2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Paper ID #50135

Classroom Approaches to RF Education with Visual-Based Modulation Recognition Labs

Mr. Erwin Karincic, Virginia Commonwealth University

Erwin Karincic received B.S. and M.S. degrees in Computer Engineering from Virginia Commonwealth University (VCU) in 2020 and 2021, respectively. He is currently pursuing a Ph.D. degree from Virginia Commonwealth University. He is an experienced security researcher with focus on reverse engineering and exploit development. An avid learner in many different fields, his research interests are cyber security, reverse engineering, exploit development, Internet of Things, software defined radio, antenna analysis, and design. He currently holds 28 cyber security certifications.

Dr. Lauren Linkous, Virginia Commonwealth University

Dr. Lauren Linkous is a Postdoctoral Research Associate specializing in Machine Learning, electromagnetics, and automation. She received her Ph.D. in Engineering from Virginia Commonwealth University. She has extensive teaching experience across electrical engineering and computer science, including courses in circuits, medical device security, electromagnetics, and machine learning. She has developed curricula on RF optimization, AI/LLM applications, and wireless communication, guiding students through hands-on labs with industry-standard tools. She also holds an M.S. in Computer Science and dual B.S. degrees in Electrical Engineering and Physics from VCU.

Dr. Erdem Topsakal, Virginia Commonwealth University

Classroom Approaches to RF Education with Visual-Based Modulation Recognition Labs

Abstract

The increasing complexity of modern wireless communication systems has created a challenge in educating the next generation of Radio Frequency (RF) engineers. With a shortage of hands-on experiential opportunities and limited applied demonstration of electromagnetic wave propagation, students often struggle to connect theoretical knowledge with practical implementation. While the fundamental theory of modulation schemes is essential, ensuring that students are ready for the workforce necessitates hands-on experience with actual signal analysis and system development. Classroom sets of RF equipment are often cost prohibitive for educational institutions, making it difficult to offer individualized instruction for students. To bridge this gap, it is imperative that students have opportunities to practice RF concepts with affordable and accessible hardware. This paper introduces a comprehensive solution to these challenges by incorporating an interactive educational platform using commonly available hardware and open-source software to teach digital modulation recognition and signal analysis through progressive hands-on modules.

The platform leverages low-cost components including ESP32 microcontrollers, CC1101 radio transceivers, and commercial Software Defined Radios (SDRs) to create an engaging learning environment where students can learn modulation schemes. A central receiver station operated by the educator includes a programmable LED array that provides visual confirmation when students successfully transmit valid packets based on provided parameters. The total cost of a single basic student station is under $75, while an advanced configuration with the HackRF One SDR remains accessible at $400 per station.

In this paper, we detail the development and delivery of five comprehensive lab exercises covering 2-FSK, 4-FSK, GFSK, MSK, and ASK modulation schemes. Each lab is designed with adjustable complexity levels, allowing educators to tailor the difficulty based on course objectives. We provide three distinct hardware configurations: (1) an ESP32/CC1101 unit to program transmitters and learn modulation parameters, (2) an RTL-SDR to capture and analyze signals and recreate them using the ESP32/CC1101, and (3) a HackRF One SDR for full receive and transmit capability for advanced SDR modules. We provide preconfigured labs meant for streamlined integration into a broad range of wireless communications courses. These labs allow students to identify modulation schemes, determine data rates and frequency deviations, and implement their own transmissions, illuminating a specific LED on the educator's receiver. Educators can easily customize packet fields to create unique challenges and prevent simple replay attacks. Assessment methodologies range from basic packet generation to complex signal analysis, supporting various learning objectives. The purpose of this paper and the supporting material is to provide educators with an effective supplement to traditional RF equipment. The practical labs enable students at all levels to gain hands-on experience required for the modern workforce.

1 Introduction

Wireless communication systems have evolved rapidly over the past two decades, changing how information is transmitted across the electromagnetic spectrum. Digital modulation techniques are foundational to modern wireless systems, offering improvements in spectral efficiency, reliability, and security over analog predecessors [1, 2, 3]. These techniques are found everywhere, from cellular networks using orthogonal frequency-division multiplexing (OFDM) to Internet of Things (IoT) sensors using basic frequency shift keying [4].

Despite widespread use in commercial and industrial applications, a gap persists between theoretical instruction and practical implementation in engineering education. Students in communications courses often develop strong mathematical foundations in modulation theory [5]. However, transitioning from mathematical foundation to physical implementation is difficult. Virtual experiments and simulations alone cannot fully bridge this theory-practice gap, as students engaged in hands-on hardware instruction develop significantly stronger conceptual understanding than those limited to software-based instruction [6, 3, 7]. Real electromagnetic propagation introduces complexities like multipath fading, interference, and noise that are difficult to convey through simulation.

High equipment costs compound this educational challenge. Hands-on RF education traditionally requires expensive instruments like spectrum analyzers and signal generators. In some cases, the annual fees for licenses, even for industry standard equipment, make purchasing this equipment to be unfeasible. These costs place individualized RF laboratory experiences beyond the reach of many educational institutions. As a result, RF and wireless communications courses often rely heavily on simulation tools such as MATLAB or software-based demonstrations, which, while valuable for developing the necessary theoretical understanding, cannot fully replicate the experience of working with actual electromagnetic signals and RF hardware [8, 9, 3].

Implementing Software Defined Radio (SDR) technology as part of RF laboratory experiences opens opportunity for new ways to teach RF concepts affordably. Low-cost SDR receivers such as the RTL-SDR, combined with software tools like MATLAB, Simulink, and GNU Radio, can create complete desktop SDR design environments suitable for undergraduate instruction [10]. SDR platforms including the RTL-SDR, HackRF One, ADALM-PLUTO, and USRP have democratized access to radio frequency experimentation by replacing expensive analog hardware with flexible software implementations [11, 12]. Deploying these systems presents challenges, such as signal processing complexity and hardware interfacing, but structured guidance can address these hurdles [13]. These platforms enable students to capture, analyze, and generate RF signals while developing skills directly applicable to the modern wireless communications industry.

Integrating SDR platforms into digital communication curricula has a positive impact on student learning outcomes. Hands-on SDR exercises lead to significantly better performance in modulation recognition and signal analysis compared to simulation-only approaches [8]. Comprehensive surveys of SDR platform capabilities have identified the RTL-SDR and HackRF One as particularly suitable for undergraduate instruction due to their low cost and broad frequency coverage [12]. Practical guidance has emerged for educators selecting SDR platforms; comparing cost, capabilities, and pedagogical effectiveness across multiple platforms to help institutions make informed decisions [14].

