**REPORT**

ON

**“Analog to Digital Conversion and Spectrum Analysis**

**of LM35 (Temperature Sensor) Signal”**

****

SUBMITTED

TO

**VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE**

**For the ADC SCE**

IN

**DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION**

**AY 2024-25**

**Semester IV**

BY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | PRN | Roll Number | Mobile Number | Email ID |
| Manas Kulkarni | 22311608 | 211058 | 7972470825 | manas.22311608@viit.ac.in |
| Samiksha Nalawade | 22311696 | 211063 | 9604107070 | samiksha.22311696@viit.ac.in |
| Rajlakshmi Desai | 22311732 | 211065 | 8605796792 | rajlakshmi.22311732@viit.ac.in |
|  |  |  |  |  |

**Class: S. Y. B. Tech Division: A Batch: A3**

**Faculty In-Charge / Evaluator**

**Dr. Pallavi Deshpande**

**Abstract**

This project explores how analog temperature signals from the LM35 sensor can be digitized using Arduino’s built-in Analog to Digital Converter (ADC) and analyzed both in time and frequency domains. By sampling the sensor’s output at regular intervals, the project showcases how real-world analog data can be captured, processed, and visualized. It also dives into the application of Fast Fourier Transform (FFT) to analyze the frequency content of the signals, helping to better understand important signal processing concepts like sampling, quantization, and aliasing. Through this project, the gap between theory and real-world application of ADC is effectively bridged.

**Introduction**

Analog signals are all around us – whether it's temperature, sound, or light. But digital systems, like microcontrollers and computers, can’t directly understand these analog signals. That’s where Analog to Digital Conversion (ADC) comes in. ADC plays a crucial role in converting real-world signals into a format that digital systems can understand and process.

In this project, we used the LM35 temperature sensor to generate a real-time analog voltage corresponding to temperature. This voltage is then sampled and digitized using the Arduino Uno's ADC. To take the analysis a step further, we performed time-domain and frequency-domain studies using FFT (Fast Fourier Transform) in Python. This helped us see how the signal changes over time and what frequency components are present, including the effects of quantization noise.

**Related Work**

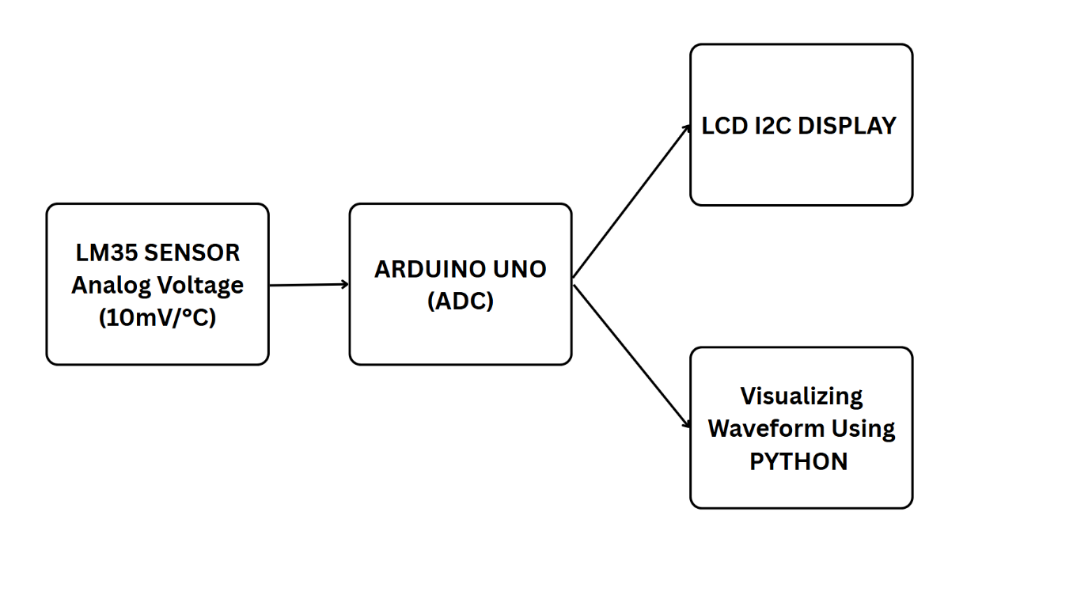
Analog-to-digital conversion is a foundational step in embedded systems where real-world signals from sensors like temperature, light, or sound need to be digitized for further processing. In this project, we adopted a similar approach by using the LM35 temperature sensor with Arduino’s built-in ADC to study temperature signals in both time and frequency domains.

