Sushant University

EMBEDDED SYSTEM WITH IOT

Lab Report File

Prepared By

Abhijit Mandal

V SEMESTER

210BTCSEAM024

B.TECH. CSE AI/ML

Experiment → 1

AIM: Interface a LED with the microcontroller chip ATMEGA328 in Proteus and WAP in IDE to simulate the blinking of LED in circuit.

Materials Required:

- Computer with Proteus software installed.
- ATmega328 microcontroller.
- 16 MHz crystal oscillator.
- Two 22pF capacitors.
- LED
- Power supply module (5V).
- Ground module.

Theory:

ATmega328 Microcontroller:

- The ATmega328 is an 8-bit microcontroller. It is widely used in various embedded systems and Arduino development boards.
- Features include 32KB of Flash memory for program storage, 1KB of EEPROM for data storage, 2KB of SRAM for data manipulation, and multiple I/O pins for interfacing with external components.
- In this experiment, the ATmega328 serves as the central processing unit responsible for controlling the LED.

Crystal Oscillator (16 MHz):

- The crystal oscillator provides the clock signal for the microcontroller, allowing it to execute instructions at a precise rate.
- A 16 MHz crystal oscillator is commonly used with the ATmega328 to provide accurate timing for the microcontroller's operations.

Capacitors (22pF):

- Two 22pF capacitors are used in conjunction with the crystal oscillator.
- These capacitors stabilise the clock signal by reducing noise and ensuring a clean and stable clock input to the microcontroller.

LED (Light Emitting Diode):

- An LED is a semiconductor device that emits light when a current passes through it.
- LEDs are commonly used for visual indicators in electronic circuits.
- In this experiment, the LED is connected to a digital output pin of the ATmega328 and blinks on and off to demonstrate the microcontroller's control over external components.

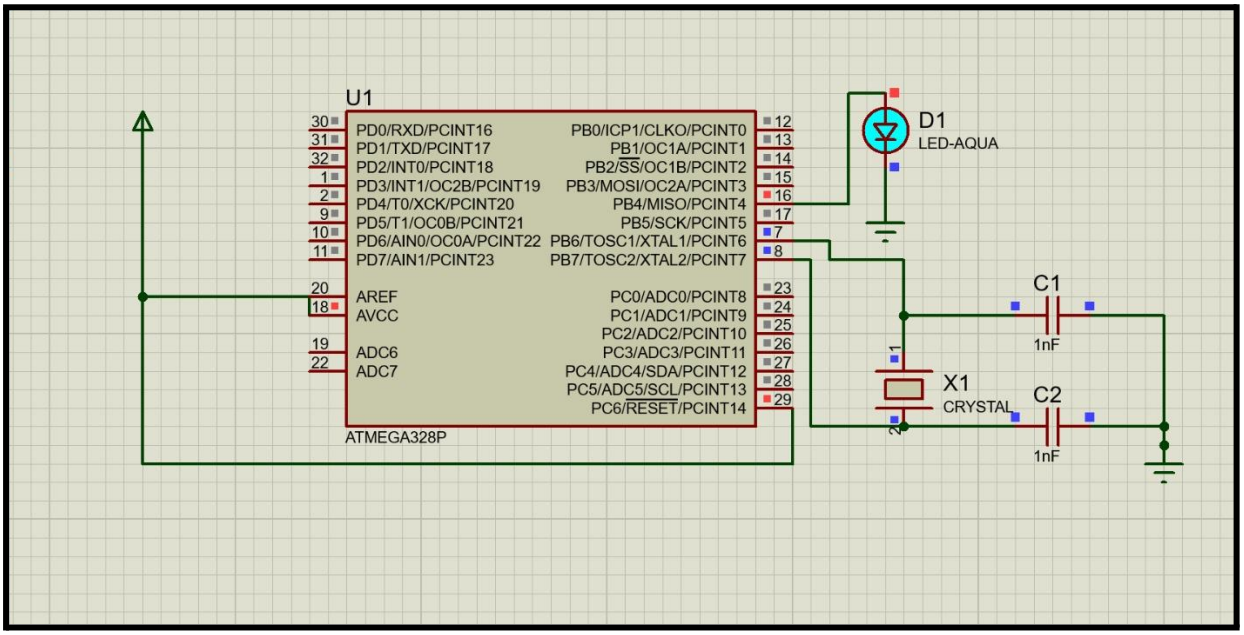
Power Supply (5V):

