EMBEDDED SYSTEM PROJECT[EE3401]

Electrical Engineering Project submitted to the

National Institute of Technology Rourkela

In partial fulfillment of ***Bachelor in technology in Electrical Engineering***

By

Praveen Kumar Yadav – 122EE0385

Rajashree Behera – 122EE0145

Aiman Tariq – 122EE0171

Under the supervision of

Dr. Supratim Gupta

Department of Electrical Engineering

National Institute of Technology Rourkela

Table of Contents

1. Introduction

Problem Statement

Imagine a circuit that needs more than just a single voltage. Your phone charger only gives 5V, but your electronics need 2.2V or 3.3V. A challenge, right? This is exactly the situation in many modern devices. They demand precise voltages to function, yet often have only one power source.

Project Scope and Objectives

This project aims to design an embedded system that can take a single 5V input and generate adjustable 2.2V and 3.3V outputs. We will use an LM317 voltage regulator and a microcontroller to achieve this. It won't just convert voltage—it'll allow you to switch between levels easily. Two buttons will let the user select which voltage they need, and the circuit will provide a stable, regulated output.

Main Goals:

- Create a stable, reliable voltage converter
- Make it user-friendly with simple switch controls
- Ensure the output voltage remains steady
- Keep the design cost-effective

2. Background and Requirements

Customer's Problem Statement

Client need multiple voltage levels from a single 5V source. In their electronic accessory circuits, some parts need 2.2V while others need 3.3V. But how to provide that when you only have one input voltage? This project solves that.

Voltage Regulation in Electronic Circuits

Voltage regulation is like a safety valve in a water pipe. If too much voltage flows through a circuit, things can get messy. Components can burn out. Outputs can become unstable. That's where regulators like the LM317 come in. They take excess voltage and reduce it to the level you need. They keep everything steady.

Requirement Analysis

Primary Requirements:

- Convert 5V to both 3.3V and 2.2V
- Switch between voltages easily
- Maintain stable output
- Protect against voltage fluctuations

Technical Specs:

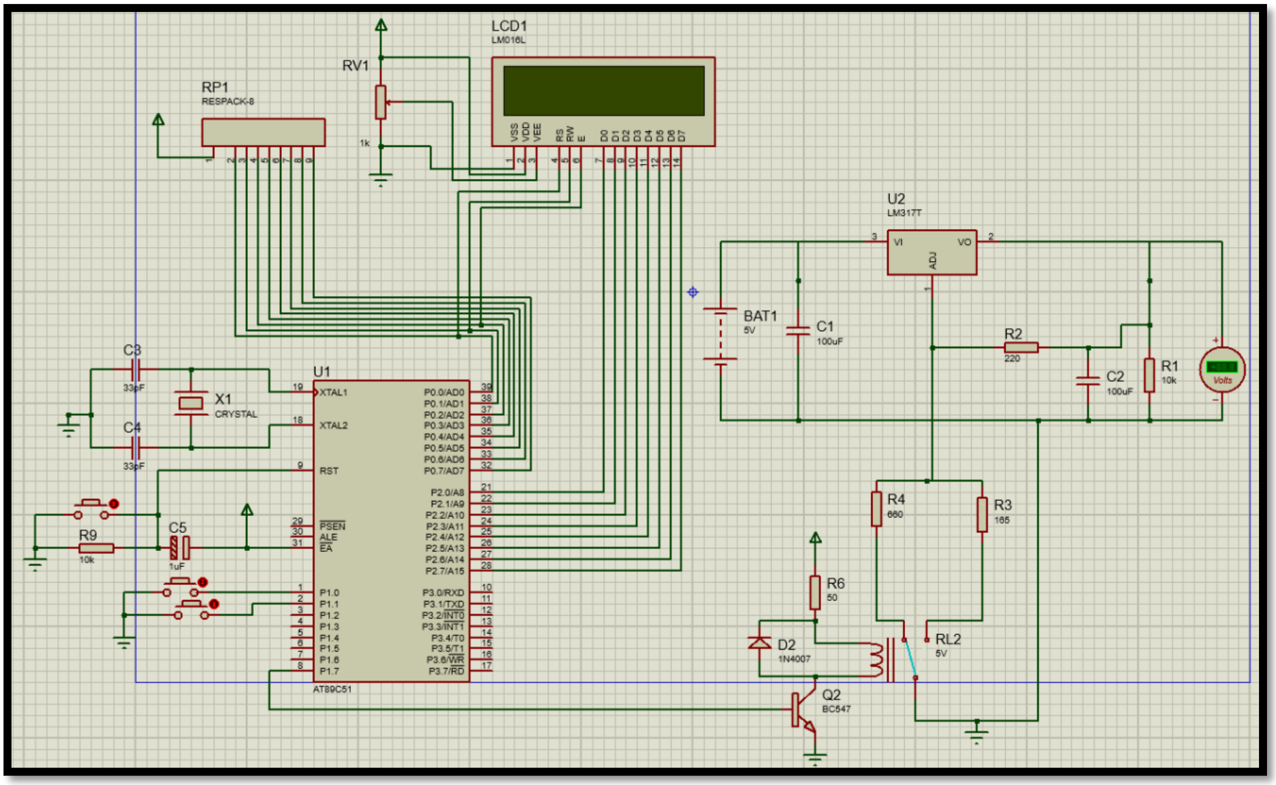
- Input voltage: 5V DC
- Output options: 3.3V or 2.2V
- Control method: Manual switches
- Protection: Short circuit and overload

