**Arduino Volt LCD**

**Measure AC/DC voltage from 9v battery and display on LCD**

**Introduction:**

Voltage measurement is a fundamental task in the field of electronics, and there are many devices and tools available to perform this task. However, with the advancement of technology, microcontrollers like the Arduino have become popular for voltage measurement tasks. The Arduino Volt LCD project is an electronics project that involves using an Arduino microcontroller to measure AC/DC voltage and display the results on an LCD screen. This project has a wide range of applications in various industries and can be easily customized to fit specific project requirements.

**Background:**

Measuring voltage is a crucial task in many electronic projects. Traditionally, voltage is measured using a voltmeter, which is a specialized device that can measure voltage accurately. However, with the advancement of technology, it has become possible to use microcontrollers like the Arduino to perform this task. The Arduino is a versatile microcontroller that can be used to perform a wide range of tasks, including voltage measurement.

**Features**

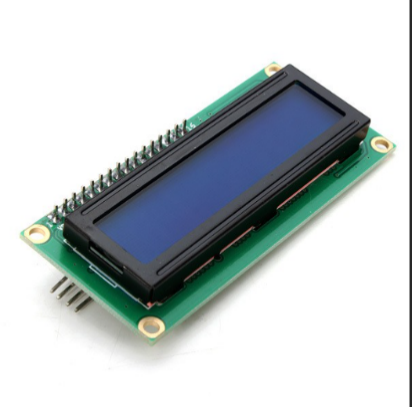
* Measures AC voltage using a ZMPT101B sensor module.
* Measures DC voltage using an analog pin connected to a 9V battery.
* Displays both AC and DC voltage readings on a 16x2 LCD screen.
* Adjustable sensitivity for the voltage sensor.
* Customizable display and formatting options.

**Components:**

For this project, we will need the following components:

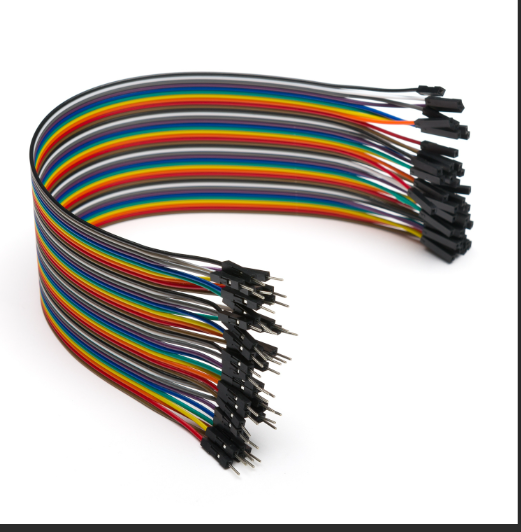
**Components**

* Arduino board (e.g., Arduino Uno)
* ZMPT101B voltage sensor module
* 16x2 I2C LCD display module
* Jumper wires
* Breadboard
* 9V battery and connector
* Potentiometer (for LCD contrast adjustment)

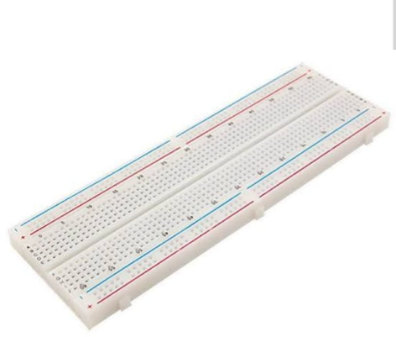


**16x2 LCD screen**

**Arduino Uno or compatible microcontroller board**

microcontroller board

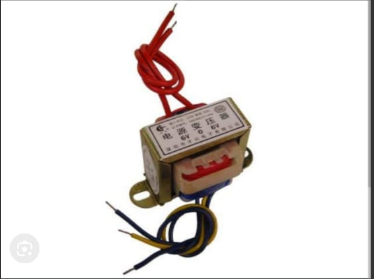
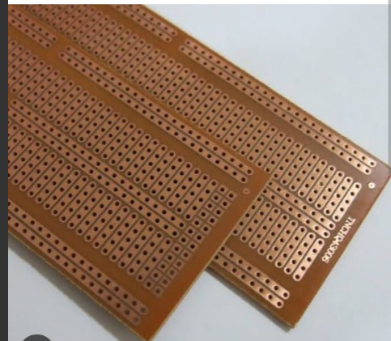
**Jumper wires**

****

**Breadboard**

**9V Battery**

**Dc Voltage sensor**

****

**Ac Voltage sensor**

**Transformer**

**Veroboard**

**Arduino Program:**

#include <ZMPT101B.h>

#define SENSITIVITY 500.0f

#include <LiquidCrystal\_I2C.h>

#define Sensorpin A1

LiquidCrystal\_I2C lcd(0x27, 16, 2);

// ZMPT101B sensor output connected to analog pin A0

// and the voltage source frequency is 50 Hz.