- The power supply module provides a stable 5-volt voltage source to power the ATmega328 and other components in the circuit.
- It ensures that the microcontroller operates within its specified voltage range.

Ground (GND):

- The ground module is used to establish a common ground reference for all components in the circuit.
- It ensures that voltage levels are measured relative to a common reference point, allowing for proper circuit operation.

Diagram:



Experiment Procedure:

1. Search for all the components in the Proteus Component Library and place it on the workspace.
2. Add a 16 MHz crystal oscillator and connect it to the XTAL pins (pins 9 and 10) of the ATmega328.
3. Connect two 22pF capacitors between each XTAL pin and ground.
4. Place an LED on the workspace and connect its anode (longer lead) to a digital output pin of the ATmega328 (e.g., Pin 12). Connect the cathode (shorter lead) to ground.
5. Right-click on the ATmega328 and select "Edit Properties".
6. Write the code in Arduino IDE.Click on the "Program File" option and browse to select the firmware (HEX file) for the ATmega328, which contains the program to blink the LED. You should have prepared this firmware beforehand using the Arduino IDE or AVR Studio.
7. Click OK to apply the settings.
8. Click on the "Play" button (green triangle) in the Proteus toolbar to start the simulation. The LED should start blinking according to the program you uploaded to the ATmega328.
9. Adjust the simulation speed using the "Run" menu to observe the LED blinking at different rates.

Code:

```
void setup(){
      pinMode(12, OUTPUT);
}
void loop(){
     digitalWrite(12, HIGH);
     delay(1000);
     digitalWrite(12, LOW);
     delay(200);
}
```

Result:

- The blinking of LED (Light Emitting Diode).

Experiment ➔ 2

AIM: WAP in IDE and stimulate the circuit in Proteus to use the Push Button.

Materials Required:

- Computer with Proteus software installed.
- ATmega328 microcontroller.
- LED
- Push Button.
- Resistor.
- Arduino IDE.

Theory:

ATmega328 Microcontroller :

- The ATmega328 is an 8-bit microcontroller. It is widely used in various embedded systems and Arduino development boards.
- Features include 32KB of Flash memory for program storage, 1KB of EEPROM for data storage, 2KB of SRAM for data manipulation, and multiple I/O pins for interfacing with external components.
- In this experiment, the ATmega328 serves as the central processing unit responsible for controlling the LED.

Arduino UNO :

- The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc.
- The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.

LED :

- A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it.