3. Component Used

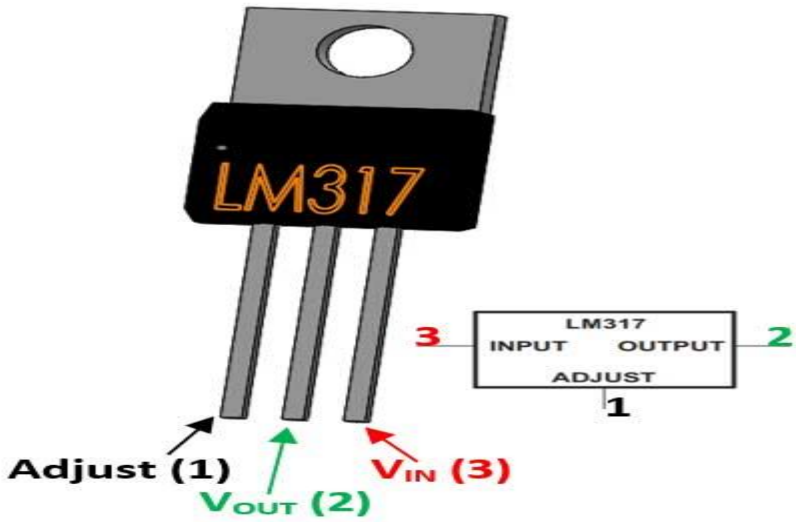
- AT89C51 Microcontroller (or any 8051 based Microcontroller)
- 8051 Programmer (Programming Board)
- 11.0592 MHz Quartz Crystal
- 5V Battery
- LM317T (Voltage Regulator)
- BC547 Silicon NPN Low Power Bipolar Transistor
- 1N4007 Diode
- 5V Relay
- 2 x 33pF Capacitor
- 2 x 100µF Capacitor
- 165Ω Resistor
- 660Ω Resistor
- 2 x 10kΩ Resistors
- 220Ω Resistor
- 2 x 50 Ω Resistors
- Push Buttons
- Programming cable

- Connecting wires
- Power Supply
- Keil µVision IDE
- Proteus (for circuit diagram)

4. Circuit Diagram and Component Description

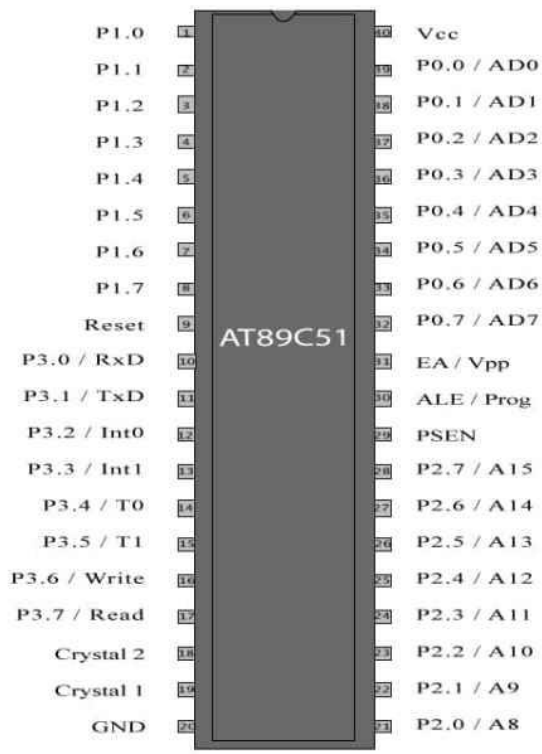


LM317 Voltage Regulator: The LM317 is an adjustable 3-terminal voltage regulator that provides an output voltage range from 1.25V to 37V. In this project, it is used to convert a 5V DC input to regulated 3.3V and 2.2V outputs, ensuring stable voltage for the accessory circuits.



Pin Diagram of LM317T Voltage Regulator

AT89C51 Microcontroller: The AT89C51 is an 8051-compatible microcontroller used to control the voltage regulation process. It features 4KB of on-chip flash memory and is based on the 80C51 instruction set. The microcontroller reads the state of switches and turn on/off the relays to provide the required output voltage, either 2.2V or 3.3V.



Pin Diagram of AT89C51 Microcontroller

16X2 LCD Display: The 16x2 Liquid Crystal Display (LCD) is a widely used alphanumeric display module that can show up to **32 characters** across **two rows**, with each row accommodating **16 characters**. It is based on the HD44780 or compatible LCD controller/driver, which simplifies interfacing with microcontrollers. Here's a detailed description of its features and functionality:

Pin Diagram of 16X2 LCD Display

Relays and NPN Transistors: 5V relay is used to switch between the two output voltages. NPN transistors (such as the BC547) are used to drive the relays, as the microcontroller pins do not provide enough current to operate the relays directly. The transistors amplify the control signal from the microcontroller to activate the relays.

Resistors and Capacitors: Resistors are used to set the appropriate output voltage for the LM317 regulator, while capacitors filter the output, ensuring voltage stability. This helps reduce voltage ripples and ensures a smooth, stable output voltage for sensitive electronic components.

5. Circuit Implementation

Detailed Circuit Explanation

The circuit consists of an input 5V power supply fed to the LM317 adjustable voltage regulator. The LM317, with external resistors, converts this 5V input into two specific output voltages: 3.3V and 2.2V. The output is controlled by an 8051 microcontroller, which uses SPST switches and relays to toggle between the two voltage levels.

Input and Output Voltage Control

Two SPST switches connected to the microcontroller's P1.0 and P1.1 pins are used to select the required voltage. When a switch is turned on, the microcontroller checks which switch is active and sends a signal to the appropriate NPN transistor. These transistors drive the relays, which select between 3.3V and 2.2V outputs. The relays are connected to the output of the LM317, ensuring smooth switching between voltages.

