Digital Physics Laboratory

Manual

Prepared by Yu-Hsiang Huang
Date: 2024/12/31

Contents

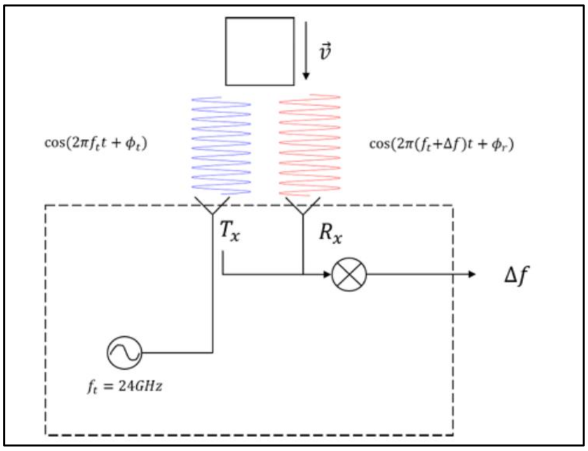
Introduction

The Digital Physics Laboratory Project leverages millimeter-wave radar technology to enable precise, non-contact motion measurement for physics experiments. This manual serves as a comprehensive guide to understanding, replicating, and utilizing the system effectively, with a focus on making it accessible for educators and students. This project is tailored to empower educators in creating engaging, hands-on experiences for students, bridging theoretical concepts with practical applications. Whether you are a teacher introducing basic mechanics or an advanced student exploring signal processing, this system provides a versatile platform to enhance learning.

System Overview

1. Millimeter-Wave Radar:

- **Model:** RFbeam Microwave GmbH K-LC7
- **Operating Frequency:** $f_t = 24GHz$
- **Function:** Captures motion data and outputs analog signals, which are processed to derive velocity and displacement.
- **Key Features:**
  - Compact design for easy integration.
  - High sensitivity for small motion detection.
- **Applications:** Non-contact motion tracking, vibration analysis, and frequency measurement.

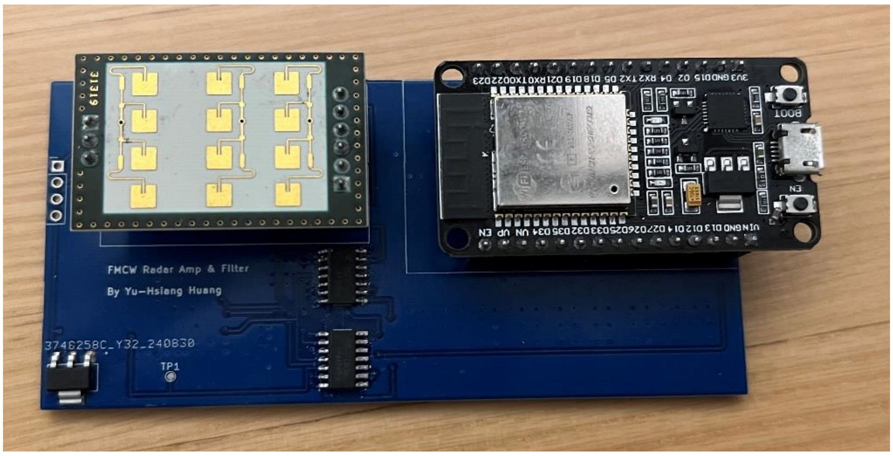


(Fig. 1: K-LC7 Module, Fig. 2: K-LC7 Schematic)

2. Signal Conditioning Circuit:

- **Purpose**: Amplifies and filters radar output signals to ensure compatibility with the ADC in the ESP32 microcontroller.
- **Design**:
  - Band-pass filter to isolate desired frequency ranges.

- **Software Tools:** Designed using KiCad for schematics and PCB layout.
- **Physical Details:**
  - Total PCB size: $10cm \times 5cm = 50cm^2$
  - Compact design allows easy integration into enclosures.
- **Applications:**
  - Noise reduction in signal acquisition.
  - Improved ADC accuracy and signal stability.



(Fig. 3: Signal Conditioning Circuit)

3.ESP32 Microcontroller:

- **Purpose**: Converts conditioned analog signals to digital data and transmits them to a computer via a serial port for analysis.
- **Key Features:**

12-bit ADC resolution for high-accuracy digitization.

Real-time data processing capabilities.

- **Software Support:**

Programmed using Arduino IDE for easy development and deployment.

- **Connections**:
  - Analog input: Receives conditioned signals from the radar.
  - Serial output: Sends digital data to a connected PC or device.
- **Applications**: Real-time motion analysis.