Remote laboratory platforms have also expanded access to RF education beyond a traditional classroom environment. Platform-agnostic remote laboratory models enable scalable, asynchronous online laboratory experimentation using SDR technology, demonstrating that geographically distributed students can effectively engage with RF hardware through carefully designed remote interfaces [15]. Similar approaches for IoT education have established frameworks for delivering hands-on wireless experiences to students regardless of their physical location [16]. This model can be expanded to utilizing SDR hardware for teaching digital modulation techniques and supplementing other course instruction. Automatic modulation classification techniques developed over the past two decades provide the theoretical foundation for many of the signal analysis exercises employed in modern SDR-based curricula [17].

Addressing the need for a hands-on, RF education platform that is practicable, obtainable, and adaptable is a multi-fold problem. The accessibility of affordable hardware platforms for students alone does not guarantee effective outcomes for RF education [3]. Consideration needs to given towards the method of instruction, the availability of material, and existing university programs. Structured laboratory curricula that leverage affordable platforms also remain scarce, compounding the issue of developing comprehensive lab material on what is often a time and budget sensitive schedule. There exists a direct need for educators to be able to access complete laboratory platforms that combine hardware, firmware, and exercises into cohesive educational experiences with clear learning outcomes.

This paper introduces an interactive platform designed to address these challenges through progressive, hands-on laboratory modules focused on digital modulation recognition and signal analysis. The platform combines hardware including ESP32 microcontrollers, CC1101 radio transceivers, and commercial SDR devices. We introduce open-source firmware and pre-configured laboratory exercises with associated hardware. As part of the platform design, we provide students with visual feedback through a programmable LED array on the receiver station, which confirms successful transmissions. This real-time feedback is designed to encourage and engage students throughout the laboratory sequence.

The presented platform supports three distinct hardware configurations accommodate different resources. The introductory configuration, requiring only an ESP32 microcontroller and CC1101 transceiver module, enables students to learn modulation transmission at a cost of approximately $30 per station [18]. Intermediate configurations add an RTL-SDR receiver for signal capture and analysis, while advanced configurations employ the HackRF One SDR for complete transmit and receive capability. This tiered approach allows educators to select the appropriate level of complexity according to course objectives and adapt to existing course expectations.

Five pre-configured laboratory modules cover digital modulation schemes: Binary Frequency Shift Keying (2-FSK), 4-level Frequency Shift Keying (4-FSK), Gaussian Frequency Shift Keying (GFSK), Minimum Shift Keying (MSK), and Amplitude Shift Keying/On-Off Keying (ASK/OOK). Each lab includes configurable parameters for preamble, sync word, address, and data fields, allowing educators to create unique challenges. Optional extensions like checksums, Manchester encoding, data whitening, and forward error correction provide additional parameters for advanced courses.

The platform presented in this work provides a complete set of laboratory instruction and experimentation for RF and wireless communications courses. These materials benefit educators seeking to supplement simulation-based courses with hands-on experiences, as well as students developing practical skills for the wireless communications industry. The key contributions of this paper are as follows:

1. We provide open-source receiver and transmitter firmware with a centralized configuration system, enabling educators to customize lab parameters without extensive firmware development.
2. We provide a tiered hardware architecture with three configurations ranging from approximately $30 to $400 per station, allowing institutions to select appropriate complexity based on budget and course objectives.
3. We develop a GNU Radio flowgraph with eight visualization windows and a decision tree methodology for visual modulation classification, enabling students to identify and distinguish between 2-FSK, 4-FSK, GFSK, MSK, and ASK signals.
4. We deliver five complete laboratory modules with configurable parameters and optional extensions (checksums, Manchester encoding, data whitening, FEC), providing educators with ready-to-deploy exercises adaptable to introductory through advanced courses.

This paper is organized as follows: Section 2 describes the hardware and software architecture of the platform. Section 3 presents the five lab modules with detailed descriptions and learning outcomes. Section 4 addresses the student learning outcomes and applicability towards standard evaluation. Section 5 discusses the solution materials available for educators. Section 6 concludes this paper.

2 System Setup

The platform consists of a central unit operated by the educator and multiple student stations. We use commercial hardware and open-source libraries to ensure accessibility.

2.1 Hardware Architecture

Table 1 lists the commercially available components and their approximate prices at the time of this publication. Although the hardware supports both transmission and reception, we designate student stations as transmitters (TX) and the educator's unit as the receiver (RX) for validation. In the presented work, SDRs serve as supporting devices for signal analysis or transmission and can be included to meet additional course goals. A custom assembly of cost effective parts means that SDRs are not necessary for all labs.

All TX/RX units use an ESP32 microcontroller and CC1101 Radio Transceiver as the primary hardware [18]. The educator's RX unit includes a 10-LED array for visual confirmation of successful transmissions when receiving a radio transmission. The type of success is correlated with specific LEDs based on the lab activity. Student setups may include an SDR (e.g. RTL-SDR, HackRF One) for analysis alongside the ESP32/CC1101 unit. A second instructor ESP32/CC1101 unit can be designated to transmit repeating signals to introduce demodulation activities in addition to the core material. Alternatively, should SDR hardware be readily accessible, a substitution can be made to handle both demodulation and transmission on a single device.

Table 1: Bill of Materials

Component	Specification	Unit Cost
ESP32 DevKit v1 (2x)	Low Cost Microcontroller	$6
CC1101 Module (2x)	315/433/868/915MHz transceiver	$9
LEDs (10x)	5mm, 20 mA	$0.02
300Ω resistors (10x)	1/4W, 5%	$0.01
Breadboard (1)	Solderless Breadboards	$3
Jumper wires (pack)	male/female and male/male	$3
RTL-SDR v3 (1x)	24–1766 MHz RX	$38
	Subtotal (without optional item)	**$74.30**
HackRF One (Optional)	30 MHz–6 GHz TX/RX	$320
	Grand Total (with optional item)	**$394.30**

The circuit for the primary RX unit is configured as shown in Fig. 1, including the additional LED array not configured with the student units. Fig. 2 shows the assembled RX unit on a breadboard, demonstrating the physical connections between the ESP32, CC1101 transceiver, and the 10-LED array used for visual feedback.