Role of LM317 in Voltage Regulation

The LM317 regulates the output voltage by adjusting the resistance on its adjustment pin. The chosen resistors set the voltage to either 2.2V or 3.3V, depending on the configuration. The regulator provides a stable output despite fluctuations in input voltage or load, ensuring that the circuits connected to the system receive the correct voltage without spikes or drops.

6. Microcontroller Programming

Algorithm Description: Voltage Selection and Display System

1. System Initialization

- Configure I/O Ports:
  - P2 set as output for LCD data
  - P0 configured for LCD control signals (RS, RW, EN)
  - P1 used for input buttons and relay control
- Initialize system state to 0 (no voltage selected)
- Perform LCD initialization sequence with multiple attempts and delays

2. Main Program Flow

a. Display Static Text

- Initial display shows "Output Voltage:" on the first line of the LCD

b. Continuous Input Monitoring (Main Loop)

- Constantly check input pins P1.0 and P1.1 for button presses
- Two distinct input conditions are monitored:
  - Button for 2.2V selection (P1.0)
  - Button for 3.3V selection (P1.1)

<u>3. Voltage Selection Mechanism</u>

a. 2.2V Selection

- Triggered when P1.0 is pressed
- Checks if current state is not already 2.2V
- Actions:
  - Turn off relay (CLR P1.7)
  - Clear second line of LCD
  - Display "2.2V" on second line
  - Update current state to 2.2V
  - Implement debounce delay for button stability

b. 3.3V Selection

- Triggered when P1.1 is pressed
- Checks if current state is not already 3.3V
- Actions:
  - Turn on relay (SETB P1.7)
  - Clear second line of LCD
  - Display "3.3V" on second line
  - Update current state to 3.3V
  - Implement debounce delay for button stability

<u>4. State Management</u>

- Maintain CURRENT_STATE variable to track voltage state
- Prevent unnecessary state changes and LCD updates
- States:

    - 0: No voltage selected
    - 1: 2.2V selected
    - 2: 3.3V selected

<u>5. LCD Interaction Subroutines</u>

a. LCD Initialization

- Multiple attempts to set LCD mode
- Configure display settings (2 lines, cursor behavior)
- Clear display

b. LCD Command and Data Writing

- Set control signals (RS, RW, EN) for command/data transmission
- Manage enable signal with delays for stable communication

c. String Display

- Transmit null-terminated strings to LCD
- Incrementally display characters

<u>6. Timing and Delay Management</u>

- Implement software delays for:
    - LCD initialization
    - Button debouncing
    - Stable signal transmission

Code Explanation

This 8051-microcontroller program manages a voltage selection system with an LCD display. The code initializes by configuring I/O ports and setting up the LCD display with a static "Output Voltage:" message. The main loop continuously monitors two input pins (P1.0 and P1.1) for button presses, allowing users to select between 2.2V and 3.3V output voltages.

When a button is pressed, the program checks the current state to prevent unnecessary updates. Upon selection, it triggers the corresponding relay and updates the LCD's second line to display the selected voltage. A state variable (CURRENT_STATE) tracks the current voltage setting.

The code includes several critical subroutines for LCD initialization, command transmission, and data writing, ensuring reliable communication with the display. Debounce delays are implemented to prevent multiple unintended triggers from button presses, providing a stable user interface for voltage selection.

Push Button Input Handling

The code implements a robust push-button input handling mechanism using input pins P1.0 and P1.1. These pins are continuously monitored in the main loop using conditional jump instructions (JNB). When a button is pressed, the code checks the current system state to prevent redundant actions. The input detection is designed with a state-comparison approach (CJNE instruction) to ensure voltage changes occur only when the requested state differs from the current state. Debounce protection is achieved through multiple long delay subroutines after each button press, which suppress rapid, unintended state transitions and provide mechanical switch stabilization. This approach ensures reliable and clean input interpretation without requiring hardware interrupt mechanisms.

Relay Control Logic

The relay control is managed through the P1.7 pin using 8051 assembly instructions SETB and CLR. When 2.2V is selected, the relay is turned off (CLR P1.7), while 3.3V selection activates the relay (SETB P1.7). The control logic ensures that relay state changes occur only when the current voltage differs from the requested state, preventing unnecessary switching. This approach minimizes electrical wear and potential system instability. The state transition is protected by adding multiple long delay subroutines after each relay operation, which provides mechanical stability, reduces electrical noise, and prevents relay contact chattering. The CURRENT_STATE variable tracks the current voltage selection, enabling precise and controlled relay management.

LCD Data Display Mechanism:

The LCD display management is implemented through a series of carefully crafted subroutines in the 8051 assembly code. When a voltage change is detected, the program first clears the second line of the LCD using the CLEAR_LINE subroutine, which writes 16 space characters to reset the display area. The LCD_CMD subroutine positions the cursor at the start of the second line using the LCD_LINE2 address (0C0H). Subsequently, the LCD_STRING subroutine is called to display the appropriate voltage string ("2.2V" or "3.3V") by sequentially writing characters from a predefined message in the code segment. This approach ensures clean, synchronized visual feedback that precisely reflects the current voltage selection state.

This process allows seamless switching between voltages, controlled by the microcontroller based on user input.