ZMPT101B voltageSensor(A0, 50.0);

void setup() {

Serial.begin(9600);

// Change the sensitivity value based on value you got from the calibrate

// example.

voltageSensor.setSensitivity(SENSITIVITY);

lcd.init();

lcd.backlight();

}

void loop() {

// read the voltage and then print via Serial.

float voltage = voltageSensor.getRmsVoltage();

Serial.print("Ac\_Voltage : ");

Serial.println(voltage);

//-----------------------------

int value = analogRead(Sensorpin);

double dcvoltage = map(value, 0, 1024, 0, 2500);

dcvoltage /= 100;

Serial.print("Voltage : ");

Serial.println(dcvoltage);

lcd.setCursor(0, 0);

lcd.print("AC: ");

lcd.print(voltage);

lcd.setCursor(0, 1);

lcd.print("DC: ");

lcd.print(dcvoltage);

delay(1000);

}

**Code Explanation:**

1. **Include Libraries:**
   * **#include <ZMPT101B.h>:** This line includes the library for the ZMPT101B voltage sensor.
   * **#include <LiquidCrystal\_I2C.h>:** This line includes the library for the LCD display using I2C communication protocol.
2. **Define Constants:**
   * **#define SENSITIVITY 500.0f:** This line defines the sensitivity value for the voltage sensor. The sensitivity value might need calibration based on the specific sensor used.
   * **#define Sensorpin A1:** This line defines the analog pin A1 as the pin for reading DC voltage.
3. **Initialize LCD Object:**
   * **LiquidCrystal\_I2C lcd(0x27, 16, 2);:** This line initializes the LCD object with the I2C address 0x27, and it specifies the dimensions of the LCD (16 columns and 2 rows).
4. **Initialize Setup Function:**
   * **void setup():** This function is called once when the Arduino is powered on or reset.
   * **Serial.begin(9600);:** Initializes serial communication with a baud rate of 9600 for debugging purposes.
   * **voltageSensor.setSensitivity(SENSITIVITY);:** Sets the sensitivity of the ZMPT101B sensor.
5. **Initialize Loop Function:**
   * **void loop():** This function is repeatedly executed after the setup function.
   * **float voltage = voltageSensor.getRmsVoltage();:** Reads the RMS (Root Mean Square) voltage from the ZMPT101B sensor for AC voltage.
   * **int value = analogRead(Sensorpin);:** Reads the analog voltage from the specified pin (A1) for DC voltage.
   * **double dcvoltage = map(value, 0, 1024, 0, 2500);:** Maps the analog reading to a voltage range (0-2500mV).
   * **dcvoltage /= 100;:** Converts the voltage to a readable format (in volts).
   * **lcd.setCursor(0, 0);:** Sets the cursor position on the LCD to the first row, first column.
   * **lcd.print("Ac: ");:** Prints "Ac: " on the LCD**.**
   * **lcd.print(voltage);:** Prints the AC voltage reading on the LCD.
   * **lcd.setCursor(0, 1);:** Sets the cursor position to the second row, first column.
   * **lcd.print("DC: ");:** Prints "DC: " on the LCD.
   * **lcd.print(dcvoltage);:** Prints the DC voltage reading on the LCD.
   * **delay(1000);:** Delays the loop execution for 1 second (1000 milliseconds).

**Installation**

1. Connect the components according to the provided circuit diagram.
2. Install the necessary libraries:

* ZMPT101B library for interfacing with the voltage sensor.
* LiquidCrystal\_I2C library for controlling the LCD display via I2C communication.

1. Upload the provided Arduino code to the Arduino board.

**Application:**

This project can be applied in various areas such as:

