REPUBLIC OF THE PHILIPPINES

SURIGAO DEL NORTE STATE UNIVERSITY

Narcisso Street, Surigao City 8400, Philippines

Caspe,Mark Angelo J.

BSECE-3A

Laboratory Activity no.11

Larger Loads:Relays

**Introduction:**

The final circuit is a bit of a test. We combine what we learned about using transistors in CIRC03 to control a relay. A relay is an electrically controlled mechanical switch. Inside the little plastic box is an electromagnet that, when energized, causes a switch to trip (often with a very satisfying clicking sound). You can buy relays that vary in size from a quarter of the size of the one in this kit up to as big as a fridge, each capable of switching a certain amount of current. They are immensely fun because there is an element of the physical to them. While all the silicon we've played with to this point is fun sometimes you just want to wire up a hundred switches to control something magnificent. Relays give you the ability to dream it up then control it with your Arduino. Now to using todays technology to control the past.

**Objectives:**

The objective of this activity is likely to simulate the control of a relay using transistors in Proteus. This simulation would allow you to understand how to use modern technology (like transistors and microcontrollers) to control older technology (like relays) in electronic circuits. It's a practical exercise in bridging the gap between different generations of electronic components and understanding how they can work together in a circuit.

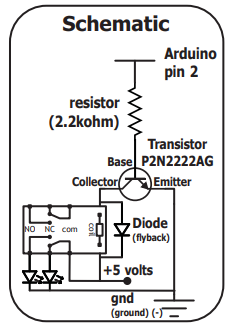
**Materials:**

1.) Atmega328P 2.) Diode (1N4001) x1 3.) Transistor P2N2222AG (TO92) x1

4.) Relay (DPDT) x1 5.) 2.2k Ohm Resistor 6.) 150 Ohm Resistor

7.) Green LED x1 8.) Red LED x1 9.) 22pf Capacitor

**Schematic Diagram:**



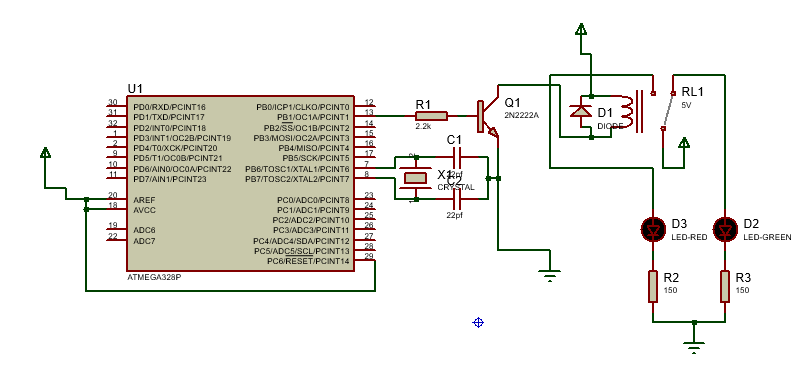
**Procedure:**

* Open Proteus and Create a New Project: Launch Proteus and start a new project where you'll build and simulate the circuit.
* Find the Components: Navigate to the Proteus component library and add the necessary components for the circuit:Arduino board (e.g., Arduino Uno)Transistors (for controlling the relay)RelaySupporting components like resistors and wires
* Refer to Schematic Diagram or Datasheets: Use the schematic diagram or datasheets for guidance on how to connect the components. Ensure you understand how to wire the transistors to control the relay effectively
* Write Arduino Code: Open the Arduino IDE and write the code to control the transistors, which in turn will control the relay. Ensure the code correctly toggles the transistor states to energize or de-energize the relay as needed.
* Save the Arduino Code: Once you've written the code, save it in a convenient location on your computer.
* Add Microcontroller to Proteus: In Proteus, add the appropriate microcontroller (e.g., ATmega328P) to your project.
* Load Arduino Code onto Microcontroller: Click on the microcontroller in Proteus and load the saved Arduino code onto it. This will simulate the behavior of the Arduino controlling the transistors and relay.
* Simulate the Circuit: Run the simulation in Proteus to observe how the Arduino controls the transistors, which in turn control the relay. Verify that the relay switches on and off in response to the Arduino code.

**Results and Discussion:**

The simulation results show that the Arduino successfully controls the relay, with the green and red LEDs indicating its state. This demonstrates the ability to switch the relay using modern technology. Further discussion could focus on the reliability of the relay switching and potential improvements for the circuit or code.

**Circuit:**



**Program:**

const int DRIVE\_PIN = 9;

void setup()

{

  pinMode(DRIVE\_PIN, OUTPUT);

}

void loop()

{

  digitalWrite(DRIVE\_PIN, HIGH);

  delay(50);

  digitalWrite(DRIVE\_PIN, LOW);

  delay(50);

}

**Findings:**

In simple terms, the simulation shows that the Arduino can control the relay, switching between green and red LEDs to indicate its state. This proves that modern technology can operate older tech like relays. Further study could focus on the timing and reliability of the switching process.

**Recommendations:**

Adjust the timing in the Arduino code, check for bouncing issues in relay contacts, test the circuit in real-world conditions, choose components carefully, and prioritize safety.

**Conclusions:**

In conclusion, the simulation demonstrates successful control of the relay by the Arduino, with the green and red LEDs indicating its state. This highlights the effective integration of modern technology with older relay technology. Further testing and optimization can enhance the circuit's performance and reliability in practical applications.