Assembly Code Explanation:

```
; Segment definitions
CSEG AT 0
    ORG 0000H
    LJMP MAIN
; Data segment for variables
DSEG AT 30H
CURRENT_STATE: DS 1    ; Store current voltage state (0=none, 1=2.2V, 2=3.3V)
; Code segment
CSEG
; LCD Commands
LCD_INIT_1 EQU 38H    ; 2 lines, 5x7 matrix
LCD_INIT_2 EQU 0CH    ; Display ON, cursor OFF
LCD_INIT_3 EQU 06H    ; Auto increment cursor
LCD_INIT_4 EQU 01H    ; Clear display
LCD_LINE1  EQU 80H    ; First line
LCD_LINE2  EQU 0C0H   ; Second line
; LCD Control pins on P0
RS BIT P0.0           ; Register Select
RW BIT P0.1           ; Read/Write
```

```
EN BIT P0.2            ; Enable
LCD_DATA EQU P2 ; LCD data pins connected to P2
MAIN:
    ; Initialize ports
    MOV P2, #0FFH      ; Set P2 as output for LCD data
    MOV P0, #0FFH      ; Set Port 0 high for control signal
    ; Initialize state
    MOV CURRENT_STATE, #0      ; Set the initial state to 0 (no voltage displayed)
    ; Wait for LCD to power up
    ACALL LONG_DELAY
    ACALL LONG_DELAY
    ; Initialize LCD with longer delays
    ACALL LCD_INIT
    ; Display static "Output Voltage:" text once
    MOV A, #LCD_LINE1
    ACALL LCD_CMD         ; Move cursor to first line
    MOV DPTR, #MSG1       ; Load address of "Output Voltage:" message
    ACALL LCD_STRING      ; Display the message
    ACALL LONG_DELAY   ; Wait for the display to settle
MAIN_LOOP:
    ; Check input pins P1.0 and P1.1
    JNB P1.0, CHECK_22V           ; If button 1 is pressed, jump to CHECK_22V
    JNB P1.1, CHECK_33V           ; If button 2 is pressed, jump to CHECK_33V
    SJMP MAIN_LOOP                ; Stay in the loop if no button is pressed
```

```
; Check for 2.2V State
CHECK_22V:
    MOV A, CURRENT_STATE        ; Load current state
    CJNE A, #1, SET_22V         ; If not already in 2.2V state, jump to SET_22V
    SJMP MAIN_LOOP              ; Otherwise, return to main loop

CHECK_33V:
    MOV A, CURRENT_STATE        ; Load current state
    CJNE A, #2, SET_33V         ; If not already in 3.3V state, jump to SET_33V
    SJMP MAIN_LOOP              ; Otherwise, return to main loop
SET_22V:
    CLR P1.7          ; Turn off Relay

    ; Update only second line
    MOV A, #LCD_LINE2
    ACALL LCD_CMD               ; Move cursor to second line
    ACALL CLEAR_LINE            ; Clear the second line

    MOV A, #LCD_LINE2           ; Move cursor to second line again
    ACALL LCD_CMD
    MOV DPTR, #VOLT_22          ; Load address of "2.2V" message
    ACALL LCD_STRING            ; Display "2.2V" on the second line

    MOV CURRENT_STATE, #1       ; Update current state to 2.2V
```

```
    ; Debounce delay
    ACALL LONG_DELAY
    ACALL LONG_DELAY      ; Extra delay for stability
    SJMP MAIN_LOOP

; Set Voltage to 3.3V
SET_33V:
    SETB P1.7             ; Turn on Relay

    ; Update only second line
    MOV A, #LCD_LINE2
    ACALL LCD_CMD                ; Move cursor to second line
    ACALL CLEAR_LINE             ; Clear only second line

    MOV A, #LCD_LINE2
    ACALL LCD_CMD                ; Move cursor to second line again
    MOV DPTR, #VOLT_33           ; Load address of "3.3V" message
    ACALL LCD_STRING             ; Display "3.3V" on the second line

    MOV CURRENT_STATE, #2        ; Update current state to 3.3V

    ; Debounce delay
    ACALL LONG_DELAY
```

```
    ACALL LONG_DELAY      ; Extra delay for stability
    SJMP MAIN_LOOP
; Clear Current LCD Line
CLEAR_LINE:
    MOV R5, #16                 ; Number of characters per line
    MOV A, #20H                 ; ASCII code for space (' ')
CLEAR_LINE_LOOP:
    ACALL LCD_DATA_WRITE      ; Write space to clear character
    DJNZ R5, CLEAR_LINE_LOOP  ; Repeat for all characters
    RET                         ; Return to caller

LCD_INIT:
    ; Perform initialization sequence with delays
    MOV A, #38H         ; First try - 8-bit mode
    ACALL LCD_CMD
    ACALL LONG_DELAY

    MOV A, #38H         ; Second try
    ACALL LCD_CMD
    ACALL LONG_DELAY

    MOV A, #38H         ; Third try
    ACALL LCD_CMD
    ACALL LONG_DELAY
```

```
    MOV A, #LCD_INIT_1   ; Set 2 lines, 5x7 matrix
    ACALL LCD_CMD
    ACALL LONG_DELAY

    MOV A, #LCD_INIT_2   ; Display ON, cursor OFF
    ACALL LCD_CMD
    ACALL LONG_DELAY

    MOV A, #LCD_INIT_3   ; Auto increment cursor
    ACALL LCD_CMD
    ACALL LONG_DELAY

    MOV A, #LCD_INIT_4   ; Clear display
    ACALL LCD_CMD
    ACALL LONG_DELAY
    RET
LCD_CMD:
    MOV LCD_DATA, A    ; Put command on P2
    CLR RS             ; RS = 0 for command
    CLR RW             ; RW = 0 for write
    SETB EN            ; EN = 1
    ACALL DELAY
    ACALL DELAY        ; Extra delay for stability
```

```
    CLR EN          ; EN = 0
    ACALL DELAY
    ACALL DELAY       ; Extra delay for stability
    RET
LCD_DATA_WRITE:
    MOV LCD_DATA, A    ; Put data on P2
    SETB RS          ; RS = 1 for data
    CLR RW           ; RW = 0 for write
    SETB EN         ; EN = 1
    ACALL DELAY
    ACALL DELAY       ; Extra delay for stability
    CLR EN          ; EN = 0
    ACALL DELAY
    ACALL DELAY       ; Extra delay for stability
    RET
LCD_STRING:
    CLR A
    MOVC A, @A+DPTR              ; Fetch character from memory
    JZ LCD_STRING_END            ; End if null terminator is reached
    ACALL LCD_DATA_WRITE         ; Write character to LCD
    INC DPTR                     ; Move to the next character
    SJMP LCD_STRING              ; Repeat for entire string
LCD_STRING_END:
    RET
```