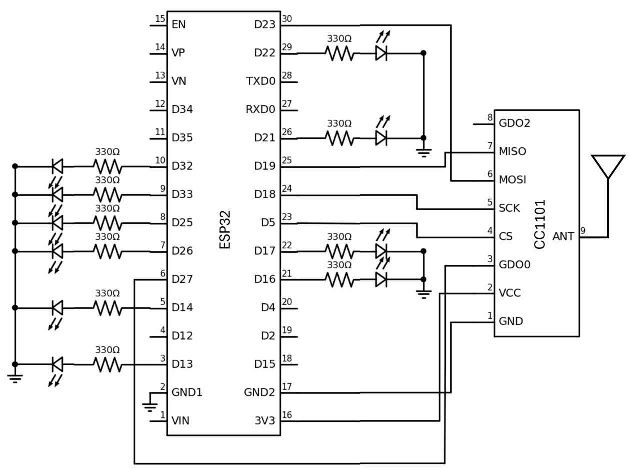

Figure 1: RX circuit diagram with ESP32 microcontroller (left) and CC1101 transceiver (right). An LED array is connected to both sides of the ESP32 [18].

2.2 Software Architecture

The firmware was developed in C++ for the ESP32 using the Arduino framework and PlatformIO. The software includes a CC1101 radio library, a receiver application, and a transmitter application. All firmware components are released open-source and with basic setup instructions at [19].

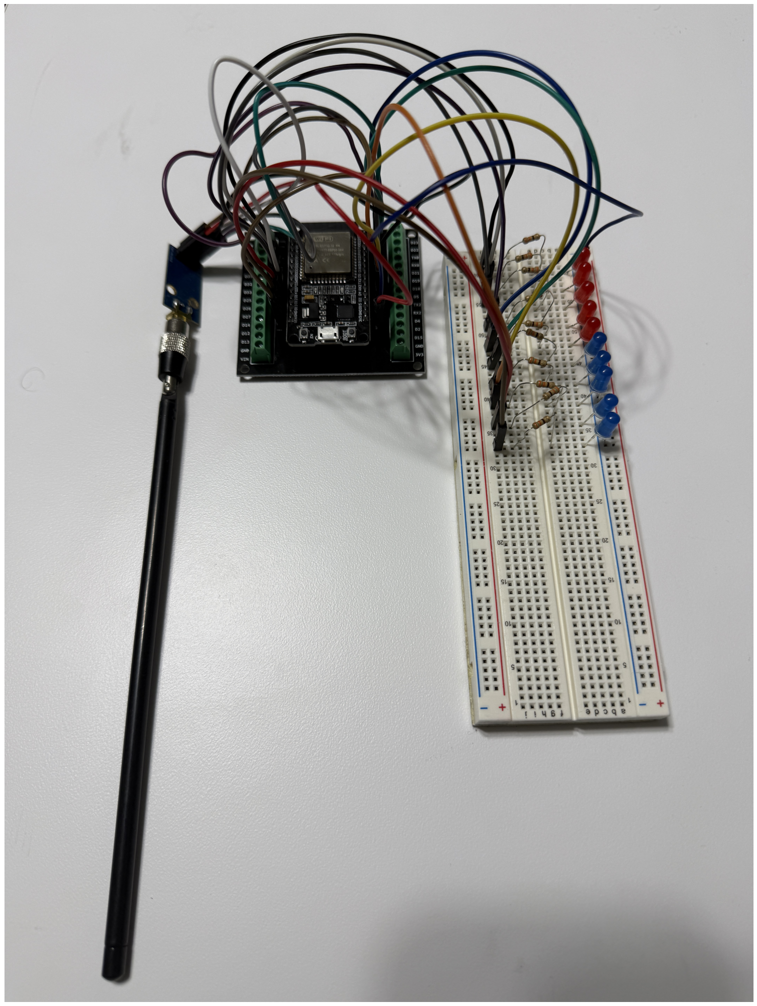

Figure 2: Assembled RX unit showing the ESP32 DevKit v1 microcontroller connected to the CC1101 transceiver module and 10-LED array on a breadboard. This configuration serves as the educator's central validation station, providing visual confirmation when students successfully transmit valid packets.

2.2.1 CC1101 Radio Library

The CC1101 library, originally from [20], provides an object-oriented interface for the transceiver. The library abstracts register manipulation complexity, exposing methods for setting modulation parameters. The `Radio` class encapsulates transceiver functionality with methods like `setModulation()` and `setFrequency()`.

Supported modulation modes include Binary FSK (2-FSK), Gaussian FSK (GFSK), Amplitude Shift Keying/On-Off Keying (ASK/OOK), 4-level FSK (4-FSK), and Minimum Shift Keying (MSK). The library enforces specific data rate ranges for each modulation type.

Frequency configuration supports the CC1101's three operational bands: 300-348 MHz, 387-464 MHz, and 779-928 MHz. The library calculates frequency register values from a floating-point specification. It also calculates optimal exponent and mantissa values for frequency deviation.

Automatic calibration ensures accurate frequency generation during state transitions. The CC1101's output power is configurable from -30 dBm to +12 dBm. Other features include configurable sync words, preamble lengths, and hardware address filtering. The library exposes signal quality metrics like Received Signal Strength Indicator (RSSI) and Link Quality Indicator (LQI).

2.2.2 Receiver Firmware

The receiver firmware operates on the educator's ESP32 station and implements continuous packet monitoring with live, visual LED feedback. This is a custom and adjustable firmware created for this work, hosted at [19]. To use, the firmware must be flashed to the ESP32 after any configuration updates. The firmware uses FreeRTOS to run radio reception in a dedicated task, keeping serial command processing responsive. It is compatible with both single student TX and multiple student TX configurations.

2.2.3 Transmitter Firmware

Similar to the instructor station receiver firmware, the transmitter firmware for this work is custom for the provided lab material. The transmitter firmware provides an interactive interface for students to send control packets via serial over USB. The serial interface accepts single-character commands: digits 0-9 and letters a-f send control values 0-15, 'r' initiates raw packet entry mode, and 'h' displays the current configuration summary.

Raw packet mode allows transmission of arbitrary bytes. This is useful for verifying parameters determined through signal analysis. The transmitter automatically calculates checksums when configured to match the receiver's settings. A status LED on the receiver station confirms valid transmission.

The platform offers implementation flexibility through firmware customization. Educators may provide the complete transmitter firmware to students, allowing focus on modulation parameters and signal analysis. Alternatively, students can receive only the hardware and CC1101 library, requiring them to develop the firmware. This approach encourages comprehensive engagement with the library API and the transmission process.

2.2.4 Session Configuration System

A key design feature of the platform is the centralized configuration system that allows educators to customize lab parameters with minimal code changes. Configurable parameters are consolidated at the top of the firmware files. Educators can modify lab exercises to create challenges by first selecting a base lab activity with the `SESSION_CONFIG` definition and then by changing individual lab parameters for custom configurations. Changing only the `SESSION_CONFIG` will select a different base lab exercise.

The configuration system exposes the following parameters for modification, each of which directly affects the lab challenge and can be adjusted independently:

SESSION_CONFIG: Selects a pre-configured session preset (1-5).

