioned of student at corner.

Now check the 3DViewer to get the idea how hardware should be:

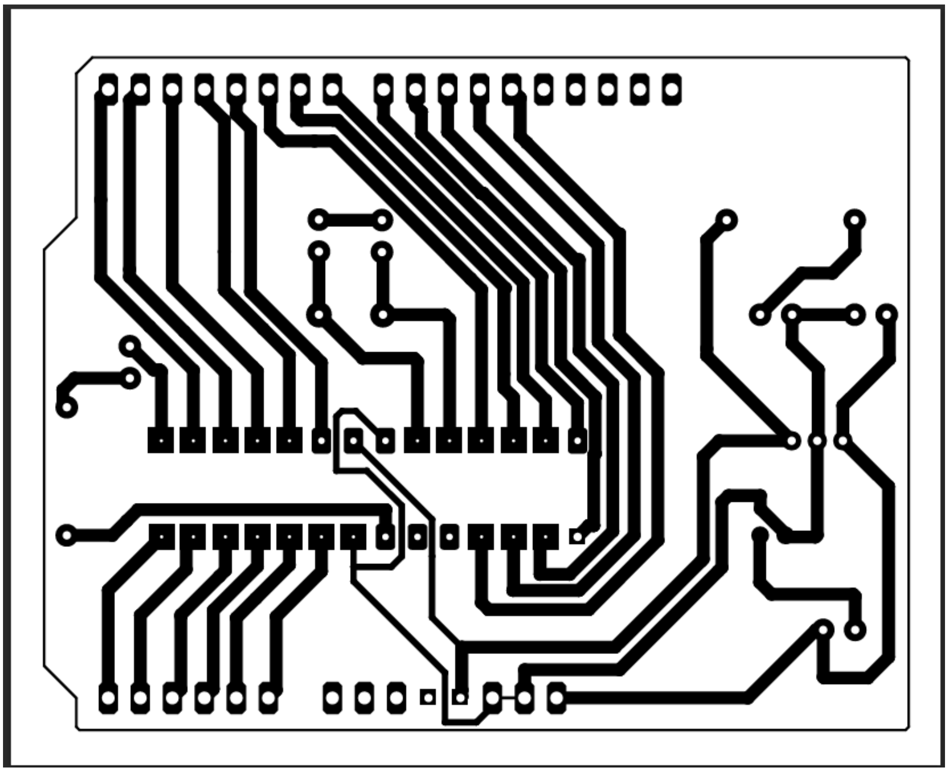


Figure 22: 3D visualizer Etched PCB

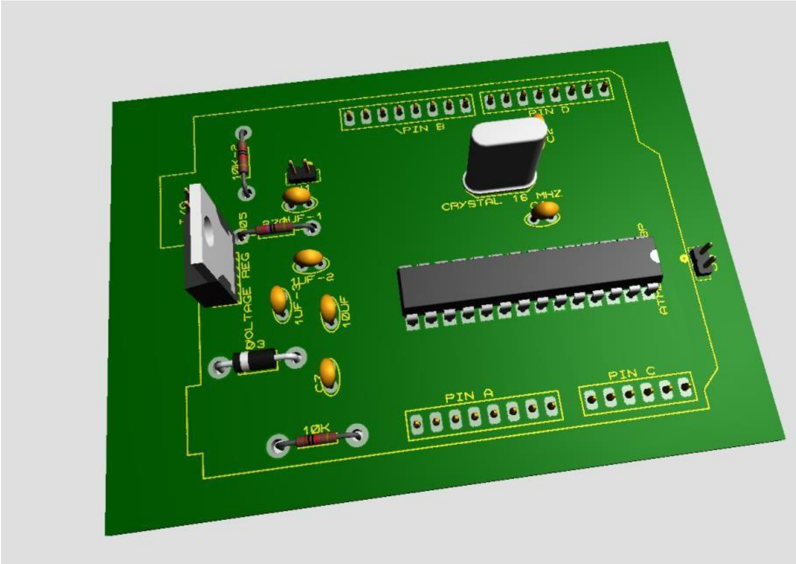


Figure 23: 3D visualizer with components

Hardware Implementation:

Bootloader burning:

For burning the bootloader image in the microcontroller, the following steps will be followed:

We require a blank Atmega328P-PU microcontroller with following components:

1. 16Mhz crystal
2. 22pF Ceramic Capacitors (2)
3. Breadboard to place the components to patch the boot loader circuit
4. Arduino UNO board (having atmegea328P chip) to use as main bootloader burning board
5. In the Arduino IDE examples menu, we can see an example sketch called “Arduino as ISP”.

