PROJECT ABSTRACT

Arduino Frequency Counter

An Arduino Uno-based digital frequency counter with LM358 input conditioning and an auto-ranging LCD readout, built as a student instrument with a buyer-run calibration procedure.

01. Title

Arduino Frequency Counter

02. Introduction / Background

Frequency measurement is one of the most common tasks in an electronics laboratory: oscillators, function generators, PWM outputs and clock sources all need their frequency verified. Commercial counters are accurate but expensive and often shared across many students. This project turns the problem into the lesson by having the student build the instrument itself — an Arduino Uno counting input pulses with its hardware timer, with an LM358 comparator stage conditioning real-world signals into clean edges.

03. Problem Statement

College labs rarely have enough frequency counters for every student, so learners use the instrument without understanding it. A project that builds a working counter around familiar hardware (Arduino, LM358, 16x2 LCD) converts a black-box instrument into a teaching platform covering timers, interrupts, signal conditioning and measurement error.

04. Objectives

- Build a working frequency counter around an Arduino Uno with a design target of 1 Hz to 2 MHz.
- Add an LM358-based input-conditioning stage so sine, triangle and noisy signals are counted reliably.
- Implement selectable gate time and a low-frequency period mode for accurate readings across the range.
- Provide an auto-ranging LCD readout and a documented buyer-run calibration procedure.

05. Proposed Methodology

The incoming signal is first squared up by the LM358 comparator with a threshold potentiometer, and a diode clamp protects the Arduino's 5 V input. The cleaned signal drives Timer1's external clock input, so edges are counted by hardware. Firmware opens a fixed gate (1 s or 0.1 s), reads the count, and computes frequency; below roughly 100 Hz it switches to measuring the period between edges. The result is auto-ranged and shown on the LCD.

06. System Architecture / Working

- Signal input → LM358 comparator stage (threshold adjustable) → diode clamp
- Squared signal → Arduino Timer1 external clock input → hardware edge counting
- Firmware gate timing → frequency computation (period mode below ~100 Hz)
- Auto-ranging logic → 16x2 LCD readout in Hz / kHz / MHz

07. Hardware / Software Requirements

- Arduino Uno (ATmega328P)
- LM358 dual op-amp, threshold potentiometer, clamp diodes
- 16x2 character LCD (HD44780)
- Breadboard/perfboard assembly, 5 V USB supply
- Arduino IDE (C/C++) firmware

08. Expected Outcomes

- A working counter reading 1 Hz to approximately 2 MHz (design target) with 1 Hz resolution at the 1 s gate.
- Clean counting of sine, triangle and noisy inputs after conditioning.
- A calibrated unit: the student documents deviation against a known reference frequency.
- Complete report, PPT and viva Q&A; covering timers, interrupts and measurement error.

09. Applications

- Verifying oscillator and function-generator frequencies in the college lab.
- Measuring PWM output frequency of motor-driver and microcontroller experiments.
- Checking clock sources and baud-rate-related frequencies during embedded projects.

10. Future Scope

- Add a second channel for simultaneous dual-frequency comparison.
- Log readings to an SD card or PC serial link for frequency-drift studies.
- Extend the conditioning stage with an attenuator for higher-voltage inputs.