MODULATION_TYPE: Specifies the digital modulation scheme, which includes options of 2FSK, 4FSK, GFSK, MSK, and ASK/OOK.

DATA_RATE: The symbol rate in kilobaud (kBaud). Valid ranges depend on the modulation type: 0.6-500 kBaud for 2-FSK, 0.6-250 kBaud for GFSK and ASK/OOK, 0.6-300 kBaud for 4-FSK, and 26-500 kBaud for MSK. Lower data rates produce narrower bandwidth signals that are easier to capture but may require longer observation times. Higher data rates challenge students with wider bandwidth signals requiring appropriate receiver filter configuration.

DEVIATION: The frequency deviation in kilohertz (kHz), applicable to FSK-based modulation schemes. This parameter controls the frequency separation between symbol levels and directly affects both the modulation index and the required receiver bandwidth. For 2-FSK and GFSK, the deviation specifies the frequency offset from carrier for logic 1 (with logic 0 at the opposite offset). For 4-FSK, adjacent symbols are separated by the deviation value, with the total frequency span being three times the deviation. For MSK, this parameter is set to 0 because the hardware automatically calculates the deviation as one quarter of the data rate. For ASK/OOK, deviation is not applicable and is ignored.

RX_BANDWIDTH: The receiver filter bandwidth in kilohertz (kHz). This parameter must be set wide enough to accommodate the signal bandwidth plus any frequency offset or drift. Setting the bandwidth too narrow causes signal loss; setting it too wide admits additional noise and interference. Typical values range from 58 kHz to 812 kHz, with 150 kHz being appropriate for most lab configurations at 1.2 kBaud.

PREAMBLE_BYTE: A single byte (0x00-0xFF) that serves as an application-level preamble in the packet structure. This byte is the first field students must match in their transmissions. Changing this value between class sections prevents solution sharing, as packets with incorrect preambles are silently rejected.

ADDRESS_VALUE: A byte value providing basic station addressing. This value refers to the LED address on the instructor station, not a communication channel address. Each session preset uses a different address and educators can assign unique addresses to individual students or groups for assessment purposes.

USE_CHECKSUM: A boolean flag (true/false) that enables or disables checksum validation. When disabled (the default for introductory exercises), packets are validated based only on preamble, LED address, and correct packet length. When enabled, the receiver computes a checksum over the packet data and rejects packets where the transmitted checksum does not match or if transmission was corrupted. This adds complexity appropriate for intermediate to advanced exercises.

CHECKSUM_TYPE: When checksums are enabled, this parameter (0, 1, or 2) selects the algorithm: 0 for byte summation (SUM), 1 for exclusive-or (XOR), or 2 for CRC-8. Each algorithm has different error detection properties that students learn through implementation.

The LED mapping array provides additional customization flexibility. This 16-element array maps control values to 10 physical LEDs, with -1 indicating unmapped values. Educators can modify this array to create challenges where students must discover the mapping relationship through experimentation, or where certain control values produce no visible output. The default mapping leaves several control values unmapped, creating intentional gaps that students will need to explore until they understand the mapping system.

Configuring a custom lab is straightforward and requires little modification. To do this, educators need to open the firmware source file, locate the configuration section (clearly marked with comment banners), and modify the desired parameters. For example, to create a more challenging version of Lab 1 with checksum validation, an educator would change `USE_CHECKSUM` from `false` to `true` and select the checksum algorithm. The transmitter firmware must be updated with matching parameters for students to successfully communicate with the receiver. This firmware update is not reflected in the primary interactive student interface.

This design philosophy ensures that educators can adapt the platform to their specific course objectives without requiring deep firmware development expertise. All protocol-defining parameters are exposed at the top of the source file with descriptive comments, making customization accessible to instructors with basic programming knowledge.

The receiver provides serial output for debugging, including packet bytes, RSSI, LQI, and validation status. A debug mode shows all received packets for troubleshooting.

Packet validation has multiple stages. First, it verifies packet length. Then, it validates the preamble and address. If enabled, it computes and compares the checksum. Only valid packets trigger LED state changes.

2.2.5 Packet Structure

The packet structure is:

```
[PREAMBLE][ADDRESS][CONTROL][CHECKSUM(optional)]
```

The preamble byte allows the receiver to prepare for the incoming message. The address field identifies stations, allowing multiple student TX groups to operate simultaneously and to follow common patterns of communication protocols. The control field maps to LED output states. The optional checksum field provides error detection.

Hardware-generated preamble bits and a 16-bit sync word precede each packet. Sync mode requires a full 16-bit match. The CC1101 handles these fields automatically.

2.3 Hardware Configurations

There are three student configurations available in the provided material.

1. **Configuration 1 (ESP32/CC1101 Only):** Students use the ESP32/CC1101 unit to learn modulation transmission.
2. **Configuration 2 (RTL-SDR + ESP32/CC1101):** Students capture signals using the RTL-SDR, analyze them with open-source software tools, and recreate transmissions using the ESP32/CC1101.

3. **Configuration 3 (HackRF One):** Students use the HackRF SDR for complete signal analysis and transmission.

2.4 SDR-Based Visual Modulation Analysis

In Configuration 1 exercises, students are given all necessary signal parameters for transmission. In Configuration 2 and 3 exercises, students identify the modulation scheme and extract signal parameters from captured transmissions before recreating packets. A GNU Radio Companion flowgraph enables this detection based on signal signatures.

2.4.1 Analysis Flowgraph Architecture

The provided GNU Radio flowgraph captures at 433.92 MHz with a 1 MHz sample rate, routing received I/Q samples through a low-pass filter (100 kHz cutoff) before branching into multiple parallel analysis paths [19]. Eight visualization windows display different signal characteristics:

Envelope Time Display: A `Complex to Mag` block extracts the signal envelope for time-domain display. ASK/OOK signals show clear amplitude variations between on and off states. FSK signals produce a flat, constant envelope. This is the primary method for identifying ASK modulation.

Amplitude Distribution Histogram: Displays the distribution of envelope values. ASK/OOK signals produce two distinct peaks corresponding to the on and off power levels. Constant-envelope modulations (all FSK variants) produce a single peak. This histogram confirms ASK identification from the envelope time display.

Instantaneous Frequency Time Display: A `Quadrature Demod` block extracts instantaneous frequency deviation from the carrier. The time-domain display reveals transition characteristics: 2-FSK produces sharp square-wave transitions between two frequency levels, GFSK shows smooth rounded transitions, and MSK exhibits sinusoidal frequency changes. This display differentiates 2-FSK from GFSK and MSK.

