TABLE OF CONTENTS

ABSTRACT

Bootloader programming is a crucial step in microcontroller development, enabling ICs to be ready for code uploading through simplified interfaces. This project presents a Universal Bootloader Burner Circuit Board specifically designed for DIP-based microcontrollers like ATmega328P, ATtiny85, ATtiny84, and ATtiny2313A. The aim is to eliminate the need for complex jumper wire setups by offering a compact, multi-IC compatible socket platform. It allows direct bootloader burning using an Arduino UNO as ISP, ensuring ease of access, time-saving, and error-free connections. The board includes all essential components such as voltage references, capacitors, and crystals, ensuring each IC is ready for programming.

This project is highly beneficial in embedded systems laboratories, academic projects, and prototype development environments. It streamlines the bootloading process, making it more efficient and user-friendly for developers and students alike.

INTRODUCTION

In embedded systems development, **bootloader programming** is the first essential step in configuring a microcontroller to receive user-defined programs. This process allows microcontrollers to be programmed over interfaces like USB or UART after a bootloader is installed.

Traditionally, burning a bootloader requires **manual wiring** for each individual microcontroller, which is **time-consuming**, prone to **connection errors**, and inconvenient when working with multiple ICs.

To address these challenges, this project introduces a **Universal Bootloader Burner Circuit** designed specifically for **DIP (Dual In-line Package) ICs**. The custom-designed PCB integrates **multiple IC sockets**, pre-routed signal lines, and the necessary passive components like **crystals and capacitors**. It allows for **fast and error-free bootloading** using an **Arduino UNO programmed as ISP (In-System Programmer)**.

This universal solution improves workflow in **educational laboratories**, **project development**, and **prototype testing**, offering a reliable and efficient method for preparing multiple microcontrollers for firmware uploading.

OBJECTIVE OF THE PROJECT

The main objectives of the Universal Bootloader Burner Circuit for DIP ICs are:

- To design a compact and reusable PCB that enables easy bootloader burning for various DIP microcontrollers.
- To eliminate the need for repetitive and error-prone manual wiring, which is usually required when working with different ICs.
- To support multiple popular microcontroller ICs, including:
  - ATmega328P
  - ATtiny84
  - ATtiny85
  - ATtiny2313A
- To simplify the bootloader burning process by using an Arduino UNO as ISP, enabling a standard and accessible programming setup.
- To create a user-friendly, time-efficient, and accurate tool for students, educators, and developers involved in embedded system projects.

COMPONENTS USED

The following components were used in the design and development of the

Microcontroller ICs (DIP Packages)

- ATmega328P-PU
- ATtiny85
- ATtiny84
- ATtiny2313A

IC Sockets

- 28-pin DIP IC Bed – for ATmega328P
- 20-pin DIP IC Bed – for ATtiny84, ATtiny2313A
- 8-pin DIP IC Bed – for ATtiny85

Passive Components

- **Crystals:**
  - 16 MHz (for ATmega328P)
  - 8 MHz (for ATtiny series)
- **Capacitors:**
  - 22pF (used with crystals)
  - 10µF Electrolytic Capacitor
- **Resistors:**
  - 1kΩ
  - 10kΩ