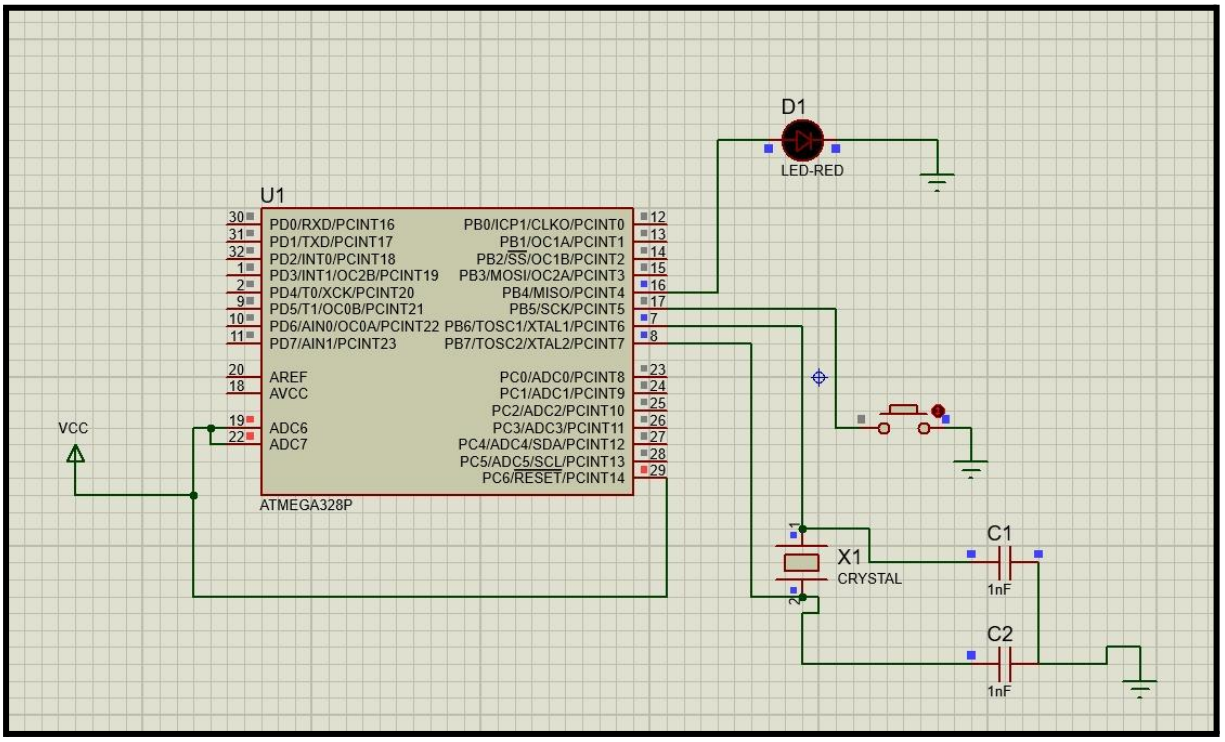
Resistors :

- A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element.

Push Button :

- A push button is a simple type of switch that controls an action in a machine or some type of process.

Diagram:



Experiment Procedure:

1. Open Proteus window.
2. Select 'New Project' and give suitable project name and location to the project. A design sheet will be opened.
3. Select the 'component mode' button and select LED, resistor, push button. Place them at an appropriate distance apart.
4. Make connections by left clicking from one terminal to another.
5. Start Arduino IDE and write the code.
6. Export the code to the compiled binary.
7. Double click AtMega328 and export the compiled file in edit properties.
8. Click the start button in the leftmost corner.
9. The required output will be shown on the screen.

Arduino IDE Code :

```
const switchPin = 2; //Connect the switch to digital pin 2.
const ledPin = 13; // use the built-in LED on the Arduino UNO.

void setup(){
 pinMode(switchPin, INPUT); // set switch as INPUT.
 pinMode(ledPin, OUTPUT); // set LED as OUTPUT.
}

void loop(){
 int switchState = digitalRead(switchPin); //Read the state of the switch.

 if (switchState == LOW){ // check if the switch is not pressed (LOW).
  digitalWrite(ledPin, HIGH); // Turn on the LED.
 }
 else{
  digitalWrite(ledPin, LOW); // Turn off the LED.
 }
}
```

Result:

- The **LED** (Light Emitting Diode) will turn On/Off using the **Push Button**.

Experiment → 3

AIM: Control the intensity of LED using potentiometer & ATmega328 on Proteus.

Materials Required:

- Computer with Proteus software installed.
- ATmega328 microcontroller
- Potentiometer
- Virtual Monitor
- LED
- Power Supply Module (5V)
- Ground Module
- Arduino IDE

Theory:

ATmega328 Microcontroller :

- The ATmega328 is an 8-bit microcontroller. It is widely used in various embedded systems and Arduino development boards.
- Features include 32KB of Flash memory for program storage, 1KB of EEPROM for data storage, 2KB of SRAM for data manipulation, and multiple I/O pins for interfacing with external components.
- In this experiment, the ATmega328 serves as the central processing unit responsible for controlling the LED.