Frequency Distribution Histogram: Displays the distribution of instantaneous frequency values. The number of peaks indicates the modulation level: two peaks indicate 2-FSK, GFSK, or MSK whereas four peaks indicate 4-FSK. Adjusting the `Quadrature Demod` gain is critical for resolving all four peaks in 4-FSK signals. Insufficient gain compresses the peaks, potentially causing misclassification as 2-FSK.

Phase Trajectory Display: A `Complex to Arg` block extracts the instantaneous phase. This display distinguishes GFSK from MSK when the frequency histogram shows two peaks. GFSK exhibits a S-curved shape caused by Gaussian filtering phase transitions. MSK displays linear ramp phase changes with sharp direction changes. The phase plot provides definitive identification between these two modulations.

Spectrum Display: An FFT-based `frequency sink` shows the signal spectrum. The number of spectral peaks can confirm FSK level (two for binary FSK, four for 4-FSK). Spectral shape also differs: GFSK and MSK have smoother roll-off with reduced side lobes compared to standard FSK. ASK shows a double-sideband amplitude modulation spectrum centered on the carrier.

Waterfall Display: Provides a time-frequency view of signal activity. Students use this to confirm the carrier frequency and observe signal bandwidth over time. The waterfall also helps identify intermittent transmissions.

Constellation Display: Shows the complex signal in I/Q space. ASK signals appear as points along a line (varying amplitude, constant phase). FSK signals trace a circle (constant amplitude, varying phase). While less critical for classification, the constellation provides additional confirmation.

2.4.2 Visual Classification Decision Tree

Students classify signal modulation by checking displays in the GNU Radio flowgraph in sequence:

1. **Check envelope time display:** If amplitude varies between distinct on/off levels, the signal is ASK/OOK. Confirm with that the amplitude histogram is showing two peaks. If envelope is constant, proceed to step 2.
2. **Check frequency distribution histogram:** Count the peaks. Four peaks indicate 4-FSK. Two peaks indicate a binary FSK variant (2-FSK, GFSK, or MSK). Adjust the `Quadrature Demod` gain slider to ensure peaks are properly resolved.
3. **Check instantaneous frequency time display:** Examine the transition shape between frequency levels. Sharp square-wave transitions indicate standard 2-FSK. Smooth, rounded transitions indicate GFSK or MSK. Proceed to step 4 to distinguish between them.
4. **Check phase trajectory display:** Linear ramp phase changes indicate MSK. S-curved, rounded phase transitions indicate GFSK.

After identifying the modulation type, students extract key signal parameters. The frequency histogram peak spacing reveals the frequency deviation. Measuring transition timing in the time-domain displays allows symbol rate estimation. The waterfall display confirms the carrier frequency. With these parameters, students configure their ESP32/CC1101 transmitter (Configuration 2) or generate SDR transmissions in GNU Radio (Configuration 3).

2.4.3 Tunable Parameters

The flowgraph includes real-time adjustable parameters via GUI sliders:

RX Gain: Adjusts SDR receive gain from 0 to 70 dB. Students learn to set gain high enough for adequate signal level while avoiding saturation or excessive noise amplification.

Quadrature Demod Gain: Scales the frequency demodulator output. This parameter is critical for the frequency distribution histogram. For 4-FSK signals, incorrect gain compresses or spreads the four frequency levels, making them difficult to distinguish. Students adjust this slider until all frequency levels appear as distinct peaks.

3 Lab Modules

This work provides five lab modules covering different modulation schemes, each corresponding to a session configuration preset. The configuration system in Section 2 allows educators to customize preamble bytes, map LED address values, modulation types, data rates, frequency deviations, and checksum settings. The LED mapping array can deliberately unmap certain values for discovery challenges.

The labs are designed to work with all three hardware configurations described in Section 2. A consistent packet structure lets students focus on modulation-specific parameters. Each session configuration preset defines the necessary radio parameters.

Configuration 1 provides explicit modulation parameters for programming the ESP32/CC1101 units. Configuration 2 requires signal capture and analysis using an SDR. Configuration 3 offers a comprehensive experience, using an SDR for both analysis and transmission.

3.1 Concept and Implementation

The adaptable configuration and tiered approach for configuring complexity allows the labs to supplement a broad range of instruction. This enables educators to adjust complexity based on existing course objectives and learning outcomes. Students engaged in hand-on hardware-based instruction develop stronger conceptual understanding and intuition for RF concepts compared to theory-driven approaches [8, 3]. Structured laboratory curricula that combine low-cost hardware with clear learning objectives enable scalable, effective RF education across institutions. The following labs are designed to integrate affordable SDR platforms into communications curricular and address three critical needs: 1. bridging the theory-practice gap via exposure to real-world propagation effects, 2. providing individualized and customization hands-on access without expensive, proprietary equipment, and 3. promote students developing practical skills that are directly applicable to the modern workforce [3, 7, 12]. The following lab modules progress through a linear sequence that builds in complexity, introducing concepts that culminate in comprehensive exercises exposing students to different modulation schemes, signal parameter extraction, and successful transmissions towards targeted addresses.

3.2 Lab 1: 2-FSK Modulation

3.2.1 Session Configuration

Lab 1 uses a session configuration that sets the modulation type to 2-FSK. The configuration defines a unique preamble byte and address value. Checksums are disabled by default to simplify the initial exercise.

3.2.2 Description

Binary Frequency Shift Keying (2-FSK) introduces digital frequency modulation. 2-FSK transmits binary data by shifting the carrier frequency between two discrete values. One frequency represents a logical 0 and the other a logical 1. The separation between these states is the frequency deviation.

Operating at 433.92 MHz in the ISM band, this lab allows for low-power unlicensed transmission. The default configuration specifies a 1.2 kbps data rate and 25 kHz frequency deviation. This produces a high modulation index, ensuring detection even with moderate noise. A 150 kHz receiver bandwidth accommodates frequency drift.

The handout provides preamble, address, and sync word values. Checksums are disabled to reduce complexity. The receiver validates packets against these parameters and illuminates the corresponding LED upon receiving a valid packet.

In Configuration 1, students receive all modulation parameters and must write C++ code to configure the CC1101 registers. The challenge lies in translating abstract parameters into specific register values. The CC1101 library simplifies this with high-level API calls:

```
radio.setModulation(MOD_2FSK);
radio.setFrequency(433.92);
radio.setDataRate(1.2);
radio.setFrequencyDeviation(25.0);
```

Iterating through control field values illuminates mapped LEDs, proving the ability to generate valid packets. The LED mapping array determines which control values activate physical LEDs, with some values intentionally unmapped to create discovery challenges.