Indicators

- **LEDs:**
  - Power LED
  - Programming Status LED
  - Error LED

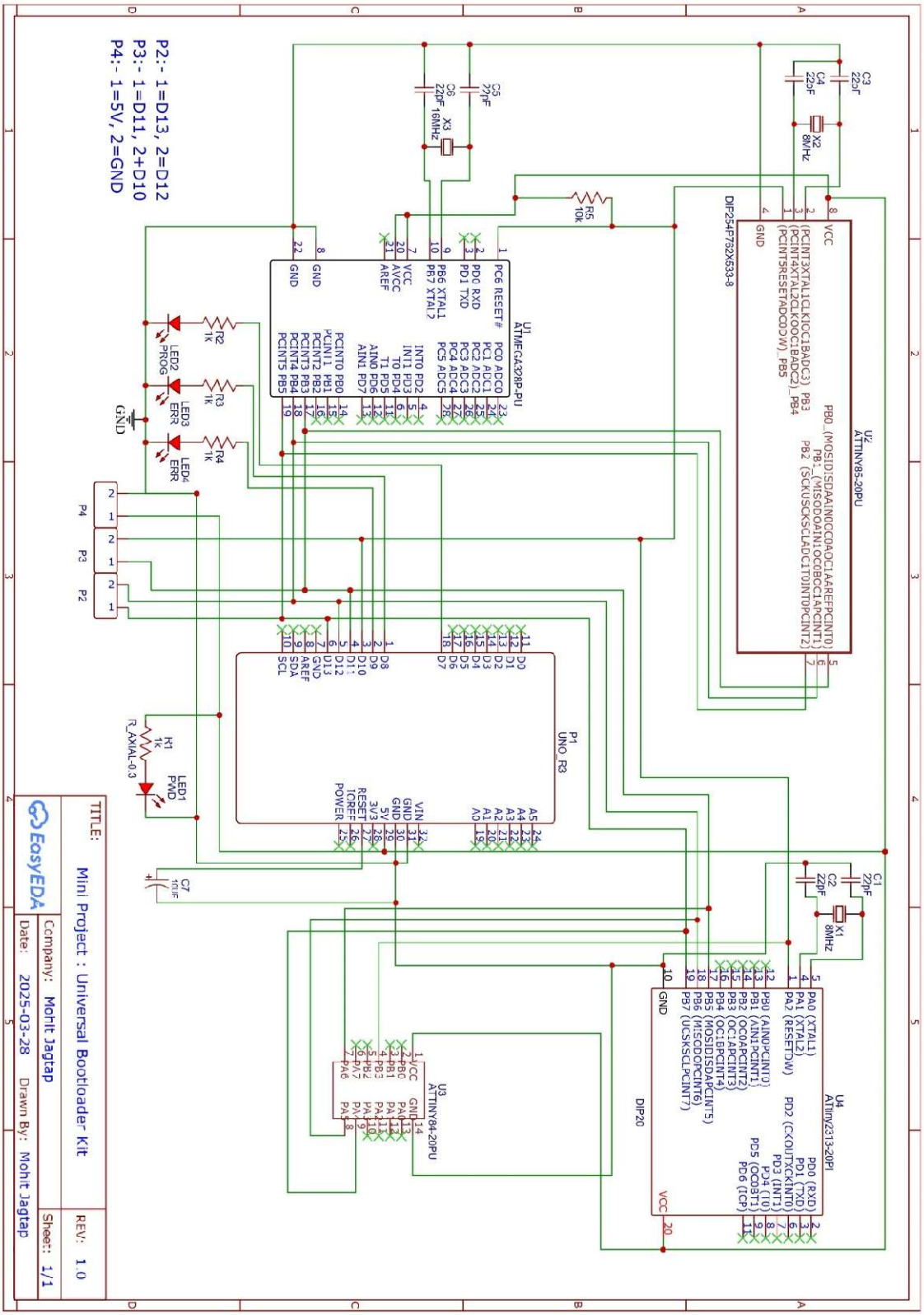
Controller and Interface

- **Arduino UNO** (used as ISP programmer)
- **6-pin ISP Header** (for connection with Arduino)

Custom PCB

- Multi-socket custom-designed printed circuit board (fabricated)

SCHEMATIC


P2:- 1=D13, 2=D12
P3:- 1=D11, 2+D10
P4:- 1=5V, 2=GND
TITLE:
Mini Project : Universal Bootloader Kit
Company: Mohit Jagtap
Date: 2025-03-28
Drawn By: Mohit Jagtap
Sheet: 1/1
REV: 1.0
EasyEDA
U1 ATMEGA328P-PU
U2 ATTINY85-20PU
U3 ATTINY84-20PU
U4 ATtiny2313-20PI
P1 UNO_R3
DIP20
DIP14
DIP8
LED1 PWD
LED2 PROG
LED3 ERR
LED4 ERR
GND

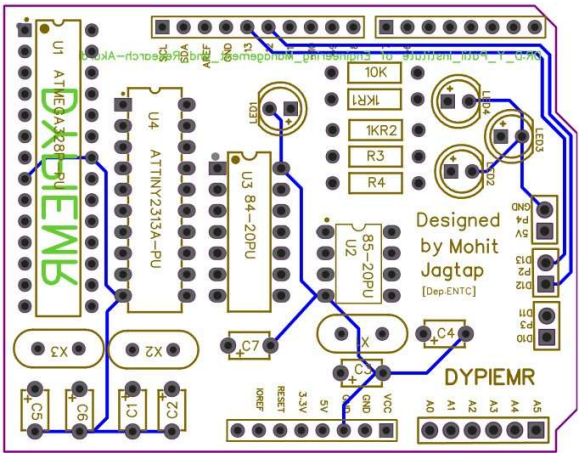
CIRCUIT DIAGRAM AND EXPLANATION

The circuit diagram represents the complete wiring and component connections for the Universal Bootloader Burner Circuit. It includes the connections between the Arduino UNO (used as ISP) and the multiple IC sockets on the custom PCB.