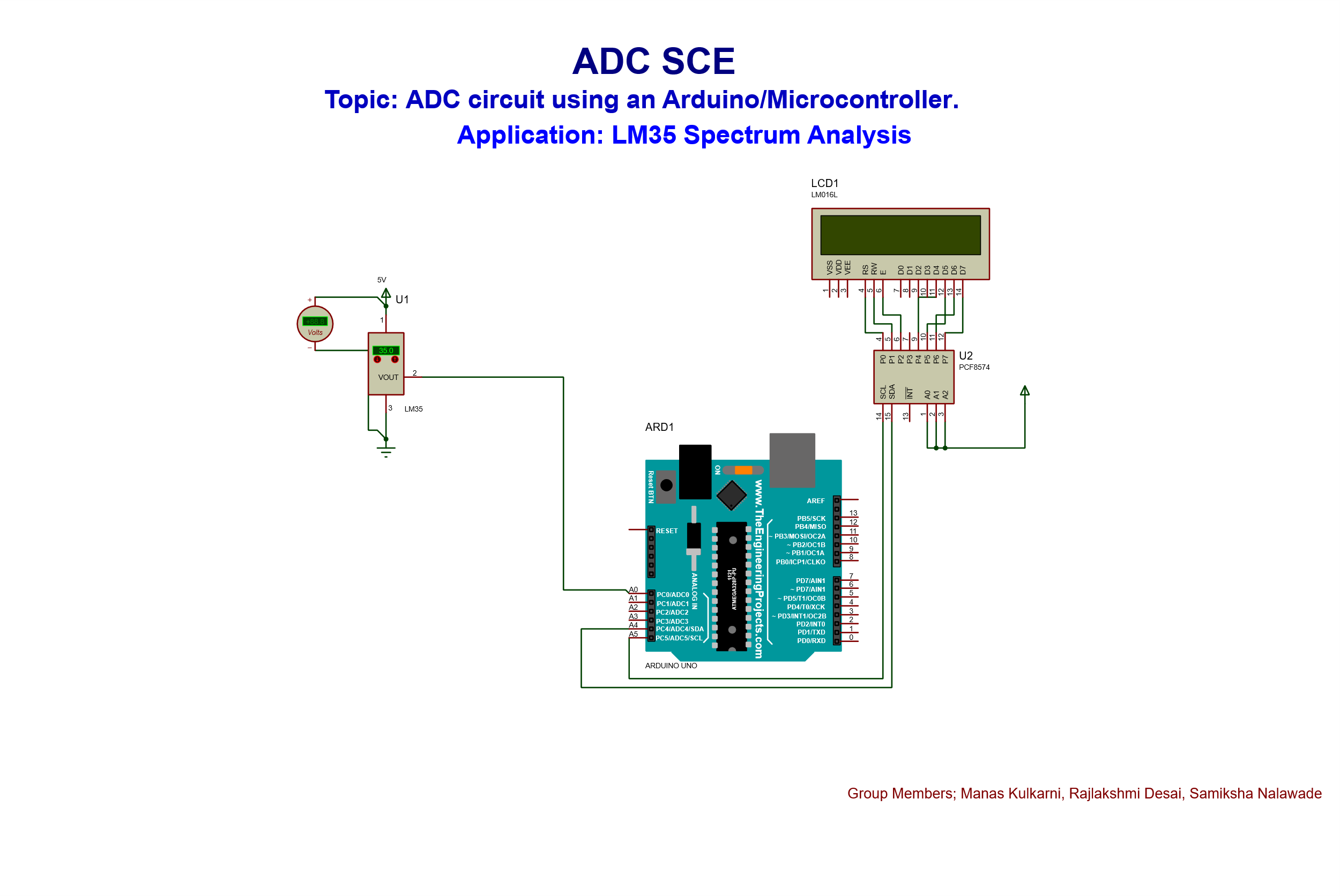
Several related studies align with this approach. Hussain (2014) designed a temperature measurement and control circuit focused on analog signal behavior, offering insight into signal conditioning prior to digitization. Bello et al. developed a Temperature Data Acquisition (TDAq) system for agroforest environments, involving sensor interfacing, ADC, and digital logging—closely related to our method of real-time data acquisition and processing.

Similarly, Otepola (2016) constructed an automatic room temperature controller using a microcontroller, showcasing how ADC-based systems can drive control mechanisms. Mohd Yusof (2008) extended ADC applications to wireless systems for remote temperature monitoring, demonstrating how digitized data can be analyzed or transmitted for real-time feedback.

Inspired by such real-world implementations, our project integrates these ideas into a cohesive system that samples analog signals from the LM35, digitizes them, and visualizes both time and frequency domain behavior using FFT—bridging hardware with signal processing principles.