Configuration 2 introduces signal analysis using the visual modulation analysis GNU Radio flowgraph. The educator transmits a reference 2-FSK signal, and students capture it using the RTL-SDR. Using the frequency histogram, students observe two distinct peaks corresponding to the mark and space frequencies, confirming 2-FSK modulation. Sharp square-wave transitions in the instantaneous frequency display distinguish this from GFSK or MSK. Students measure peak separation to determine frequency deviation, then use Universal Radio Hacker (URH) [21] to demodulate and extract the preamble, address, and control bytes before recreating the transmission with their ESP32/CC1101 unit.

Configuration 3 follows a similar analysis workflow using the HackRF One for both reception and transmission. Students can optionally generate their transmissions directly in GNU Radio rather than using the ESP32/CC1101, providing deeper SDR experience.

3.2.3 Learning Outcomes

After completing this lab, students will understand the principles of frequency shift keying modulation, including how carrier frequency, frequency deviation, and signal spectrum relate mathematically. Students will gain practical skills configuring the CC1101 transceiver for 2-FSK transmission through register manipulation.

The packet construction process teaches students to generate valid communication packets with correct preamble sequences, sync words for packet boundary detection, and address fields for access control. Transmitting different control values to activate LEDs reinforces the connection between transmitted data and physical outcomes.

For students completing Configuration 2 or 3, additional outcomes include SDR-based signal capture experience and using spectral analysis tools to determine unknown signal parameters.

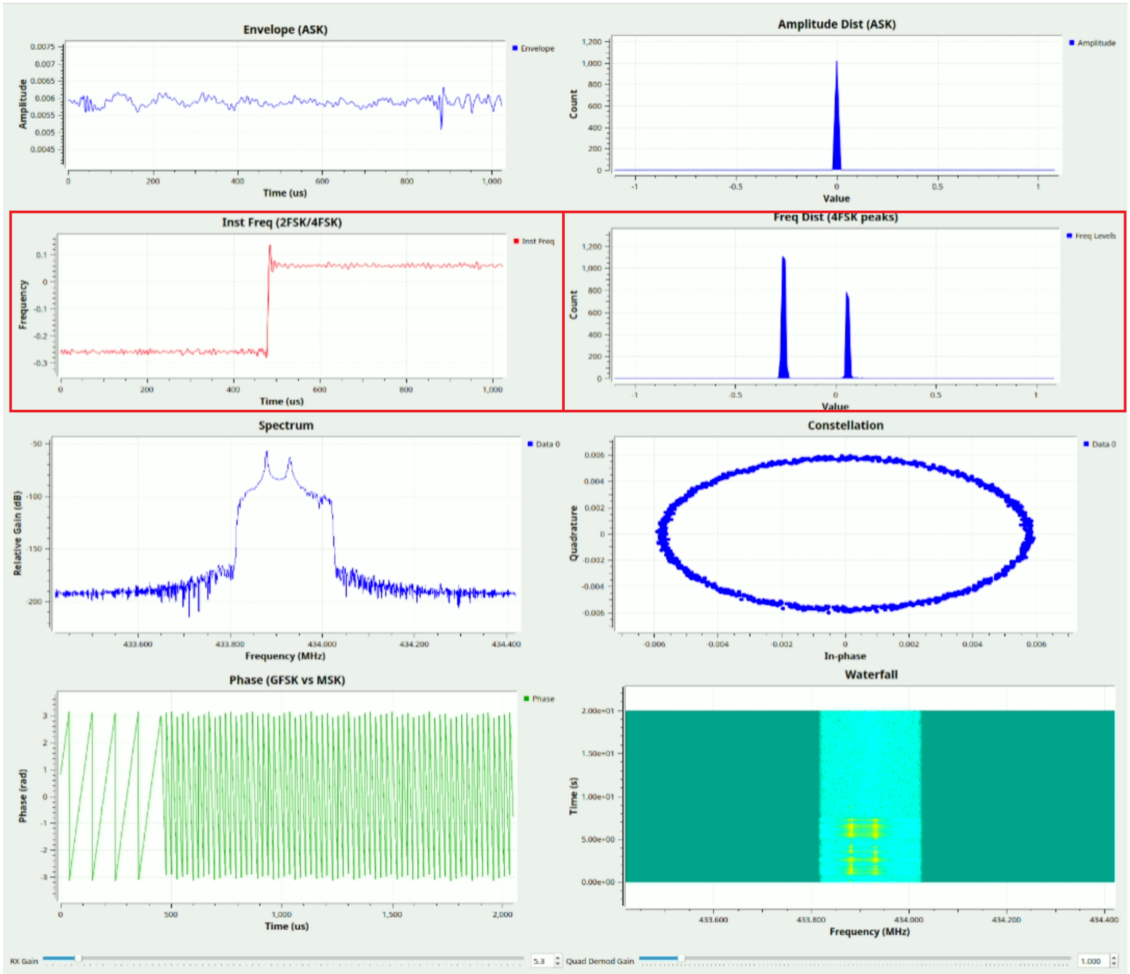

Figure 3: SDR visualization of 2-FSK modulation showing (a) frequency histogram with two distinct peaks, (b) instantaneous frequency display with sharp square-wave transitions between frequency states, (c) waterfall display showing the characteristic dual-frequency signature, and (d) flat envelope confirming constant-amplitude FSK modulation.

These students will learn signal demodulation and protocol reverse engineering applicable to RF engineering work.

3.3 Lab 2: 4-FSK Modulation

3.3.1 Session Configuration

Lab 2 uses a session configuration with the modulation type set to 4-FSK. A distinct preamble byte and address value are assigned. The 25 kHz deviation means the total frequency span across all four symbol levels is 75 kHz (three times the deviation), with symbols at $-3\Delta f$, $-\Delta f$, $+\Delta f$, and $+3\Delta f$ from the carrier. Note that Manchester encoding cannot be used with 4-FSK due to CC1101 hardware constraints.

3.3.2 Description

4-level Frequency Shift Keying (4-FSK) uses four discrete frequency levels. Each symbol encodes two bits, representing 00, 01, 10, and 11. This doubles data throughput for a given symbol rate compared to 2-FSK.

This lab also operates at at 433.92 MHz. The configuration uses a specific, set preamble and address. With a 1.2 kBaud data rate and 25 kHz deviation, the four frequency levels are spaced at $-3\Delta f$, $-\Delta f$, $+\Delta f$, and $+3\Delta f$ from the carrier. Inner frequencies are one deviation unit from the carrier, while the outer frequencies are three units away.

In Configuration 1 exercises students receive modulation parameters and configure their transmitters. 4-FSK doubles the bit rate but reduces frequency separation between adjacent symbols, which increases noise susceptibility and requires precise frequency discrimination.