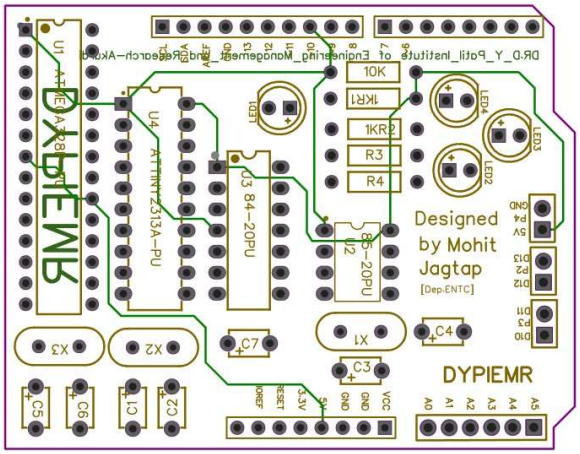
Circuit Explanation:

- **Arduino UNO ISP Pins:**
  - MISO (D12), MOSI (D11), SCK (D13), RESET (D10) – These are connected to all IC sockets using proper routing.
  - VCC and GND – Common power supply lines to all IC sockets.
- **IC Sockets:**
  - Each IC socket is wired based on the respective microcontroller's pin configuration.
  - Tracks are isolated to avoid crosstalk or programming errors.
- **Crystals and Capacitors:**
  - A 16 MHz crystal with 22pF capacitors is used for ATmega328P.
  - 8 MHz crystals are used for ATtiny ICs requiring external clocks.
  - These ensure stable clock signals during bootloader burning.
- **Status LEDs:**
  - Power LED glows when the board is powered.
  - Programming LED blinks during active bootloader burning.
  - Error LED lights up in case of a programming fault or incorrect connection.
- 10µF Capacitor between RESET and GND on Arduino helps prevent unwanted resets during programming.

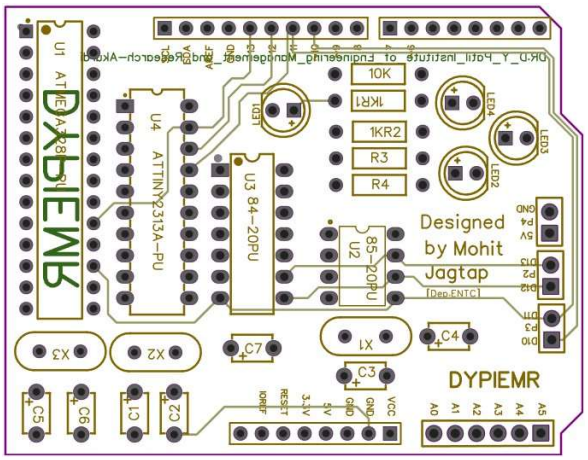
PCB DESIGN AND LAYERS



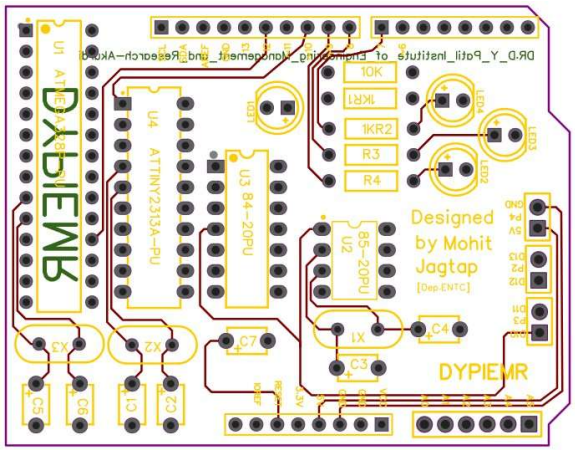
Bottom Layer



Layer 2



Layer 3



Top Layer

WORKING OF THE CIRCUIT

Steps to Operate the Circuit kit :