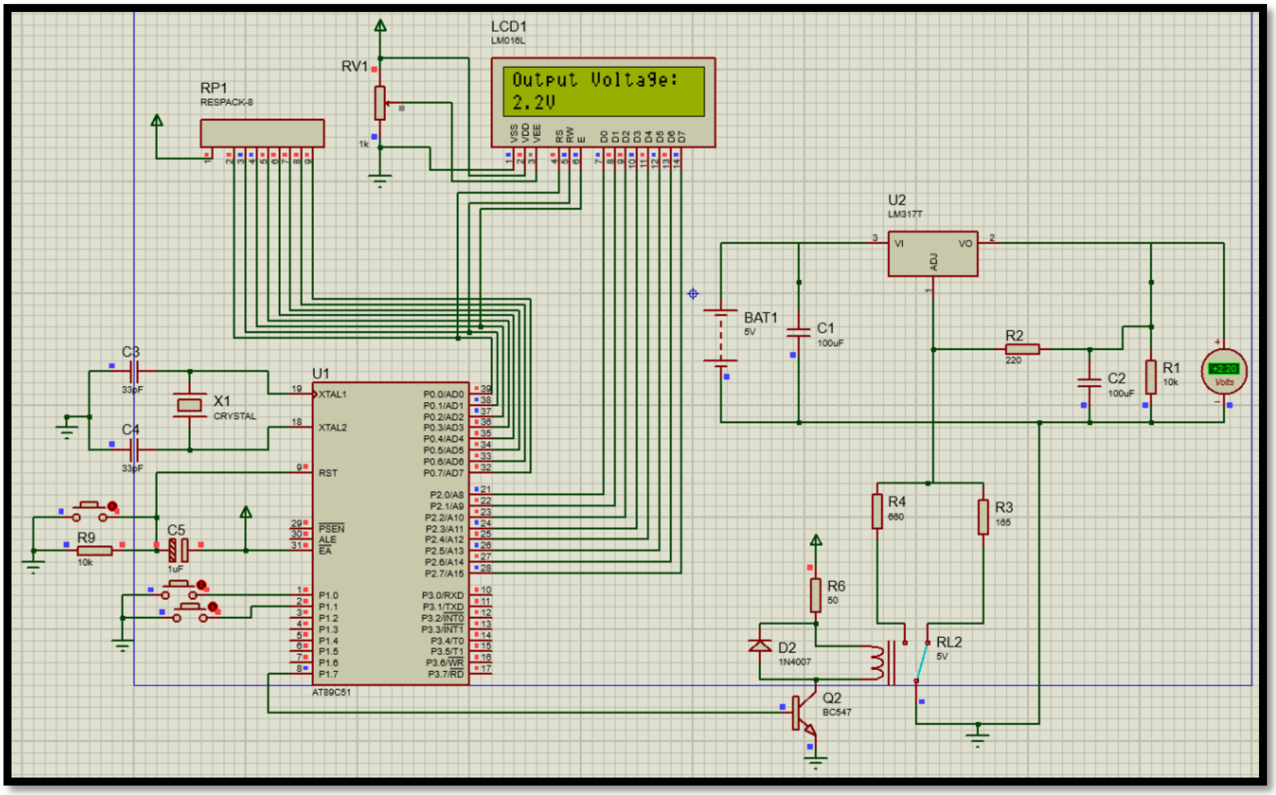
```
DELAY:
    MOV R7, #100                ; Delay counter
DELAY_LOOP:
    DJNZ R7, DELAY_LOOP         ; Decrement and loop
    RET

LONG_DELAY:
    MOV R6, #255                ; Outer loop counter
LONG_DELAY_OUTER:
    MOV R7, #255                ; Inner loop counter
LONG_DELAY_INNER:
    DJNZ R7, LONG_DELAY_INNER       ; Inner loop
    DJNZ R6, LONG_DELAY_OUTER       ; Outer loop
    RET
; Messages in code segment
MSG1:     DB 'Output Voltage:', 0       ; First line message
VOLT_22:  DB '2.2V', 0                  ; Second line message for 2.2V
VOLT_33:  DB '3.3V', 0                  ; Second line message for 3.3V
END
```

