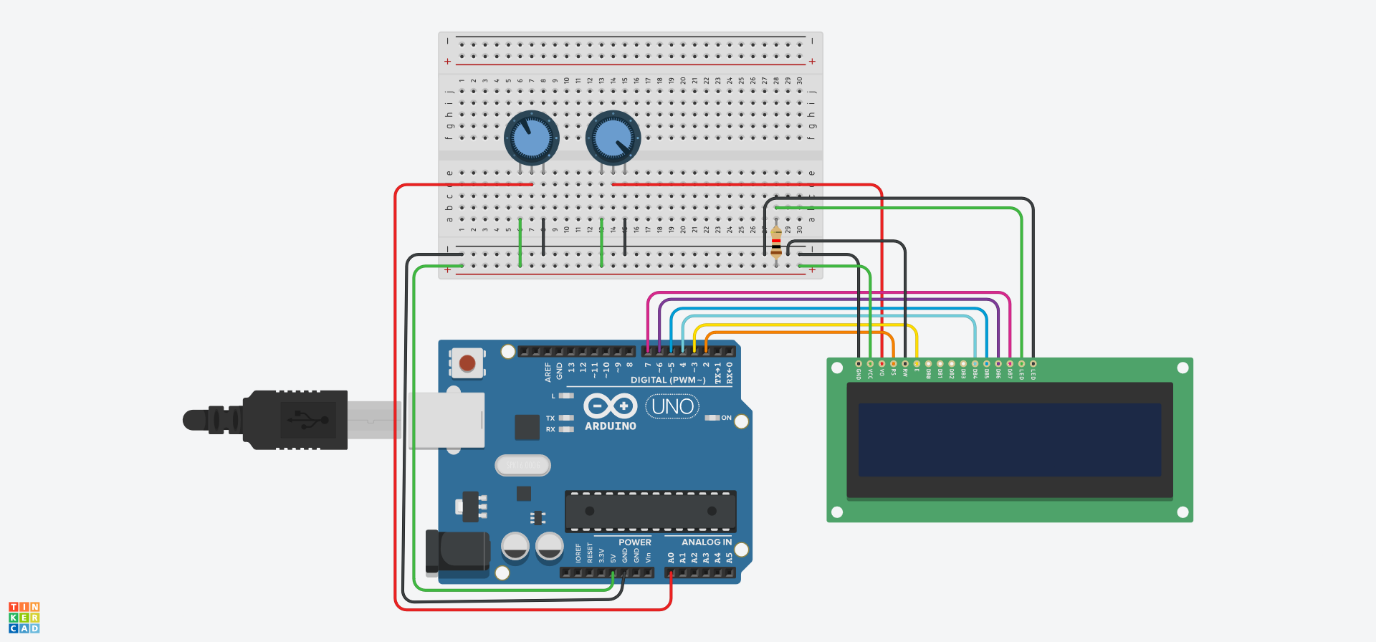
**Schematic diagram of the assembled circuit:**



**Program code in C++ using the Arduino IDE:**

#include <LiquidCrystal.h>

LiquidCrystal lcd**(**2**,** 3**,** 4**,** 5**,** 6**,** 7**);**

void setup**()** **{**

lcd**.**begin**(**16**,** 2**);**

**}**

void loop**()** **{**

double averageVoltage **=** findAverageValue**();**

int adcValue **=** analogRead**(**A0**);**

double voltageValue **=** convertAdcValueToVoltage**(**adcValue**);**

lcd**.**clear**();**

printVoltageValue**(**voltageValue**);**

printAvarageVoltageValue**(**averageVoltage**);**

delay**(**200**);**

**}**

double findAverageValue**()** **{**

double sumVoltage **=** 0**;**

**for** **(**int i **=** 0**;** i **<** 50**;** i**++)** **{**

int adcValue **=** analogRead**(**A0**);**

sumVoltage **+=** **((**double**)**adcValue **/** 1023**)** **\*** 5**;**

delay**(**20**);**

**}**

**return** sumVoltage **/** 50**;**

**}**

double convertAdcValueToVoltage**(**int adcValue**)** **{**

**return** **((**double**)**adcValue **/** 1023**)** **\*** 5**;**

**}**

void printVoltageValue**(**double voltageValue**)** **{**

lcd**.**setCursor**(**0**,** 0**);**

lcd**.**print**(**"U: "**);**

lcd**.**print**(**voltageValue**,** 2**);**

lcd**.**print**(**" [V]"**);**

**}**

void printAvarageVoltageValue**(**double averageVoltage**)** **{**

lcd**.**setCursor**(**0**,** 1**);**

lcd**.**print**(**"Usr: "**);**

lcd**.**print**(**averageVoltage**,** 2**);**

lcd**.**print**(**" [V]"**);**

**}**

**Description of the circuit operation, including a detailed explanation of each line of code and the behavior of the circuit during operation:**