1. IC Insertion:
    - Insert the target IC (e.g., ATmega328P, ATtiny85, ATtiny84, or ATtiny2313A) into the appropriate DIP socket on the PCB. Ensure the IC is placed correctly, with the notch aligned to the marked direction.
2. Connecting to Arduino:
    - Connect the Arduino UNO to the PCB via the 6-pin ISP header. The connections from the Arduino to the IC include MOSI, MISO, SCK, RESET, VCC, and GND.
3. ArduinoISP Sketch Upload:
    - Upload the "ArduinoISP" sketch to the Arduino using the Arduino IDE. This prepares the Arduino to function as an ISP, which will communicate with the target IC for bootloader programming.
4. Burning the Bootloader:
    - Once the ArduinoISP sketch is uploaded, select the target board and the "Burn Bootloader" option in the Arduino IDE.
    - The Arduino sends the necessary signals to the IC via the ISP pins to burn the bootloader into the microcontroller’s memory.
5. Status Indication:
    - The Power LED will light up to indicate that the circuit is powered.
    - The Programming LED will blink during the burning process.
    - If there are any errors, such as connection issues, the Error LED will light up to indicate a fault.
6. Bootloader Completion:
    - Once the bootloader is successfully burned, the IC can now be programmed using the USB-to-Serial interface for uploading the final user code.

COST ANALYSIS

Component	Quantity	Cost per Unit (INR)	Total Cost (INR)
ATmega328P-PU IC	1	250	250
ATtiny85 IC	1	100	100
ATtiny84 IC	1	120	120
ATtiny2313A IC	1	150	150
28-pin DIP Socket	1	30	30
20-pin DIP Socket	2	25	50
8-pin DIP Socket	1	20	20
16 MHz Crystal	1	15	15
8 MHz Crystal	2	10	20
Capacitors (22pF)	2	5	10
Capacitors (10µF)	1	10	10
Resistors (1kΩ)	2	2	4
Resistors (10kΩ)	2	2	4
LEDs (Power, Programming, Error)	3	5	15
Arduino UNO	1	400	400
PCB Fabrication	1	500	500

Total Cost for Project Development: ₹ 1628

TOOLS AND SOFTWARE USED

The development of the Universal Bootloader Burner Circuit involved several tools and software:

1. Altium Designer

- Purpose: Schematic design and PCB layout.
- Features: Multi-layer support, advanced routing, real-time design rule checking.

---

2. Arduino IDE

- Purpose: Programming Arduino UNO for bootloader burning.
- Features: Uploads the ArduinoISP sketch and communicates with target ICs via ISP.

---

3. Altium CircuitMaker

- Purpose: Circuit simulation and testing.
- Features: Cloud-based, real-time simulation, integrates with Altium Designer.

---

4. Soldering Tools

- Purpose: Assemble and solder components onto the PCB.
- Tools: Soldering iron, solder wire, desoldering pump.

---

5. Multimeter

- Purpose: Testing voltages, continuity, and signal integrity.

---

6. PCB Fabrication Services

- Purpose: Manufacture the custom PCB based on Altium design files.

ADVANTAGES

1. Time-Saving
   - The custom-designed PCB eliminates the need for manually wiring each microcontroller, reducing the time spent on setup and bootloader burning.

2. Error Reduction
   - The PCB design ensures correct and consistent connections, minimizing the risk of human error during the bootloader burning process.

3. Multi-IC Support
   - The circuit is compatible with multiple microcontrollers like ATmega328P, ATtiny85, ATtiny84, and ATtiny2313A, offering flexibility for various projects.

4. Reusable Design
   - The PCB can be reused for multiple bootloading sessions, making it cost-effective for educational and prototype development environments.

5. Compact and Easy to Use
   - The design is compact, easy to assemble, and doesn't require complex setups, making it ideal for students and small-scale developers.

APPLICATIONS

1. Embedded Systems Laboratories

Ideal for educational labs, allowing students to quickly burn bootloaders into microcontrollers without the need for manual wiring setups.

2. Educational and Academic Projects

Useful in student projects, simplifying the bootloader burning process and focusing on the application development rather than setup.