Arduino UNO :

- The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc.
- The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.

LED :

- An LED is a semiconductor device that emits light when a current passes through it.
- LEDs are commonly used for visual indicators in electronic circuits.
- In this experiment, the LED is connected to a digital output pin of the ATmega328 and blinks on and off to demonstrate the microcontroller's control over external components.

Potentiometer :

- A potentiometer, often referred to as a "pot," is a variable resistor with three terminals.
- By adjusting the potentiometer's shaft, you can change the resistance between its terminals.
- In this experiment, the potentiometer acts as an analog input device, allowing you to vary the voltage applied to the ATmega328's analog pins. This is useful for simulating analog sensor inputs.

Virtual Monitor:

- The virtual monitor in Proteus simulates the output display of data or information.
- It can be used to visualise data or results generated by the microcontroller.

- In this experiment, the virtual monitor represents the display interface where the output of the microcontroller, such as sensor readings or messages, can be observed.

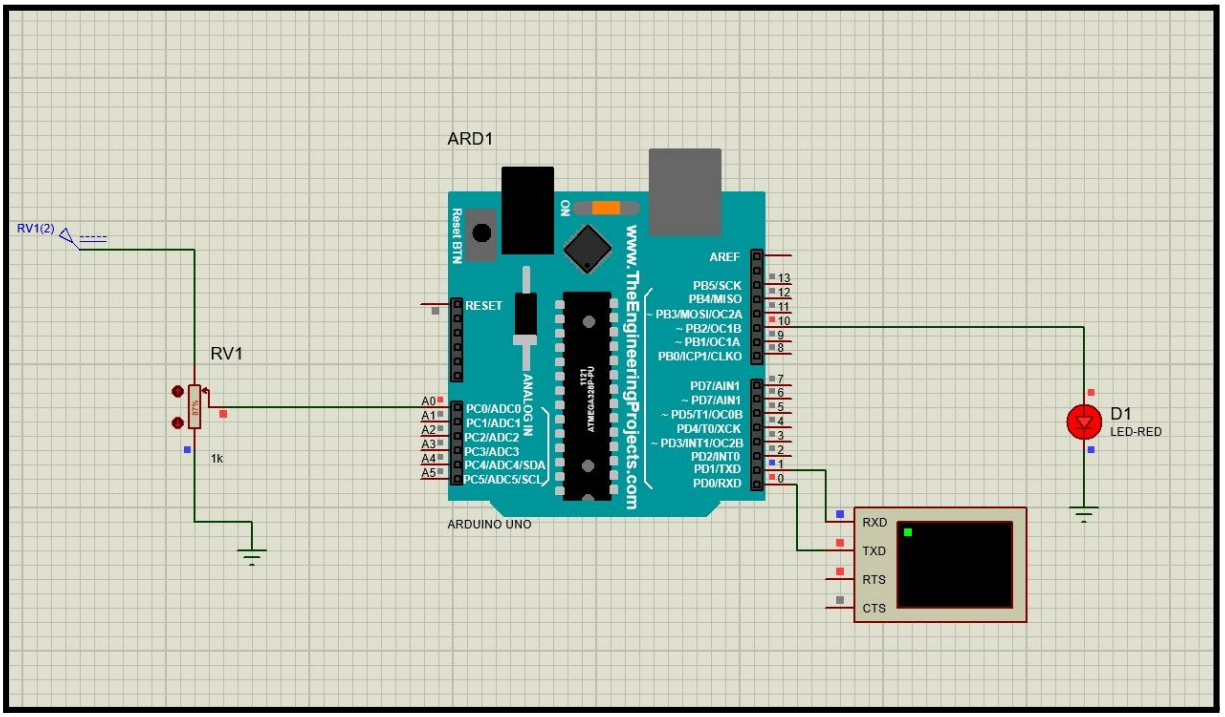
Power Supply (5V):

- The power supply module provides a stable 5-volt voltage source to power the ATmega328 and other components in the circuit.
- It ensures that the microcontroller operates within its specified voltage range.