If we upload the bootloader code to the required chip, it will basically act as an AVR programmer.

And set-up Arduino as an ISP.

Now here we will start the procedure to program the bootloader:

- Connect Arduino UNO to computer through usb
- Open the Arduino IDE
- Now go to menu and Open → Examples → Select” Arduino ISP”

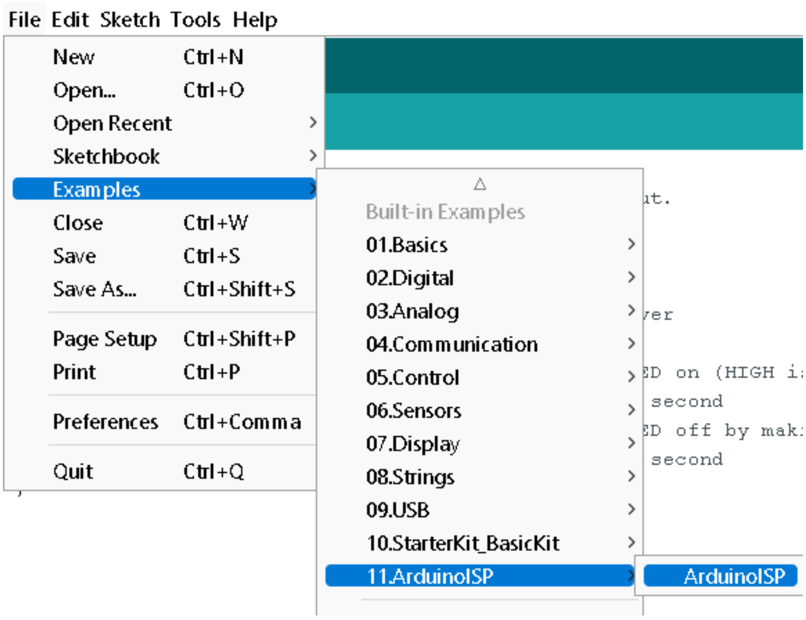


Figure 24: Selecting ArduinoISP

- First compile the shown example and then upload the Sketch on to the board (When finished, close the IDE, and disconnect your Arduino)