3. Prototype Development and Testing

Helps in the quick setup of bootloaders for rapid prototyping, especially in product development cycles that require frequent updates to microcontroller firmware.

4. Small Scale Production Lines

Can be used in small-scale production environments where multiple microcontrollers need to be programmed quickly and efficiently.

LIMITATIONS

1. Limited to DIP ICs

The circuit only supports DIP (Dual Inline Package) microcontrollers, limiting its compatibility with surface-mount devices (SMDs) or other package types.

2. Manual IC Placement

ICs must be manually inserted into their respective sockets, which may lead to errors or damage if not done carefully.

3. Dependence on Arduino as ISP

The circuit relies on Arduino UNO as an ISP (In-System Programmer), meaning any issues with the Arduino or its programming can disrupt the bootloading process.

4. No Built-In Power Supply

The circuit does not include a dedicated power supply for the microcontrollers, requiring an external power source to operate.

FUTURE SCOPE

The Universal Bootloader Burner Circuit can be enhanced and expanded in the future to increase its functionality and versatility:

1. Support for More IC Types

- **Future Enhancement:** Add support for **PIC**, **STM32**, and other microcontroller families to make the circuit more versatile and useful for a broader range of applications.

2. USB-C Port for Power and Programming

- **Future Enhancement:** Integrating a **USB-C** port for both power and direct serial programming would simplify the design and improve its usability.

3. Compact SMD Version

- **Future Enhancement:** A **Surface-Mount Device (SMD)** version could be developed for industrial use, making the design more compact and suitable for mass production environments.

4. Onboard Microcontroller for Auto Bootloading

- **Future Enhancement:** Including an **onboard microcontroller** could automate the bootloader burning process, eliminating the need for an external Arduino as ISP.

LITERATURE SURVEY

Bootloading is essential for preparing microcontrollers for programming. Traditionally, it requires manual wiring, which is time-consuming and error-prone. To address this, various methods and references were studied during the project's initial phase.

1. Arduino ISP Method

The **Arduino UNO** can be used as an **In-System Programmer (ISP)** by uploading the "ArduinoISP" sketch. This method allows a standard Arduino board to program the bootloader onto other microcontrollers. This approach is widely accepted in educational setups due to its low cost and easy availability of resources. **Reference:** Arduino Official Documentation (https://www.arduino.cc/).

2. Microcontroller Datasheets

The datasheets for **ATmega328P**, **ATtiny85**, **ATtiny84**, and **ATtiny2313A** were thoroughly studied to understand the pin configuration, required clock settings, bootloader requirements. This helped in designing the universal circuit that supports multiple IC types.

3. Existing Bootloader Programming Setups

Several online DIY setups use breadboards for bootloader burning. While these setups work, they require manual wiring for each IC, increasing the chances of connection mistakes. Learning from these designs, this project aimed to create a **permanent PCB solution** that eliminates repetitive work and ensures reliable, error-free bootloading.

4. Use of Custom PCBs in Development

Fabricating a dedicated PCB helps in prototyping, repeated use, and reduces human error significantly. Using design software like **Altium**, a professional PCB layout was created ensuring proper clearances, track widths, and standard ISP connectivity.

5. Academic and Industrial Practices

In academic laboratorie, the need for quick and reliable microcontroller preparation is crucial. Existing industry practices use dedicated programming boards for specific microcontrollers. This survey motivated the idea of designing a **universal, multi-IC compatible board** for academic and hobbyist usage.

CONCLUSION

The Universal Bootloader Burner Circuit for DIP ICs provides a practical and efficient solution for embedded system developers. By eliminating the need for manual wiring and reducing human error, this circuit simplifies the process of bootloader burning for a variety of microcontrollers, including ATmega328P, ATtiny85, ATtiny84, and ATtiny2313A. The circuit's compact, reusable design, along with its easy setup, makes it ideal for educational use, prototype development, and small-scale production.

This project not only contributes to enhancing the workflow in embedded systems laboratories but also serves as a cost-effective solution for hobbyists and developers. The ability to reprogram microcontrollers quickly and efficiently aids in the rapid development and testing of firmware, ultimately accelerating the development cycle in various applications.

With its current design and potential for future enhancements, this circuit is poised to be an invaluable tool in the field of embedded systems development.