Ground (GND):

- The ground module is used to establish a common ground reference for all components in the circuit.
- It ensures that voltage levels are measured relative to a common reference point, allowing for proper circuit operation.

Diagram



Experiment Procedure

1. Set up the ATmega328, potentiometer, and LED as described in the previous procedure.
2. Connect one end (usually the left terminal) of the potentiometer to the 5V power supply in Proteus.
3. Connect the other end (usually the right terminal) of the potentiometer to the ground (GND) in Proteus.
4. Connect the wiper terminal (the middle terminal of the potentiometer) to one of the analog input pins on the ATmega328. Typically, you would use analog pin A0.
5. Connect the virtual terminal's pins as follows:
    a. Connect the RX pin of the virtual terminal to a digital output pin of the ATmega328 (e.g., Pin 1).
    b. Connect the TX pin of the virtual terminal to a digital input pin of the ATmega328 (e.g., Pin 2)
6. Right-click on the ATmega328 and select "Edit Properties."
7. Click on the "Program File" option and browse to select the firmware (HEX file) for the ATmega328.

Arduino IDE Code :

```
int ledPin = 10;
int ptmPin = A0;
float ptmValue = 0;

void setup(){
 pinMode(ledPin, OUTPUT); // set led as OUTPUT.
 pinMode(ptmPin, INPUT); // set Potentiometer as INPUT.
 Serial.begin(9600);
}

void loop(){
 float ptmValue = analogRead(ptmPin); //Read the ptmValue from ptm.
 float brightness = map(ptmValue, 0, 1023, 0, 255);
 analogWrite(ledPin, brightness);
 Serial.println(brightness);
 delay(100);
}
```

Result:

- The intensity of the **LED** (Light Emitting Diode) changes due to the change in the resistance of the **Potentiometer**.

Experiment → 4

AIM: Control the intensity of LED using LDR & Arduino UNO in Proteus.

Materials Required:

- Computer with Proteus software installed.
- ATmega328 microcontroller.
- Arduino UNO
- Virtual Monitor.
- LED
- LDR
- Resistors (10k)
- Power supply module VCC (5V).
- Ground module.

Theory:

ATmega328 Microcontroller :

- The ATmega328 is an 8-bit microcontroller. It is widely used in various embedded systems and Arduino development boards.
- Features include 32KB of Flash memory for program storage, 1KB of EEPROM for data storage, 2KB of SRAM for data manipulation, and multiple I/O pins for interfacing with external components.
- In this experiment, the ATmega328 serves as the central processing unit responsible for controlling the LED.

Arduino UNO :

- The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc.
- The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.

LED :

- An LED is a semiconductor device that emits light when a current passes through it.
- In this experiment, the LED is connected to a digital output pin of the ATmega328 and blinks on and off to demonstrate the microcontroller's control over external components.

LDR (Light Dependent Resistor):

- A Light Dependent Resistor, commonly known as an LDR or a photocell, is a passive electronic component that exhibits a change in its resistance with changes in ambient light levels.
- In this experiment, the LDR is employed as a light sensor. It acts as an analog input device, similar to the potentiometer, but it responds to changes in light rather than adjusting resistance manually. The LDR's resistance decreases as the ambient light level increases and vice versa.

Virtual Monitor:

- The virtual monitor in Proteus simulates the output display of data or information.
- It can be used to visualise data or results generated by the microcontroller.

- In this experiment, the virtual monitor represents the display interface where the output of the microcontroller, such as sensor readings or messages, can be observed.

Resistors :

- A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element.