**Block Diagram / Schematic**





Here’s how the system is structured:

1. **LM35 Temperature Sensor**Generates analog voltage proportional to temperature (10mV per °C).
2. **Arduino UNO**
   * Samples the analog voltage at a rate of 10 Hz.
   * Converts the analog signal into a 10-bit digital value using its internal ADC.
3. **LCD Display (Optional)**
   * Displays real-time temperature readings.
4. **Serial Monitor / CSV Output**
   * Records raw ADC values and quantized 8-bit data for analysis.
5. **Python (with SciPy & Matplotlib)**
   * Performs FFT on collected data.
   * Plots both time-domain and frequency-domain graphs.

**Design Considerations**

1. **LM35 Sensor**Chosen for its ease of use and precise linear output of 10mV per degree Celsius, which simplifies conversion and calibration.
2. **Arduino UNO**Equipped with a 10-bit ADC, meaning it can map input voltages from 0 to 5V into values ranging from 0 to 1023 – good enough for detecting small temperature variations.
3. **Sampling Rate (10 Hz)**The chosen rate satisfies the Nyquist criterion for the slow-changing nature of temperature data (expected max frequency < 5 Hz), avoiding aliasing while keeping the data size manageable.
4. **Quantization**For demonstration purposes, the 10-bit ADC output is scaled down to 8-bit (0–255), mimicking lower-resolution ADCs and showing quantization effects.
5. **Optional RC Filtering**A resistor-capacitor (RC) low-pass filter can be added to the LM35 output to reduce noise if present.
6. **Python Libraries**SciPy and Matplotlib were selected for their ease of handling numerical data and powerful visualization capabilities.

**Methodology / Working**

The system operates in the following steps:

1. **Signal Acquisition**  
   The LM35 continuously senses temperature and outputs an analog voltage.
2. **Analog to Digital Conversion**  
   Arduino reads this voltage every 100 milliseconds (10 Hz), converting it to a digital value using its ADC.
3. **Real-Time Display**  
   The temperature is optionally displayed on an LCD or printed to the serial monitor.
4. **Data Logging**  
   The ADC readings (both 10-bit and scaled-down 8-bit values) are saved in CSV format for offline analysis.
5. **Python-Based Signal Analysis**
   * The CSV data is loaded in Python.
   * FFT is applied to observe frequency components.
   * Time-domain plots show how temperature changes over time.
   * Frequency-domain plots reveal the presence of quantization noise and signal stability.

**Hardware Output**

A circuit board with wires and a display

AI-generated content may be incorrect.

**Results**

1. **Time Domain Observations:**

The analog signal recorded from the LM35 was smooth and consistent, showing slow variations typical of temperature changes.

The 8-bit quantized signal appeared as a step-wise waveform, highlighting the limitations of lower-resolution ADCs and the concept of quantization noise.

1. **Frequency Domain Observations (Using FFT):**

Both analog and digital signals exhibited strong low-frequency components (below 1 Hz), consistent with the slow-changing nature of temperature.

The digitally quantized signal included extra high-frequency noise, caused by quantization.

1. **Impact of Sampling Rate:**

The 10 Hz sampling rate was sufficient, as there were no significant high-frequency components in the signal.

No aliasing effects were observed, confirming the suitability of the sampling rate for this application.

A graph of different colored lines

AI-generated content may be incorrect.

Output Graphs (Matplotlib)

**Conclusion**

This project demonstrates how real-world analog signals can be efficiently captured and analyzed using digital systems. By combining an LM35 sensor, Arduino ADC, and Python FFT analysis, we were able to explore critical signal processing concepts hands-on.

The LM35 proved reliable for continuous temperature monitoring. Arduino’s 10-bit ADC offered sufficient resolution for capturing meaningful changes, and FFT analysis revealed the underlying frequency content and effects of quantization.

Ultimately, this project highlights how foundational concepts like sampling, quantization, and frequency analysis are applied in real-life systems, offering valuable insights for students and hobbyists working with sensor data and embedded systems.

**References**

1. Hussain, H. A. (2014). Temperature measurement and control circuit.
2. Bello, R. S., Nebo, U. E., & Onilude, M. A. Development of a Temperature Data Acquisition (TDAq) Device in Agroforest Environments.
3. OTEPOLA, A. A. (2016). DESIGN AND CONSTRUCTION OF AN AUTOMATIC ROOM TEMPERATURE CONTROLLER (Doctoral dissertation, DEPARTMENT OF MECHANICAL/BIOMEDICAL ENGINEERING, COLLEGE OF ENGINEERING, BELLS UNIVERSITY OF TECHNOLOGY).
4. Mohd Yusof, K. N. (2008). WIRELESS TEMPERATURE MONITORING SYSTEM.
5. Sensor, B. C. T., & Sensor, F. R. C. T. (1999). LM35 Precision Centigrade Temperature Sensors. *TX, USA: SNIS159H Datasheet, Texas Instruments*.