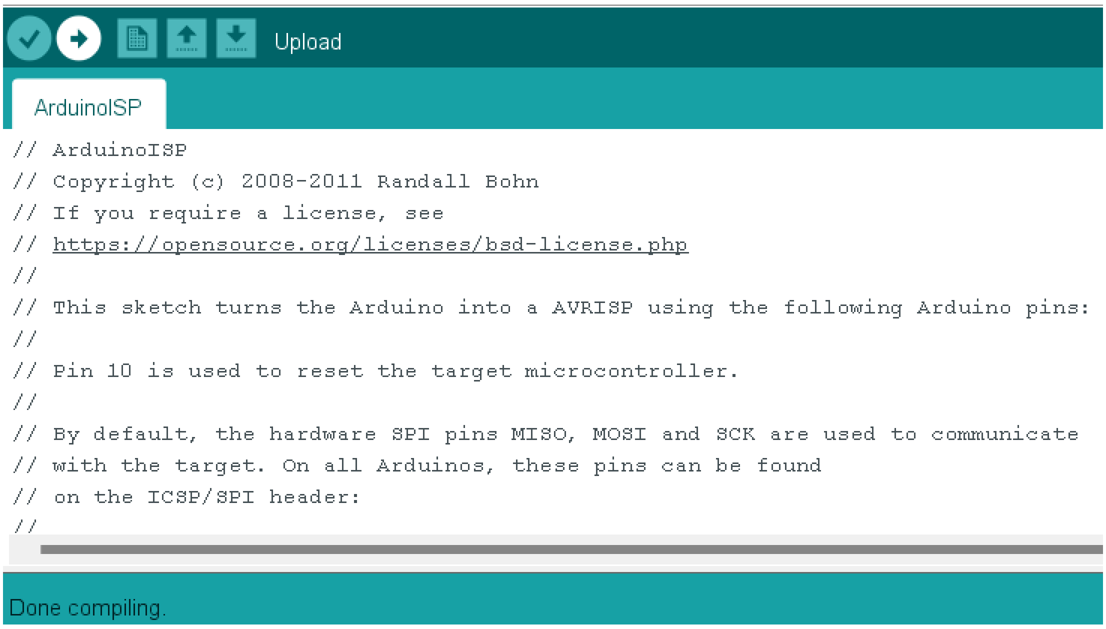


Figure 25: Compiling example

- Once the code uploading is complete, go to Tools→Programmer→Arduino as ISP.

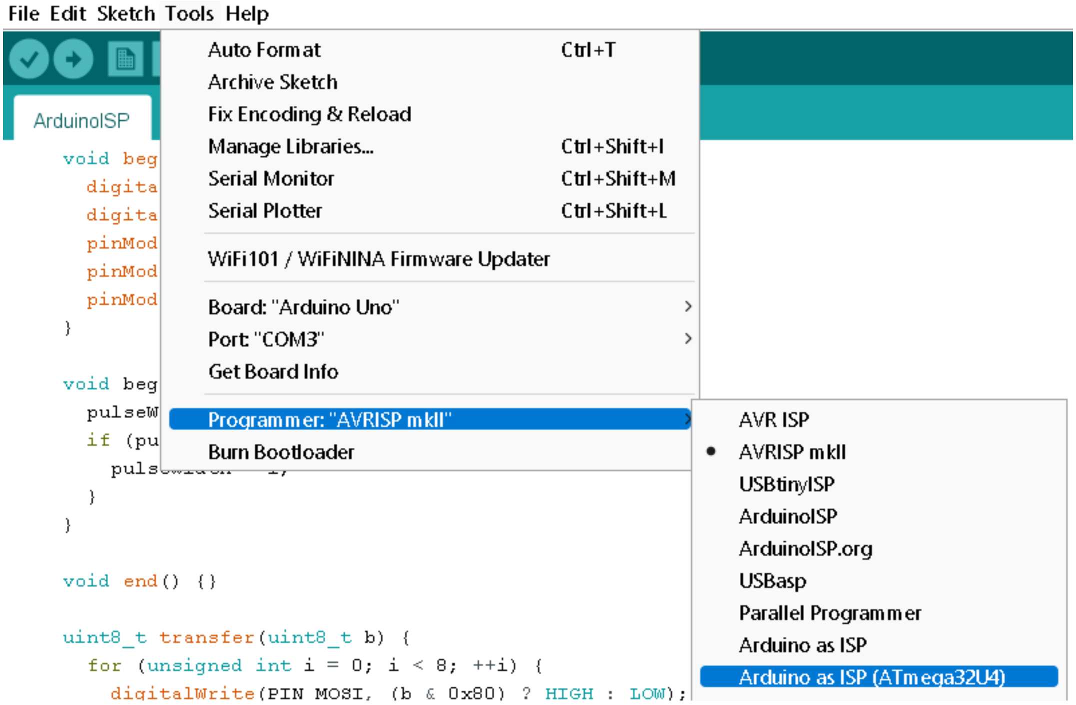


Figure 26: Selecting programmer

- Then click Burn Bootloader from tools and wait for a few minutes until it gives notification of completion.