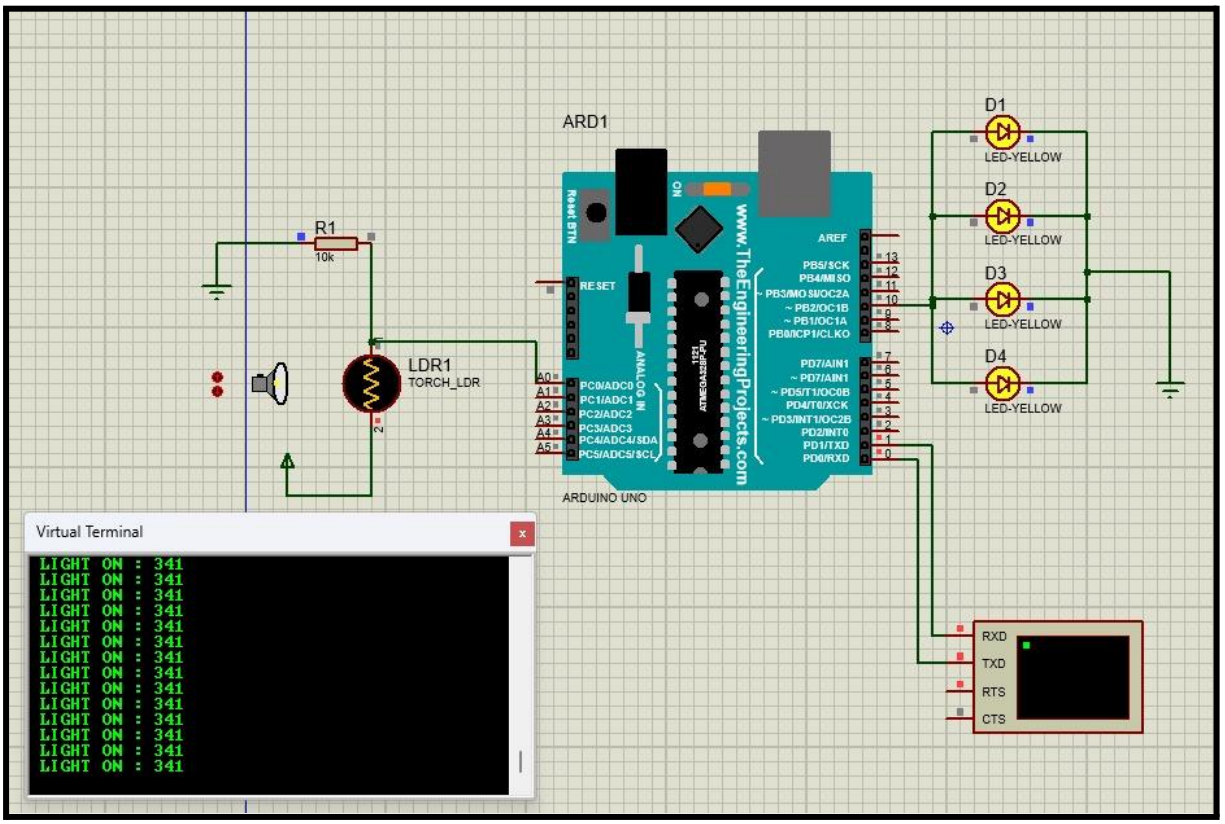
Power Supply (5V):

- The power supply module provides a stable 5-volt voltage source to power the ATmega328 and other components in the circuit.
- It ensures that the microcontroller operates within its specified voltage range.

Ground (GND):

- The ground module is used to establish a common ground reference for all components in the circuit.
- It ensures that voltage levels are measured relative to a common reference point, allowing for proper circuit operation.

Diagram



Experiment Procedure

1. Open Proteus
    - Launch the Proteus software on your computer.
2. Create a New Project
    - Create a new project in Proteus and save it with a suitable name.
3. Select Components
    - In the Proteus workspace, locate and select the components you need:
        i. Arduino UNO (from the "Arduino" library)
        ii. LDR (Light Dependent Resistor)
        iii. LED (Light Emitting Diode)
        iv. Potentiometer (if used)
        v. Virtual Monitor (for display)
4. Wiring Connections
    - Connect the components by wiring them together.
        i. Connect the LDR to one of the analog input pins of the Arduino UNO.
        ii. Connect the LED to one of the digital output pins of the Arduino UNO.
        iii. Connect the virtual monitor to the Arduino UNO for data display.