Configuration 2 and 3 exercises challenge students to distinguish between 2-FSK and 4-FSK signals using the visual analysis flowgraph. The critical indicator of which modulation is in use is the frequency distribution histogram: 4-FSK signals produce four distinct peaks rather than two. Students must adjust the `Quadrature Demod` gain slider to properly resolve all four frequency levels. Insufficient gain compresses the peaks together, potentially causing signal modulation to be misclassified as 2-FSK. Once identified, students measure the frequency spacing between adjacent peaks to determine the deviation parameter.

3.3.3 Learning Outcomes

After this lab, students will understand multi-level frequency modulation and how additional symbol levels improve bandwidth efficiency while reducing noise margins. The lab builds skills in CC1101 configuration for 4FSK modulation, including setting modulation format bits and understanding how deviation register values affect frequency spacing. Students learn the spectral characteristics of 4-FSK signals and how they differ from binary FSK in the time and frequency domains.

For Configurations 2 and 3, the lab provides students experience in modulation classification. Students learn to identify 4-FSK signals from their spectral signatures and to extract the additional parameters needed for successful demodulation and signal recreation.

3.4 Lab 3: GFSK Modulation

3.4.1 Session Configuration

Lab 3 uses a session configuration with the modulation type set to GFSK. Unique preamble and address values are assigned. The CC1101 enforces a maximum data rate of 250 kBaud for GFSK (versus 500 kBaud for 2-FSK) due to Gaussian filtering overhead. The checksum type is preset to XOR, allowing educators to quickly enable checksum validation for a more challenging variant, if that is desired.

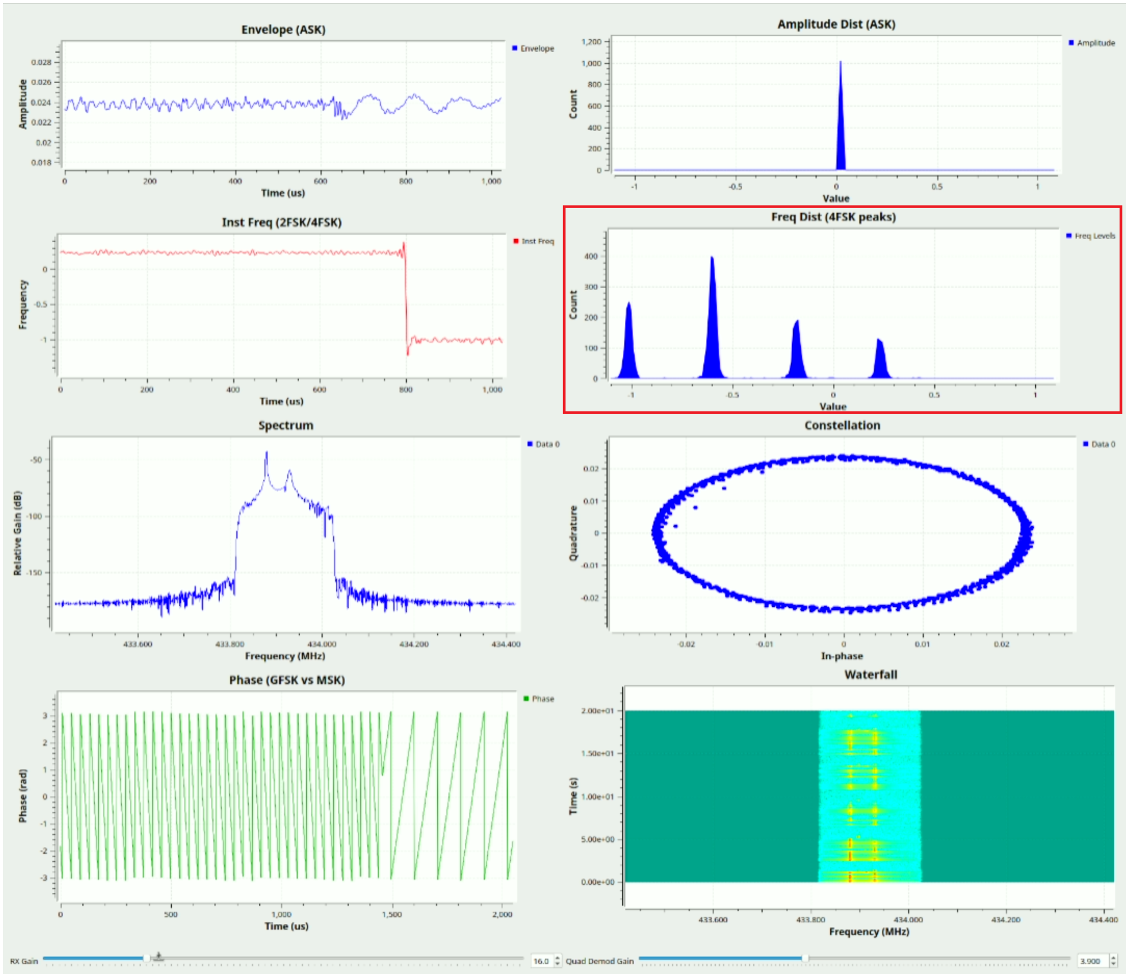

Figure 4: SDR visualization of 4-FSK modulation showing (a) frequency histogram with four distinct peaks, (b) instantaneous frequency display showing four discrete frequency levels with transitions between all symbol states, (c) waterfall display with the wider bandwidth characteristic of 4-FSK, and (d) flat envelope confirming constant-amplitude modulation.

3.4.2 Description

Gaussian Frequency Shift Keying (GFSK) applies a Gaussian low-pass filter to the modulating data signal before frequency modulation. This filtering smooths abrupt frequency transitions at bit boundaries, replacing instantaneous jumps with gradual changes. The result is reduced spectral side lobes and improved adjacent channel performance.

Operating at 433.92 MHz, the configuration specifies a 1.2 kBaud data rate and 25 kHz deviation. GFSK is used in Bluetooth Classic. Outside of the provided lab, students can observe Bluetooth signals with SDR equipment and compare spectral characteristics to their GFSK transmissions. While this platform uses the 433 MHz band, the modulation principles remain identical to Bluetooth's 2.4 GHz operation.

In Configuration 1, students configure their units for GFSK. They understand how Gaussian filtering affects spectral characteristics and the time-domain waveform. Comparing GFSK transmissions with Lab 1's 2-FSK signals reveals spectral differences.

