REPUBLIC OF THE PHILIPPINES

SURIGAO DEL NORTE STATE UNIVERSITY

Narcisso Street, Surigao City 8400, Philippines

Caspe,Mark Angelo J

BSECE-3A

Laboratory Activity no.3

Spin Motor Spin:Transistor & Motor

**Introduction:**

The Arduino's pins are great for directly controlling small electric items like LEDs. However, when dealing with larger items (like a toy motor or washing machine), an external transistor is required. A transistor is incredibly useful. It switches a lot of current using a much smaller current. A transistor has 3 pins. For a negative type (NPN) transistor you connect your load to collector and the emitter to ground. Then when a small current flows from base to the emitter a current will flow through the transistor and your motor will spin (this happens when we set our Arduino pin HIGH). There are literally thousands of different types of transistors, allowing every situation to be perfectly matched. We have chosen a P2N2222AG a rather common general purpose transistor.

**Objectives:**

The objective of this activity is to demonstrate how to use an Arduino to control larger electrical components, such as a toy motor, using a transistor for switching. By simulating the circuit in Proteus, users can understand the basic principles of transistor operation and how to interface it with an Arduino. The simulation provides a safe environment to experiment with different configurations and ensure proper functionality before implementing the circuit in hardware.

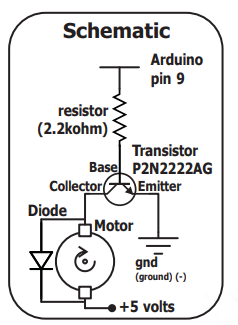
**Materials:**

1.) Atmega328P 2.) Crystal 3.) TransistorP2N2222AG (TO92) x1 4.) 22pf Capacitor

5.) Toy Motor x1 6.) Diode (1N4001) x1 7.) 2.2k Ohm Resistor Red-Red-Red x1

8.)Push Button

**Schematic Diagram:**



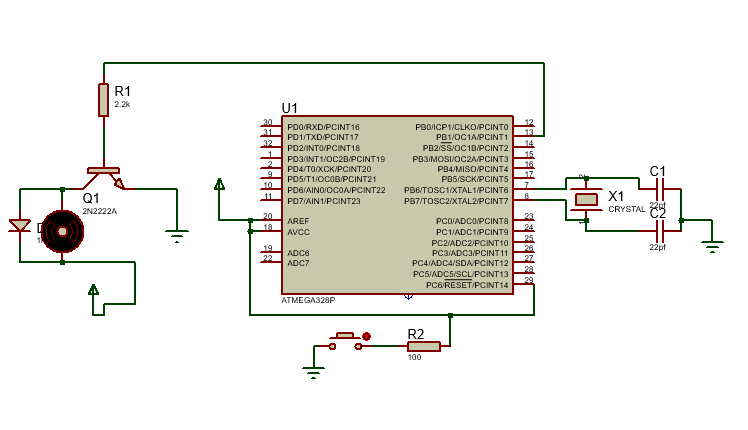
**Procedure:**

* Opening Proteus.
* Referring to the schematic diagram, assemble or connect the components accordingly.
* Switch to Arduino IDE and create the program for controlling the toy motor.Save the program.
* Return to Proteus, click on ATmega328P, and load the saved program (.hex file).
* Simulate the circuit in Proteus to run the program and observe its behavior.

**Results and Discussion:**

In the simulation conducted in Proteus, the motor initiates spinning as expected upon running the code. Through adjustments made in the code, precise control over the motor's speed is achieved. By manipulating the values within the analogWrite() function, responsible for setting the PWM output, users can regulate the motor's speed. Lower values yield slower speeds, while higher values escalate the speed accordingly. This capability empowers users to finely tune the motor's performance, enabling seamless adaptation to diverse application requirements.

**Circuit:**

****

**Program:**

int motorPin = 9;

void setup()

{

 pinMode(motorPin, OUTPUT);

}

void loop()

{

 motorOnThenOff();

 motorOnThenOffWithSpeed();

 motorAcceleration();

}

void motorOnThenOff()

{

  int onTime = 2500;

  int offTime = 1000;

  digitalWrite(motorPin, HIGH);

  delay(onTime);

  digitalWrite(motorPin, LOW);

  delay(offTime);

}

void motorOnThenOffWithSpeed()

{

  int onSpeed = 200;

  int onTime = 2500;

  int offSpeed = 50;

  int offTime = 1000;

  analogWrite(motorPin, onSpeed);

  delay(onTime);

  analogWrite(motorPin, offSpeed);

  delay(offTime);

}

void motorAcceleration()

{

  int delayTime = 50;

  for(int i = 0; i < 256; i++){

    analogWrite(motorPin, i);

    delay(delayTime);

  }

  for(int i = 255; i >= 0; i--){

    analogWrite(motorPin, i);

    delay(delayTime);

  }

}

**Finding:**

The findings of this activity demonstrate that by using an Arduino and a transistor, you can effectively control the speed of a motor in a Proteus simulation. By adjusting the values within the code's analogWrite() function, users can regulate the motor's speed, allowing for versatile application in various projects. This level of control ensures that the motor's performance can be customized to meet specific requirements, enhancing its utility in practical applications.

**Recommendations:**

Based on the findings of this activity, it is recommended to explore further experimentation with motor control in Proteus using Arduino. This could involve testing different PWM values to understand their impact on motor speed more comprehensively. Additionally, exploring the integration of sensors or external inputs to dynamically adjust motor speed based on environmental conditions or user inputs could enhance the versatility and functionality of the system. Lastly, documenting and sharing the results and code snippets can benefit others in the Arduino and Proteus community, fostering collaboration and knowledge sharing.

**Conclusions:**

In conclusion, this activity shows that using an Arduino and a transistor in Proteus allows for precise control over motor speed. By adjusting code values, users can easily customize the motor's performance for different projects. Further experimentation and sharing findings can help expand knowledge and applications in Arduino-based motor control.