5. Configure Arduino Code
    - Write the Arduino code to read the LDR's analog input, adjust the LED's intensity based on the LDR value, and display relevant information on the virtual monitor.
6. Upload Code
    - Upload the Arduino code to the virtual Arduino UNO in Proteus.
7. Simulate the Experiment
    - Run the simulation in Proteus to start the experiment.
    - Adjust the potentiometer or vary the light intensity on the LDR to observe changes in LED brightness on the virtual monitor.
8. Analyse Results
    - Carefully observe and analyse the behaviour of the LED's intensity as it responds to changes in light detected by the LDR.
9. Save and Document

Arduino IDE Code

```
int ldrPin = A0;
int ledPin = 10;
void setup(){
 pinMode(10, OUTPUT); // set led as OUTPUT.
 pinMode(A0, INPUT); // set LDR as INPUT.
 Serial.begin(9600);
}
void loop(){
 float pinValue = analogRead(ldrPin);
 if (pinValue <= 500){
   digitalWrite(10, HIGH);
   Serial.println("Light On : ");
   Serial.println(pinValue);
 }
 else{
   digitalWrite(10, LOW);
   Serial.println("Light Off: ");
   Serial.println(pinValue);
 }
}
```

Result: The intensity of the **LED(Light Emitting Diode)** changes due to the change in the resistance of the **LDR (Light Dependent Resistor)**.

Experiment ➔ 5

AIM: Design a model using LM35 and Atmega328 to print temp value on 16x2 lcd.

Materials Required:

- Computer with Proteus software installed.
- Arduino UNO
- Arduino IDE
- 16x2 LCD
- LM35
- Power supply module VCC (5V).
- Ground module.

Theory:

Arduino UNO :

- The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc.
- The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.

16x2 LCD :

- A 16x2 LCD display is a very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines.

Arduino IDE :

- The Arduino Integrated Development Environment is a cross-platform application that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards.

LM35 Temperature Sensor :

- LM35 is a temperature measuring device having an analog output voltage proportional to the temperature.
- It provides output voltage in Centigrade (Celsius).