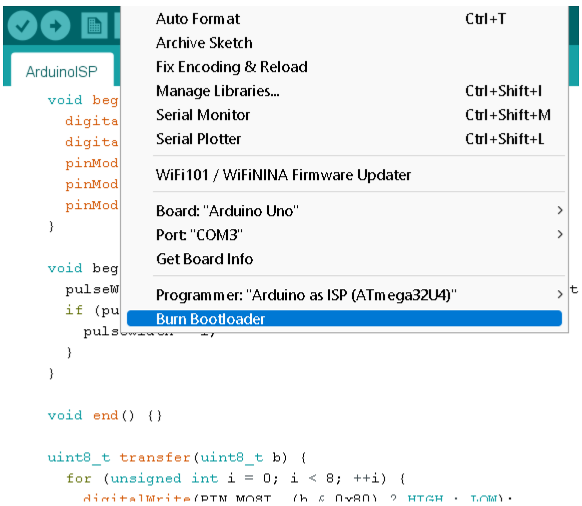
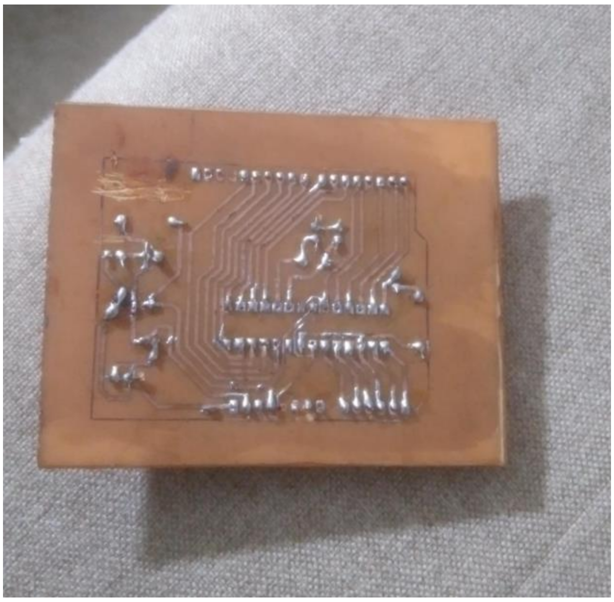


Figure 27: Burn Bootloader

IMPLEMENTATION ON PCB:

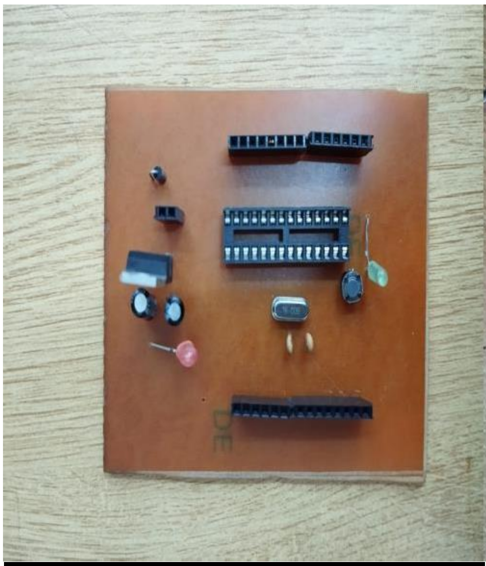
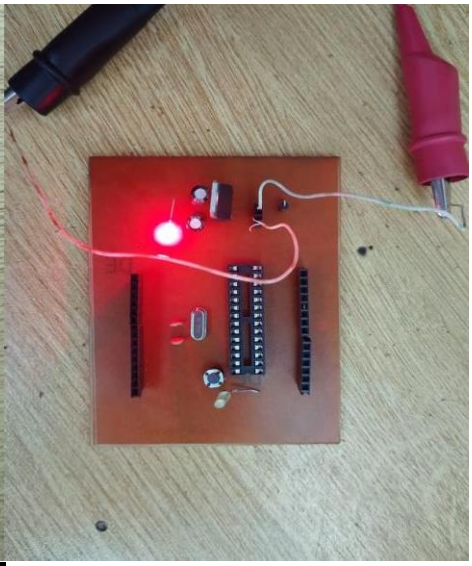
1. For PCB we must print final PCB layout on shinny side of sticker paper.
2. Make all the writing thing mirrored to that they may come straight upon etching.
3. Select the output color to be black from the PCB design.
4. Trim the copper board according to the size of the layout using a hacksaw or a cutter.
5. Next, scrub the copper side of the PCB using abrasive sponge to remove the top oxidelayer it also helps the image from the paper to stick better.
6. Now transfer the printed image on the sticker paper using hot iron to transfer the tonnerfrom paper onto the board, for this step align the board edge properly with the sticker sheet to that the print will be straight.
7. Use tape to secure the sticker paper in place and also place a extra sheet of paper and cloth on top of the circuit so that the heat from the iron can spread evenly on the print.