7. Testing and Results

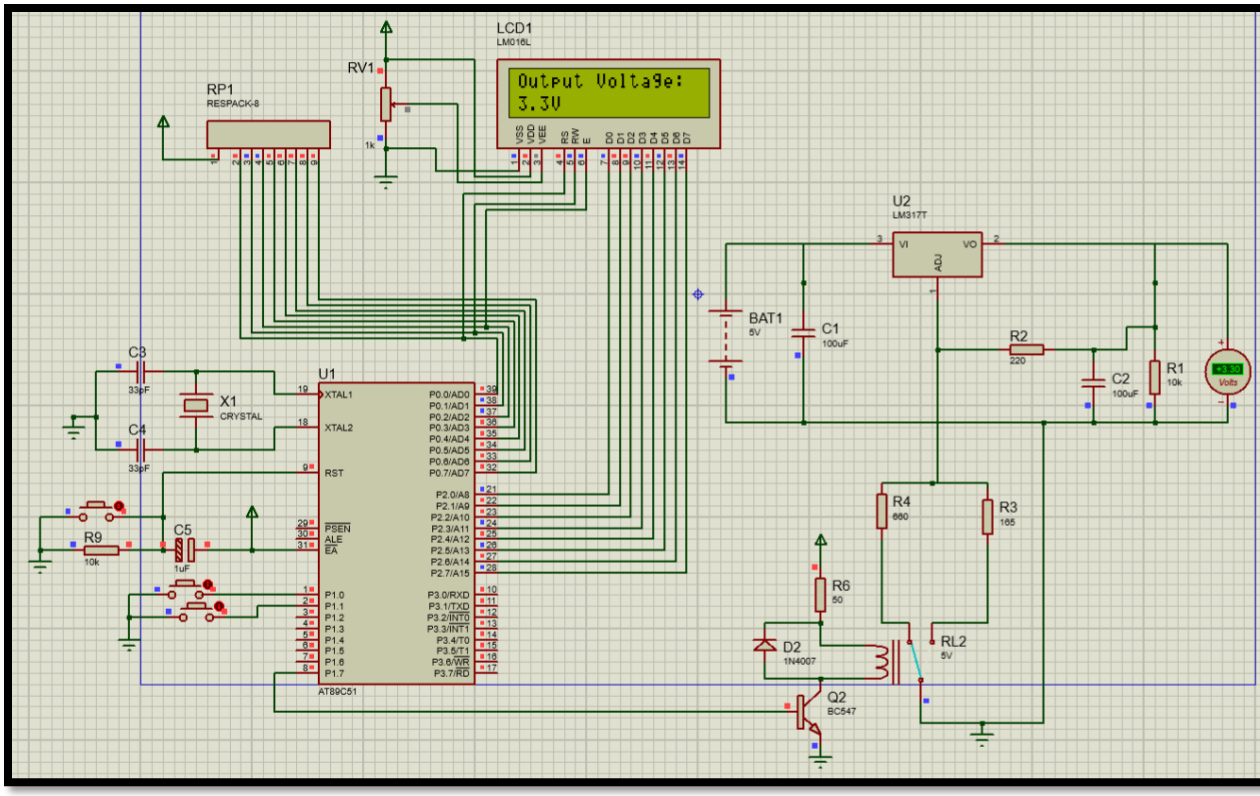
Test Cases for Different Voltage Levels

Case 1: 2.2V Output Selection



- Input Pin: P1.0 (button 1)
- Relay Control: P1.7 cleared (relay turned off)
- LCD Display: "2.2V" displayed on the second line
- System State: CURRENT_STATE updated to 1
- Verification Method:
    - Confirm relay state change
    - Validate LCD display update
    - Ensure no repeated state transitions