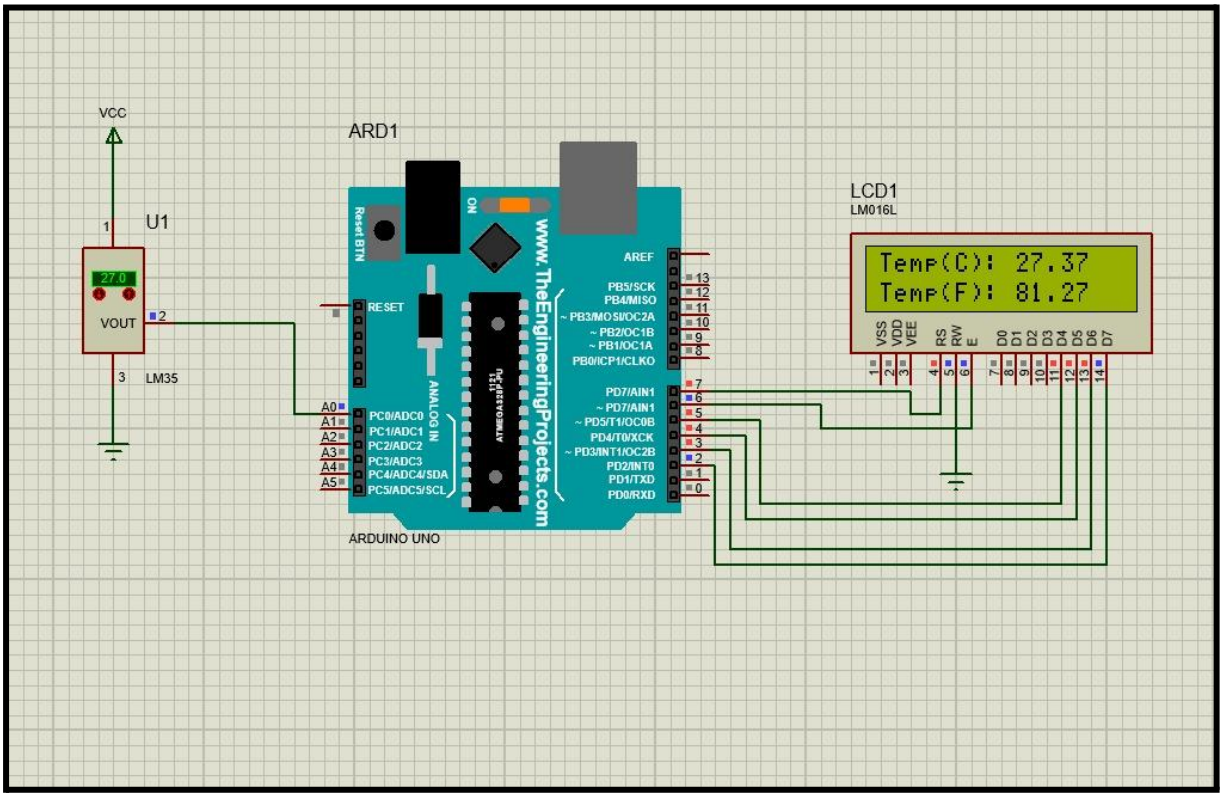
Power Supply (5V):

- The power supply module provides a stable 5-volt voltage source to power the Arduino UNO and other components in the circuit.
- It ensures that the microcontroller operates within its specified voltage range.

Ground (GND):

- The ground module is used to establish a common ground reference for all components in the circuit.
- It ensures that voltage levels are measured relative to a common reference point, allowing for proper circuit operation.

Diagram



Experiment Procedure

1. Open Proteus
    - Launch the Proteus software on your computer.
2. Create a New Project
    - Create a new project in Proteus and save it with a suitable name.
3. Select Components
    - In the Proteus workspace, locate and select the components you need:
        i. Arduino UNO (from the "Arduino" library)
        ii. LM35 (Temperature Sensor)
        iii. 16x2 LCD (Liquid Crystal Diode)
        iv. VCC (Power Supply)
        v. GND (Ground)
4. Writing Connections
    - Connect the components by wiring them together.
        i. Connect the LM35 Temperature Sensor to one of the analog input pins of the Arduino UNO.
        ii. Connect the LCD to the digital output pins of the Arduino UNO.
        iii. Connect the power supply VCC to the LM35 Sensor.
5. Configure the Code
    - Write the Arduino code to read the LM35's analog input, convert the sensor value into degree celsius and degree Fahrenheit based on the LM35 value, and display the temperature in both units on the 16x2 LCD.
6. Upload Code
    - Upload the Arduino code to the virtual Arduino UNO in Proteus.
7. Simulate the Experiment
    - Run the simulation in Proteus to start the experiment.
    - Observe changes in LM35 and what’s displaying on the LCD.
8. Save and Document

Arduino IDE Code

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(7,6,5,4,3,2);

const int lm35Pin = A0;

void setup(){
 lcd.begin(16, 2);
 lcd.print("Temperature");
}

void loop(){

 int sensorValue = analogRead(lm35Pin);
 float temperatureC = (sensorValue / 1023.0) * 500;
 float temperatureF = (temperatureC * 9.0 /5.0) + 32.0;

 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Temp(C): ");
 lcd.print(temperatureC);
 lcd.setCursor(0, 1);
 lcd.print("Temp(F): ");
 lcd.print(temperatureF);
 delay(1000)
}
```

Result: The respective temperature values both in degree celsius and degree fahrenheit are displayed successfully on the **16x2 LCD**.