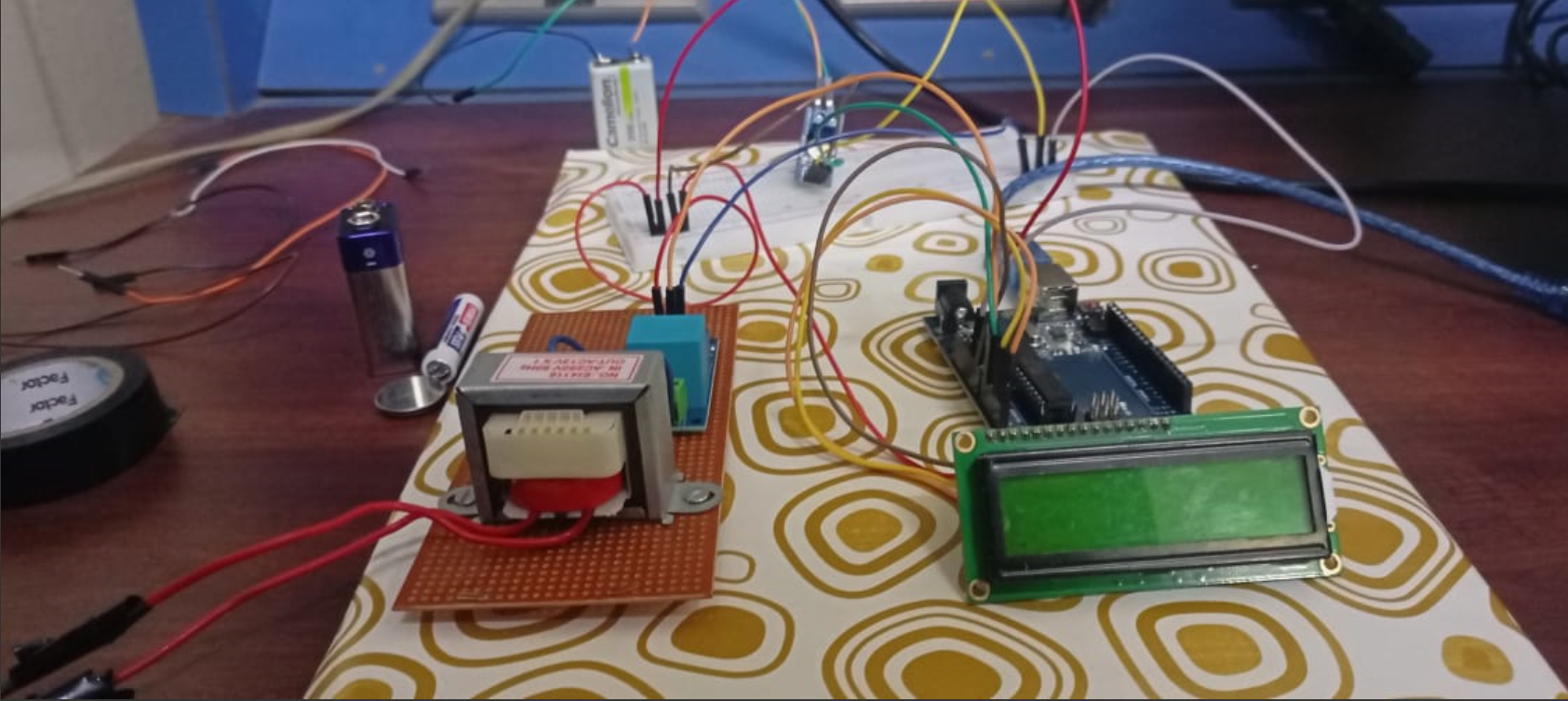
* It can be used as a monitoring tool for batteries to ensure the proper functioning of the devices.
* It can be used in devices like inverters, UPS to monitor the battery voltage levels.

This project can be used as a learning tool to understand the analog to digital conversion process in microcontrollers.

**Methodologies:**

* The 9V battery is connected to the breadboard using jumper wires.
* The 10kΩ potentiometer is connected to the breadboard to adjust the contrast of the LCD screen.
* The LCD screen is connected to the Arduino board using jumper wires.
* The Arduino program is uploaded to the board.
* The voltage is measured from the battery using the analog pin of the Arduino.
* The voltage is displayed on the LCD screen.

**Working Principle:**

The Arduino VoltLCD project uses voltage sensors or transducers that convert the AC/DC voltage into a voltage that can be read by the Arduino's analog input pins. The voltage sensors can be connected to the AC/DC power source that needs to be measured. The voltage sensors convert the voltage into a voltage that can be read by the Arduino's analog input pins.  
  
The Arduino uses the analog input pins to read the voltage and then converts the voltage reading into a digital value. The Arduino then processes this digital value and displays the voltage reading on an LCD screen. The LCD screen can be connected to the Arduino's digital pins, and the voltage readings can be displayed on the screen. 

**Usage:**

The Arduino VoltLCD project can be used in various applications where voltage measurement is necessary. For example, in home automation, the project can be used to measure the voltage of the power supply to different devices. In industrial control systems, the project can be used to monitor the voltage of motors, pumps, and other equipment. In scientific experiments, the project can be used to measure the voltage of electronic circuits.  
**Future Scope:**

This project can be extended to include additional features, such as:

* Displaying the battery level as a percentage.
* Adding a low battery indicator.
* Creating an alarm when the battery voltage drops below a certain level.

**Conclusion:**  
In conclusion, the Arduino VoltLCD project is a useful and versatile project that can be used to measure AC/DC voltage and display the results on an LCD screen using an Arduino microcontroller. The project has many potential applications in various industries and can be easily customized to fit specific project requirements. The project also demonstrates the versatility of the Arduino microcontroller and its ability to perform a wide range of tasks.