<u>Case 2: 3.3V Output Selection</u>

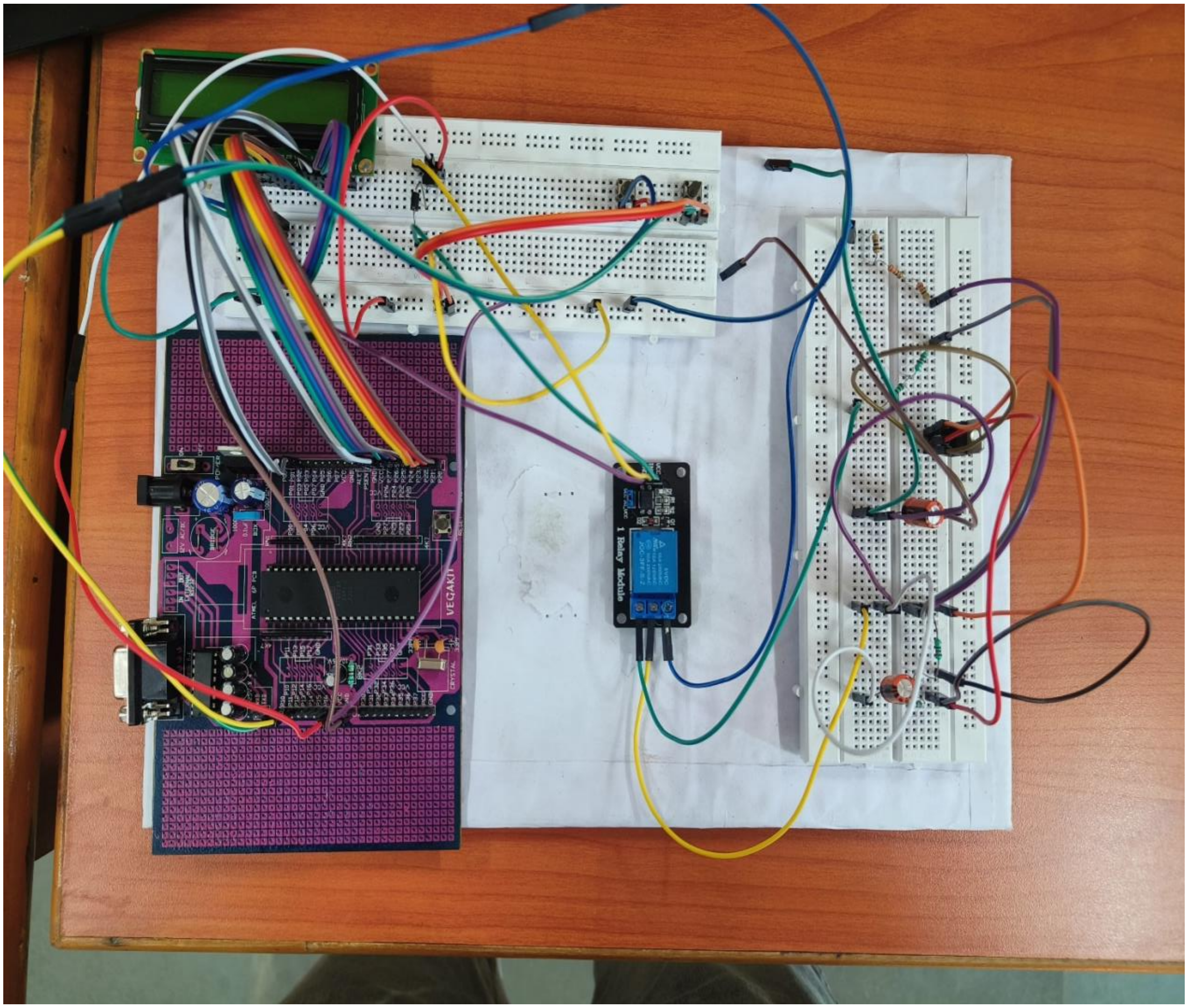


- Input Pin: P1.1 (button 2)
- Relay Control: P1.7 set (relay turned on)
- LCD Display: "3.3V" displayed on the second line
- System State: CURRENT_STATE updated to 2
- Verification Method:
    - Confirm relay state change
    - Validate LCD display update
    - Ensure no repeated state transitions

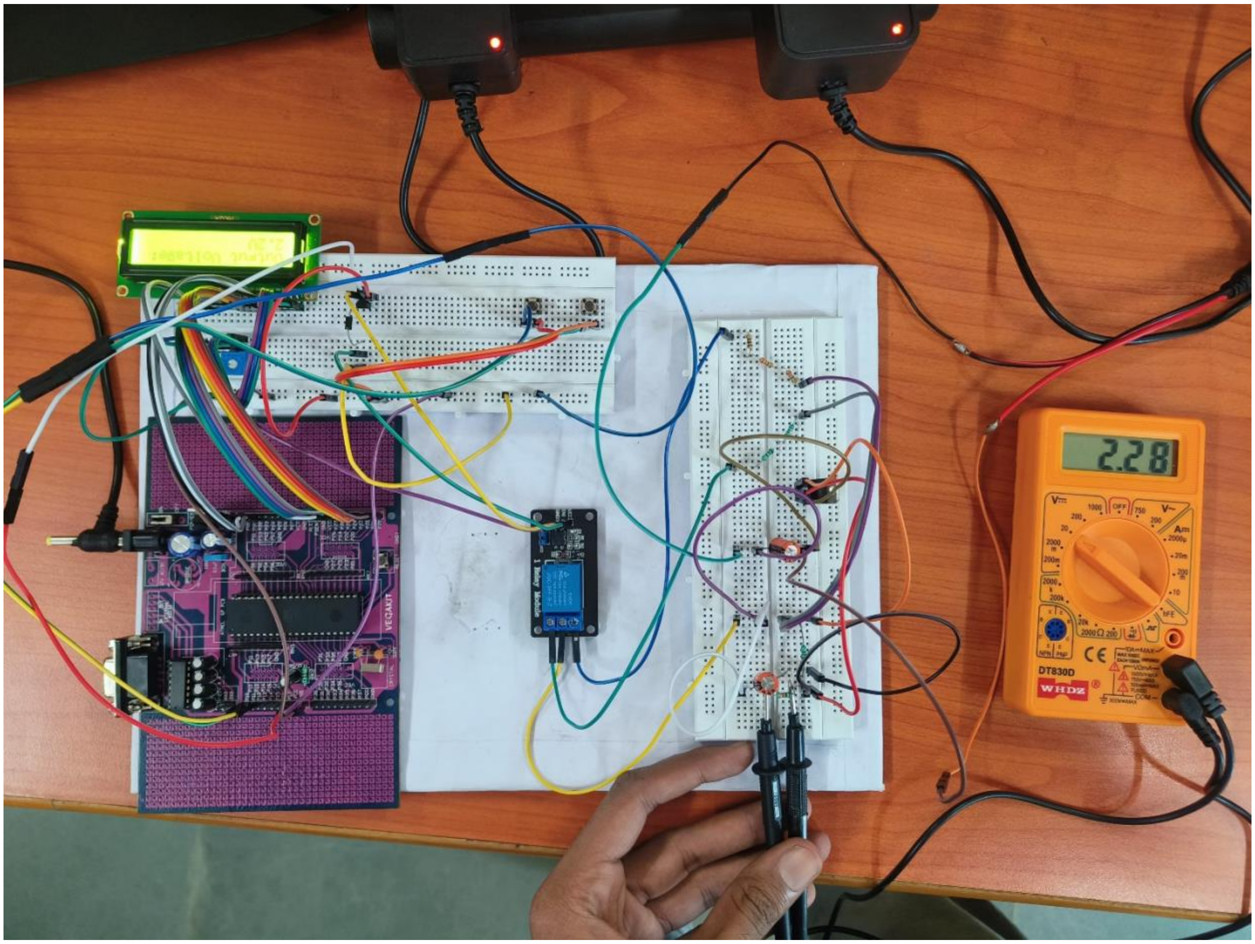
Measuring Stability of Output Voltages

The output voltages were measured under different load conditions using a multimeter. The voltage readings were stable for both 2.2V and 3.3V, with minimal fluctuation (within ±0.05V). This confirms that the LM317 maintained a stable output even when switching between levels or under varying loads, ensuring consistent performance for sensitive electronic components.

