GE107 Project Report:

Arduino-Based Oscilloscope

Group: 39
Submission Date: 12th May, 2025

Submitted to: Mr. Basant Subba
Completed under the guidance of: Mr. K R Ujjwal

Team Members:
Souhardya Saha (2023MEB1385)
Geethika (2023MEB1366)
T. Adil (2023MEB1389)
Prabhreet Kaur (2023MEB1368)
Kripalini Sethi (2023MEB1356)

1. Overview

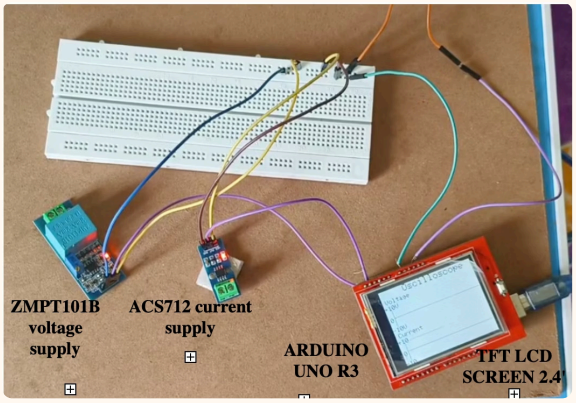
This project details the development of a low-cost, Arduino Uno-based dual-trace oscilloscope designed for educational and low-frequency diagnostic applications. It uses a ZmPT101B voltage sensor, an ACS712 current sensor, and a 2.4" TFT LCD to capture and display real-time AC/DC waveforms for both voltage and current. Aimed at overcoming the high cost of traditional oscilloscopes, this system provides an accessible tool for learning signal analysis and embedded systems, demonstrating effective real-time visualization within Arduino's hardware limitations.

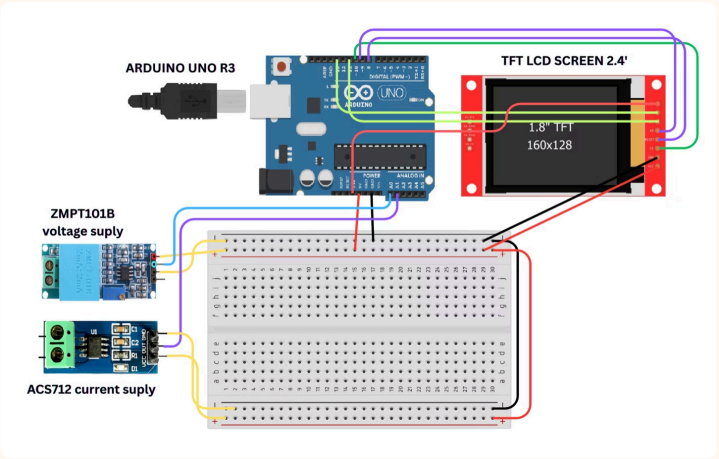
2. Objectives

Build a dual-trace oscilloscope using Arduino Uno to interface ZmPT101B (voltage) and ACS712 (current) sensors, displaying real-time AC/DC waveforms with a user-friendly interface including axes and gridlines.

3. System Overview

3a. Circuit





3b. Circuit Setup & Components

Components include:
Arduino Uno
TFT LCD (for waveform display)
ZmPT101B (AC voltage sensing)
ACS712 (current sensing)
breadboard
and jumper wires.

An Arduino Uno powers and controls the system, reading voltage via the ZmPT101B sensor (A5) and current via the ACS712 5A sensor (A4). A 2.4" ILI9341 TFT LCD, mounted on a shield, displays the waveforms. USB is used for power and code upload. Testing is done using a function generator and DC supply.

3b. Working Principle & Code Implementation

The Arduino reads analog voltage (A5) and current (A4) inputs using its ADC, mapping them to Y-coordinates on a TFT LCD—voltage in yellow (Y: 50–130) and current in cyan (Y: 160–240). Using Adafruit_TFTLCD and Adafruit_GFX libraries, the waveforms are drawn with analogRead() and map(), refreshing the screen when the x-coordinate exceeds display width. The code, written in C++ via Arduino IDE, samples at 50 Hz using a 20 ms delay.