4. Python-Based Software

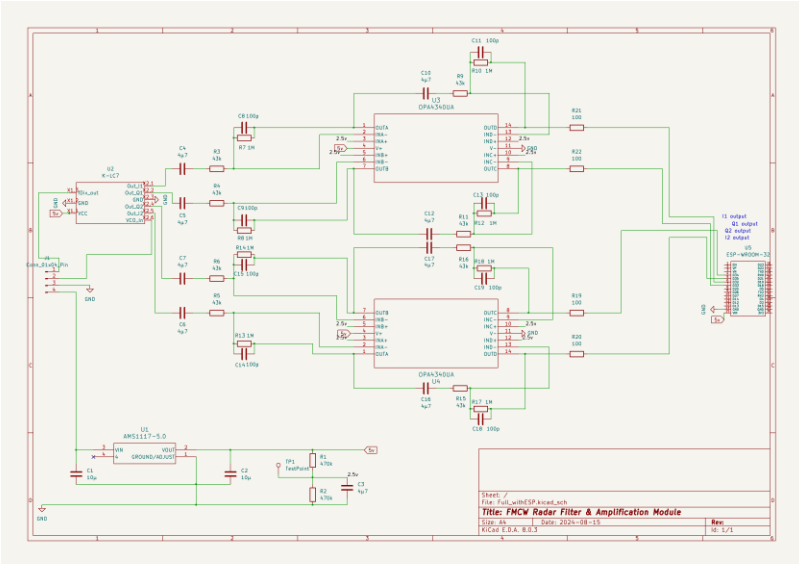
- Purpose: Processes and visualizes the collected data from the ESP32.
- **Key Features:**
  - **Signal Processing:**
    - High-pass filtering to remove low-frequency noise.
    - FFT (Fast Fourier Transform) for frequency domain analysis.
  - **Visualization:**
    - Time-domain plots for raw and filtered signals.
    - Frequency spectrum analysis for identifying dominant frequencies.
  - **Customization:**
    - Open-source Python scripts allow users to modify and expand functionality.
- **User Interface:**
  - Provides a command-line interface (CLI) for loading and analyzing data.
  - Optional GUI for real-time visualization (developed using Tkinter).
- **Applications:**
  - Understanding harmonic motion and resonance

Hardware Guide

Schematics

- The schematic (design_project.pdf) provides a detailed blueprint of the system. It includes:
  - **Radar Module**: Interfaces with the 24 GHz radar.
  - **Amplification Stage**: Features OPA4340 for low-noise signal conditioning.

- **Microcontroller Connections**: Configures ADC inputs and communication lines.
- **Power Management**: Converts 5V to 3.3V using AMS1117.



(Fig 4. Schematic)

PCB Design

- **Gerber Files**: Located in the hardware/PCB/ folder, ready for PCB fabrication.
- **3D Model**: Available in hardware/3D_model/ for visualization and enclosure design.
- **Assembly Notes:**
  - Solder components carefully to avoid thermal damage.
  - Use decoupling capacitors to stabilize power supply lines

Firmware Guide

Overview

The firmware is designed for the ESP32 microcontroller and performs:

1. **ADC Sampling**: Converts analog signals to digital data.
2. **Data Transmission**: Sends data wirelessly for analysis.

File Structure

- adc_sampling.cpp: Initializes and manages ADC sampling.
- esp32_config.cpp: Sets up ESP32 peripherals.

- main.ino: Main entry point for the program.

Deployment

1. Open the firmware folder in Arduino IDE.
2. Connect the ESP32 to your computer.
3. Select the appropriate board and port.
4. Upload the code to the ESP32

Software Guide

Python-Based Analysis

- The software is located in the software/ folder.
- Required dependencies are listed in requirements.txt.
- Key scripts:
  - gui.py: Provides a graphical interface for real-time/long-time analysis.
  - data_collector.py: Handles data acquisition and preprocessing.(CLI)

Running the Software

1. Install dependencies:

```
pip install -r requirements.txt
```

2.Execute the analysis script:

```
python gui.py
```

(Fig. 5: GUI demo)

Examples and Usage

Example Data

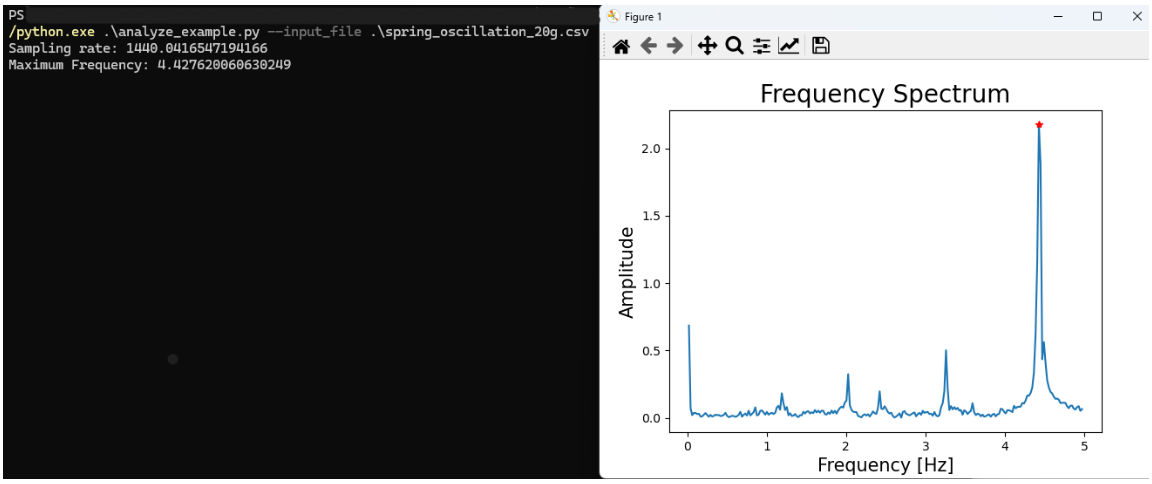
- Sample datasets are provided in the examples/ folder:
  - spring_oscillation_20g.csv: Data for 20g mass.
  - spring_oscillation_40g.csv: Data for 40g mass.