This Arduino circuit measures an analog voltage input, converts it into a digital value using the Analog-to-Digital Converter (ADC), and displays both the real-time voltage and the average voltage over 50 measurements on an LCD screen. Below is the breakdown of the hardware and software components:

**Code Explanation**

LiquidCrystal lcd**(**2**,** 3**,** 4**,** 5**,** 6**,** 7**);**

* Creates an lcd object that is connected to the Arduino's digital pins 2, 3, 4, 5, 6, and 7. These pins are used to send data to the LCD.

void setup**()** **{**

lcd**.**begin**(**16**,** 2**);**

**}**

* In the **setup()** function, the lcd.begin(16, 2) initializes the LCD with a 16x2 configuration, meaning it has 16 characters per row and 2 rows in total.

void loop**()** **{**

double averageVoltage **=** findAverageValue**();**

int adcValue **=** analogRead**(**A0**);**

double voltageValue **=** convertAdcValueToVoltage**(**adcValue**);**

lcd**.**clear**();**

printVoltageValue**(**voltageValue**);**

printAvarageVoltageValue**(**averageVoltage**);**

delay**(**200**);**

**}**

* **loop():** This function runs continuously and performs the following:
* Calls findAverageValue() to calculate the average voltage from 50 readings.
* Reads the current voltage from pin A0 using the analogRead(A0) function, which gives a value between 0 and 1023.
* Converts the ADC value into an actual voltage using convertAdcValueToVoltage().
* Clears the LCD using lcd.clear() to refresh the display.
* Calls printVoltageValue() and printAvarageVoltageValue() to display the real-time and average voltage values on the LCD.
* Delays the next reading by 200ms to allow stable measurements and prevent flickering.

double findAverageValue**()** **{**

double sumVoltage **=** 0**;**

**for** **(**int i **=** 0**;** i **<** 50**;** i**++)** **{**

int adcValue **=** analogRead**(**A0**);**

sumVoltage **+=** **((**double**)**adcValue **/** 1023**)** **\*** 5**;**

delay**(**20**);**

**}**

**return** sumVoltage **/** 50**;**

**}**

* **findAverageValue():**
* This function calculates the average voltage from 50 readings.
* Inside a loop that runs 50 times, it:
* Reads the analog value from pin A0.
* Converts the ADC value to a voltage by using the formula ((double)adcValue / 1023) \* 5.
* Sums the converted voltage in sumVoltage.
* Introduces a 20ms delay between each reading to ensure stable measurement.
* Returns the average voltage (sum divided by 50).

double convertAdcValueToVoltage**(**int adcValue**)** **{**

**return** **((**double**)**adcValue **/** 1023**)** **\*** 5**;**

**}**

* **convertAdcValueToVoltage():** Converts the ADC value (0 to 1023) to a voltage (0 to 5V). The conversion is done using the formula:
* (adcValue / 1023) \* 5, where 1023 is the maximum ADC value and 5 is the reference voltage.

void printVoltageValue**(**double voltageValue**)** **{**

lcd**.**setCursor**(**0**,** 0**);**

lcd**.**print**(**"U: "**);**

lcd**.**print**(**voltageValue**,** 2**);**

lcd**.**print**(**" [V]"**);**

**}**

* **printVoltageValue():** Displays the real-time voltage value on the first row of the LCD. It shows the label “U: “ followed by the voltage with two decimal places and the unit “[V]”.

void printAvarageVoltageValue**(**double averageVoltage**)** **{**

lcd**.**setCursor**(**0**,** 1**);**

lcd**.**print**(**"Usr: "**);**

lcd**.**print**(**averageVoltage**,** 2**);**

lcd**.**print**(**" [V]"**);**

**}**

* **printAvarageVoltageValue():** Displays the average voltage value on the second row of the LCD. It shows the label “Usr: “ followed by the voltage with two decimal places and the unit “[V]”.

**Circuit Behavior**

1. **Reading the Analog Voltage:**
   1. The Arduino reads the analog signal from pin A0, which is a voltage between 0 and 5V.
   2. This analog signal is converted into a digital value (ADC) ranging from 0 to 1023.
2. **Displaying Real-time Voltage**:
   1. The current voltage is calculated using the ADC value and displayed on the first row of the LCD in volts (V). The voltage is updated every 200ms, providing a smooth, real-time update of the measured signal.
3. **Displaying Average Voltage:**
   1. The Arduino takes 50 voltage readings, with each reading spaced 20ms apart. The average of these 50 readings is calculated and displayed on the second row of the LCD.
4. **Refreshing the Display:**
5. The LCD is cleared at the beginning of each loop cycle to prevent overlapping characters. This ensures that the voltage readings are displayed cleanly and updated correctly.