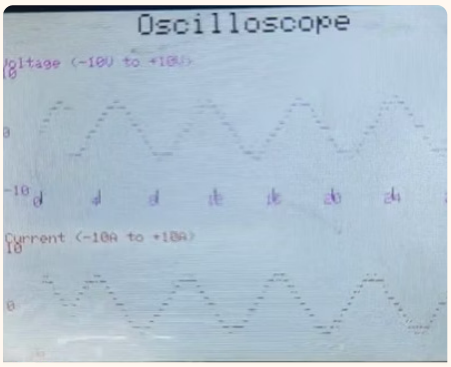
4. Results & Observations

Signal Testing Results:

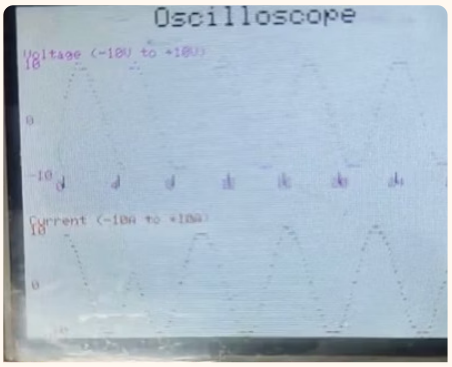
- DC Signal (5V): Flat, stable line
- 10V, 5V Sine Wave: Smooth, well-resolved waveform
- Square Waves: Clear transitions with minor noise

Observations:

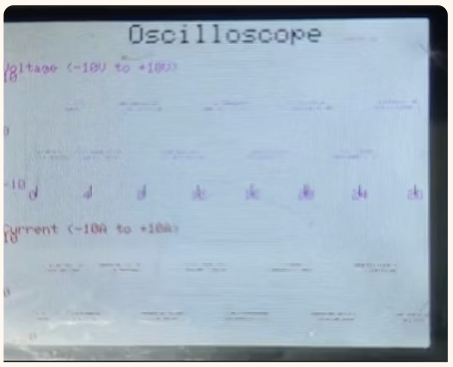
- Best results under 300 Hz
- Slight flicker during screen reset
- External noise affected current readings



5V Sin Wave



10 V Sin Wave



5V Square Wave

5. Challenges Faced:

Limited Sampling Rate

Arduino's ADC (~9.6 kHz) restricts high-frequency analysis, resulting in signal compression for frequencies >300 Hz.

Calibration

Sensors were not precisely calibrated, leading to approximate readings.

Noise

AC inputs introduced ripple, and current readings showed mild distortion.

Display Flicker

Full screen reset caused temporary flickering during refresh cycles.

Solutions: Optimized code for better performance at lower frequencies. Approximate calibration based on known test signals. Noise reduction techniques like careful grounding and wire placement were used.

6. Limitations:

- Signal Scaling: Calibration is approximate; more precise scaling would improve accuracy.
- Bandwidth: Effective for frequencies up to ~300 Hz.
- No Real-Time Triggering: Lack of pause or zoom functions limits in-depth analysis.
- TFT Refresh Delay: Some lag in fast signal transitions due to display limitations.

7. Applications

Educational Use

Ideal for students learning signal analysis, electronics, and embedded systems.

Low-frequency Diagnostics

Useful in basic diagnostic tasks for DIY electronics and hobbyist projects.

Circuit Debugging

Helps troubleshoot and visualize waveforms in electronic circuits.

Power Systems

Can be used to monitor and analyze low-frequency AC/DC signals in power systems.

8. Conclusion

The Arduino-based oscilloscope successfully displays real-time waveforms for educational purposes. Although limited by factors like sampling rate and sensor calibration, the project demonstrates the feasibility of creating an affordable, functional oscilloscope using microcontroller-based components. It serves as an excellent learning tool for signal processing, sensor interfacing, and embedded systems development.

9. Team Contributions

Team Member	Contribution Area
T. Adil	Code finalization, system integration, coordination
Souhardya Saha	Firmware development, debugging, oscilloscope logic
Geethika	Breadboard layout, write-up assistance
Prabhreet Kaur	Testing with AC/DC sources, documenting outputs
Kripalini Sethi	Sensor wiring, power and signal routing

Adil and Souhardya led the core logic and integration. Others contributed to build, testing, and documentation phases.

10. References

- Instructables. Arduino Oscilloscope Projects. https://www.instructables.com/
- Adafruit Learning System. TFT Display Guides. https://learn.adafruit.com/
- Arduino.cc Documentation. https://www.arduino.cc/