Visualization

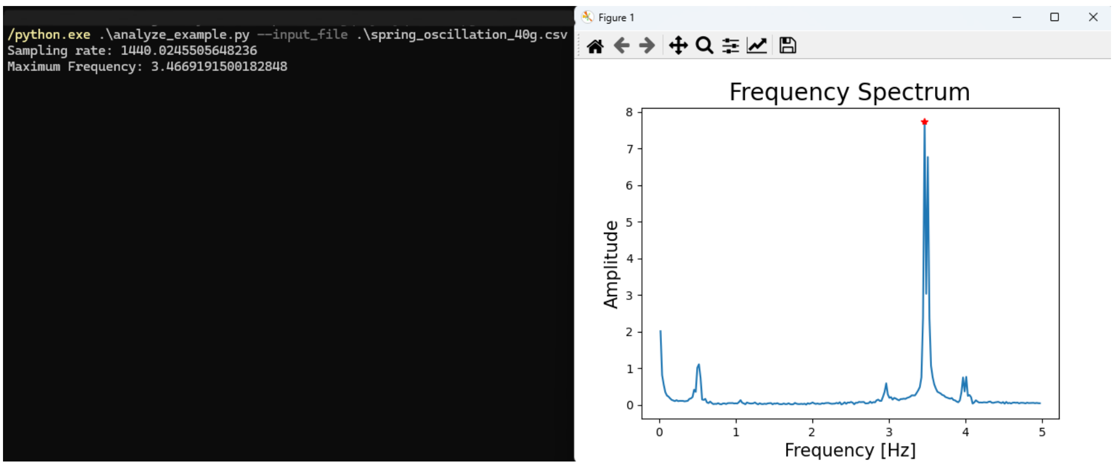
- Run the example analysis script:

```
python analyze_example.py --input_file 20g.csv
```

- **Output:**



(Fig. 6: Example output of 20g.csv)



(Fig. 7: Example output of 40g.csv)

Educational Applications

Enhancing Physics Labs

This system introduces students to cutting-edge measurement techniques, bridging traditional physics experiments with modern technology:

- **Harmonic Motion Experiments:**
  - Measure oscillation frequencies of different masses.
  - Analyze how the spring constant and damping factors affect motion.
- **Waveform Analysis:**
  - Teach students about filtering and noise reduction.
  - Visualize motion data in real-time.

Data Literacy

- **Data Collection:**
  - Show students how to collect accurate, repeatable measurements using radar and microcontrollers.
- **Data Analysis:**
  - Use Python scripts to process data, introducing students to concepts like FFT (Fast Fourier Transform) and time-domain analysis.

Cross-Disciplinary Applications

This system provides opportunities for collaboration between physics, electronics, computer science, and math courses:

- **Physics**:
  - Harmonic Motion: Analyze oscillations.
  - Wave Phenomena: Study Doppler effect, and millimeter waves.
- **Electronics**
  - Design, assemble, and test amplification and filtering circuits for radar signals.
  - Use KiCad to create and optimize compact PCB layouts.
  - Write firmware for the ESP32 to perform ADC sampling and serial communication.

- **Computer Science:**
  - Write and modify Python scripts to filter, analyze, and visualize data in real-time.
- **Math:**
  - Apply FFT to transform motion data from the time domain to the frequency domain.
  - Calculate mean, variance, and standard deviation to evaluate experimental accuracy.

Accessibility

- Open-source design ensures that schools with limited budgets can implement advanced measurement systems at a fraction of the cost of commercial solutions.
- The modular design allows educators to adapt the system to a variety of experiments, from oscillation studies to signal analysis.

Appendix

Bill of Materials (BOM)

Refer to hardware/bill_of_materials.csv for a complete list of components, including:

- Resistors, capacitors, and ICs.
- Radar module and ESP32 board.

Price estimate per system

Name	Footprint	Amount	Price(total, USD)
AMS1117	SOT223	1	0.5
OPA4340	D14	2	2.40
PinHeader	Conn_01x04_Pin	1	0.10
Capacitor	0402	19	0.38
Resistor	0402	22	0.67
ESP32	ESP32-dev	1	9.14
K-LC7	K-LC7	1	57.28