8. Using pliers to hold one end and keep it steady. Then put the hot iron on the other endfor about 10 seconds. Now, iron the sticker paper all along using the tip while applyinga little pressure for about 5 to 10 mins.
9. The heat from the iron transfers the tonner on the sticker paper to shift on the copper plate.
10. After ironing, place the printed plate in warm water. To dissolve the paper, then removethe paper gently. Remove the paper by peeling it from a low angle.
11. Take a plastic box and fill it up with warm water.
12. Dissolve 3-4 teaspoons of ferric chloride power in the water.
13. Place the PCB board into the etching solution for 15-30 mins.
14. The FeCl3 reacts with the unmasked copper and removes that copper from the PCB board, this process is called Etching. Use pliers to take out the PCB and check if the entire unmasked area has been etched or not. In case it is not etched, leave it in the solution for some more time.
15. Now, after the etching has been done remove the tonner from the surface of the track left by using some thinner or scrub it using abrasive sponge.
16. Now, drill holes using a PCB drill and solder all your cool components to the board.



Final Testing:

In the final testing phase, we connected the PCB board to source of power which turned on thepower supply part of the circuit which will provide 5 volts to the atmegea chip for its proper working, and then we placed the chip on to the bord as the chip was already boot loaded we used the same circuit configuration to upload a blink code on to the chip so that we can ensureits proper working, so when this chip is placed on the circuit the red led starts blinking like a heart beat it is also called heartbeat test for Arduino board .

LEARNING OUTCOMES:

The learning outcome of this project includes:

1. The use of microcontrollers to control the functions of embedded systems integrated in office, robots, home appliances, motor vehicles, and several other daily life machinery and objects.
2. How to deal with real life problems and resolve them through concepts of embedded systems using microcontrollers.
3. Enhance our hardware as well as software skills to frequent usage and manufacturing.
4. Through it resolving Industrial solutions and creating devices and process to control other devices will enhance our skills.

Tips & Precautions:

To get better results and avoid some serious failures, here are some tips and precautions you should follow:

1. Before starting project, do a good research about it. Visit different webs and have a thorough study and complete understanding about project.
2. In schematic portion of proteus, rename the essential components as it’ll help in PCB portion.

Before moving to PCB, check your connections in schematic it'll remove any error from start

1. In PCB, first place all components w.r.t schematic mean power supply components at one side, Arduino IC, and Sil-headers at one side and place crystal and reset button as close to Arduino IC as possible. Review components placement.
2. Rather than auto-routing, do wiring manually as you will understand your work and later modification would be easy.
3. Try to have jumpers close to zero if you are making single layer PCB in hardware.
4. Try to have tracks thickness equal to T40, T30 or T25.
5. Try shrinking size of PCB to a satisfactory size.
6. Apply Square through-hole pads because Round through-hole pads causes problems in soldering as they are vanished mostly while drilling holes.
7. Open 3D Visualizer to observe how would PCB look in hardware and study it thoroughly and do any modification needed.
8. In hardware, follow instructions in "**IMPLEMENTATION ON PCB**" (page 24).
9. Protect your hands, while dipping PCB in $FeCl_3$ as it can cause skin damage when come direct contact to skin.
10. Be careful while drilling, wear eye protection, do check whether drill bit is wobbling or not.
11. Be careful while cutting off extra PCB and wear protective gears necessarily.
12. While placing components, see 3D Visualizer and Schematic Capture for perfect placement of components.
13. While soldering doesn't overheat and component as it'll burn it and effect the board. Don't make cold solder connections.
14. Before experimenting in hardware, check every connection if components are connected right and apply power and ground at right points.
15. Troubleshoot if circuit is not working. Do it until you succeed.

These small tips and precautions will help you through any experiment both in hardware and software.

CONCLUSION:

The overall goal was to create a line-following robot utilizing our very own generic Arduino board. The purpose of the proposition is to supply fundamental guidelines to fresh understudies and fledglings who are interested in this type of business. As previously stated, the venture familiarizes new understudies with proteus and how to construct PCB, its operating component, and future sections of it in an easy and reasonable approach. On our way to the last element of this assignment, we faced a slew of unexpected scenarios and learned a slew of new things. Our knowledge of microcontrollers and printed circuit boards grew. Our expertise in circuit and PCB design were utilized in every replica and piece of equipment. The venture involved troubleshooting and collaboration with Arduino UNO, resulting in a successful task. The goal was to engage students in designing, mechatronics, and programming by constructing, building, and programming an independent robot. The easy-to-use rules made it easy for new students to build a foundation in their learning.