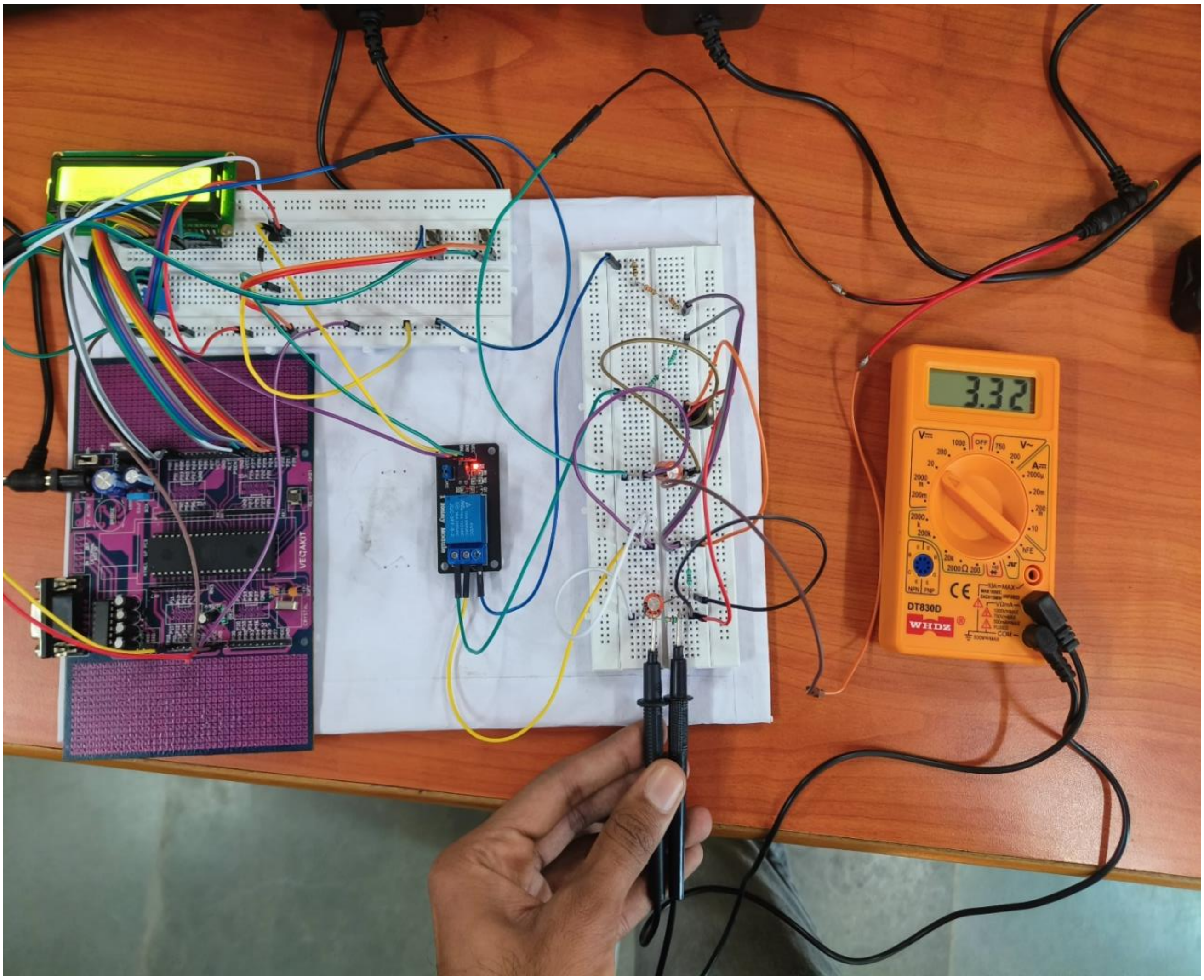
8. Hardware Design



Hardware design of Schematic (inactive)



Hardware design of Schematic (Active and Output = 2.2V)



Hardware design of Schematic (Active and Ouput = 3.3V)

9. Applications and Utilization

This system has wide applications in any scenario requiring **multiple voltage levels** from a single power source. It's especially useful in **electronic accessory circuits**, where components often require different stable voltages (e.g., sensors, communication modules). By providing **2.2V** and **3.3V** outputs from a **5V source**, this system simplifies circuit design and power management.

In embedded systems, devices frequently rely on **multiple voltage levels** to power different components. For example, **microcontrollers** may run on 3.3V, while peripherals like **LCDs** or sensors need 2.2V. This solution ensures efficient power delivery without needing multiple power supplies.

The system is also highly relevant in **prototype development**, where flexibility in voltage regulation is needed to test various components. Furthermore, it can be utilized in **portable electronics**, where reducing the number of power sources simplifies device design and enhances efficiency.

This approach could also be scaled to manage different voltage levels for **IoT devices**, **robotics**, or **home automation**, where stable, selectable voltage is critical for performance and power efficiency.

Conclusion

This project successfully created a system capable of switching between **2.2V** and **3.3V** outputs using an **LM317T** voltage regulator and an **8051 microcontroller**. The relays, controlled by SPST switch, allowed seamless toggling between the voltage levels. The system demonstrated stability under different load conditions and effectively solved the problem of needing multiple output voltages from a single 5V input. Future enhancements could improve precision and scalability for wider applications.