Configuration 2 and 3 exercises require distinguishing GFSK from 2-FSK and MSK. The frequency histogram shows two peaks for all three schemes. However, the phase trajectory display differentiates them: 2-FSK shows linear segments with abrupt slope changes at symbol boundaries, GFSK exhibits S-curved transitions, and MSK displays continuous linear ramps. Comparing waterfall displays highlights GFSK's smoother spectral rolloff.

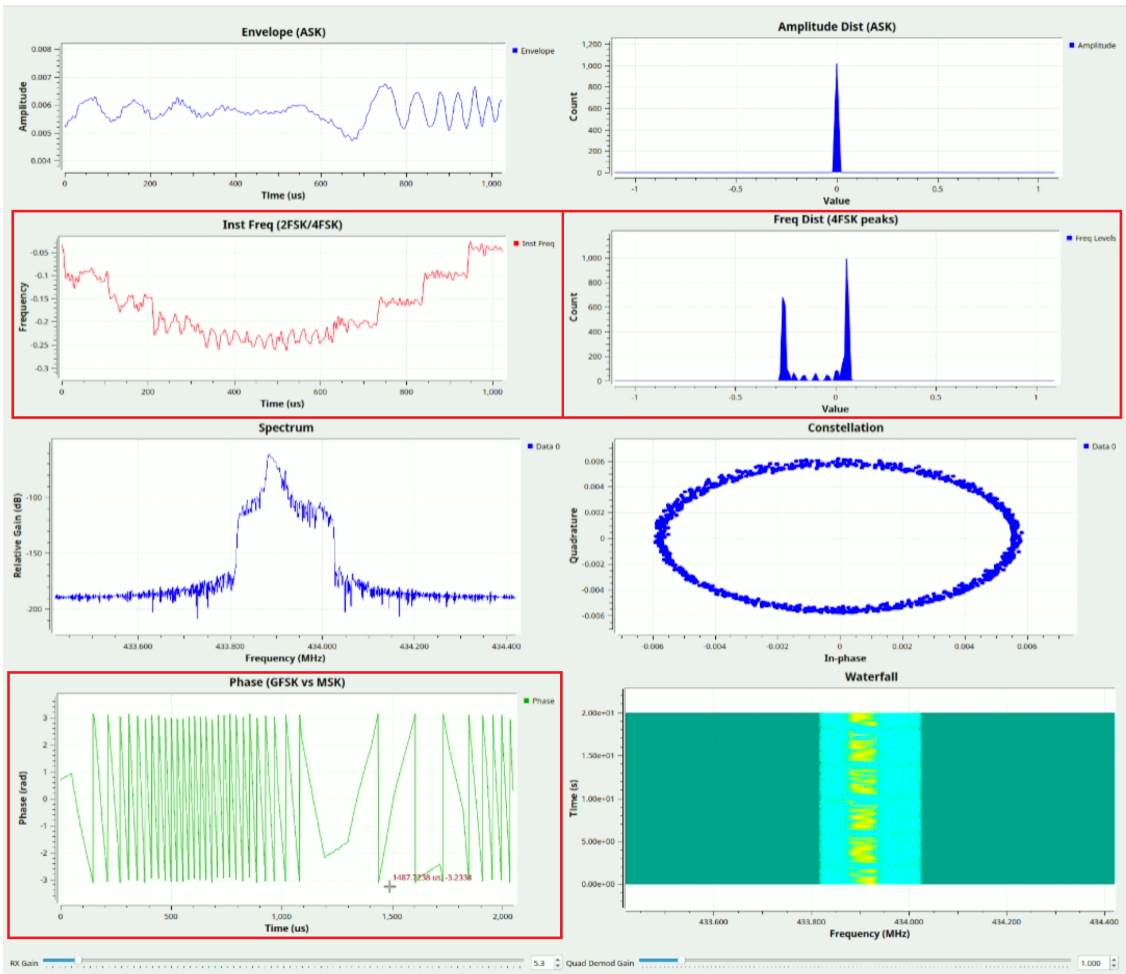

Figure 5: SDR visualization of GFSK modulation showing (a) frequency histogram with two peaks similar to 2-FSK, (b) instantaneous frequency display with smooth, rounded transitions (unlike the sharp edges of standard FSK), (c) phase trajectory with characteristic S-curved Gaussian-smoothed transitions, and (d) waterfall display demonstrating reduced spectral side lobes compared to standard 2-FSK.

3.4.3 Learning Outcomes

Students will learn pulse shaping techniques and their role in controlling signal bandwidth. The concept of the Gaussian filter and its Bandwidth-symbol Time (BT) product becomes clear through observing actual signal spectra, showing the trade-offs between spectral containment and inter-symbol interference.

Comparing FSK and GFSK spectra gives students practical spectral analysis experience and teaches them to quantify efficiency improvements. This skill is important for RF engineers designing systems within regulatory bandwidth constraints.

The lab connects classroom concepts to practical applications through GFSK's use in Bluetooth. Students completing Configuration 2 or 3 learn to identify GFSK from captured signals by its smooth spectral rolloff and reduced side lobes compared to standard FSK.

3.5 Lab 4: MSK Modulation

3.5.1 Session Configuration

Lab 4 uses a session configuration with the modulation type set to MSK. A specific preamble byte and address value are assigned. The key difference from other labs is that the deviation is set to 0. The CC1101 hardware automatically calculates deviation based on the MSK constraint, ensuring a modulation index of exactly 0.5. The library enforces a minimum data rate of 26 kBaud for MSK, and Manchester encoding cannot be used with this modulation type.

3.5.2 Description

Minimum Shift Keying (MSK) is a continuous-phase frequency shift keying (CPFSK) form with a fixed relationship between frequency deviation and bit rate. Setting the deviation to one-quarter of the bit rate yields a modulation index of 0.5. This ensures the carrier phase advances by exactly $\pm\pi/2$ radians per bit period, remaining continuous at transitions.

MSK's continuous-phase property eliminates abrupt phase discontinuities, improving spectral efficiency. This makes MSK suitable for bandwidth-constrained applications like satellite communications.

Operating at 433.92 MHz with a 1.2 kBaud data rate, the lab uses specific preamble and address values. The deviation parameter is set to 0, signaling the hardware to automatically calculate the required deviation ($\Delta f = R_b/4$). This constraint prevents invalid parameters and illustrates the modulation index's intrinsic tie to the scheme.

In Configuration 1, students configure their units for MSK. They understand how MSK modulation operates given the description above.

Configuration 2 and 3 exercises use phase trajectory analysis. MSK's linear ramp phase change of $\pm\pi/2$ per symbol period is its distinguishing feature. This contrasts with GFSK's S-curved transitions and standard FSK's discontinuities. While the frequency histogram shows two peaks, the phase trajectory provides identification. Students also observe MSK's spectral efficiency in the waterfall display.

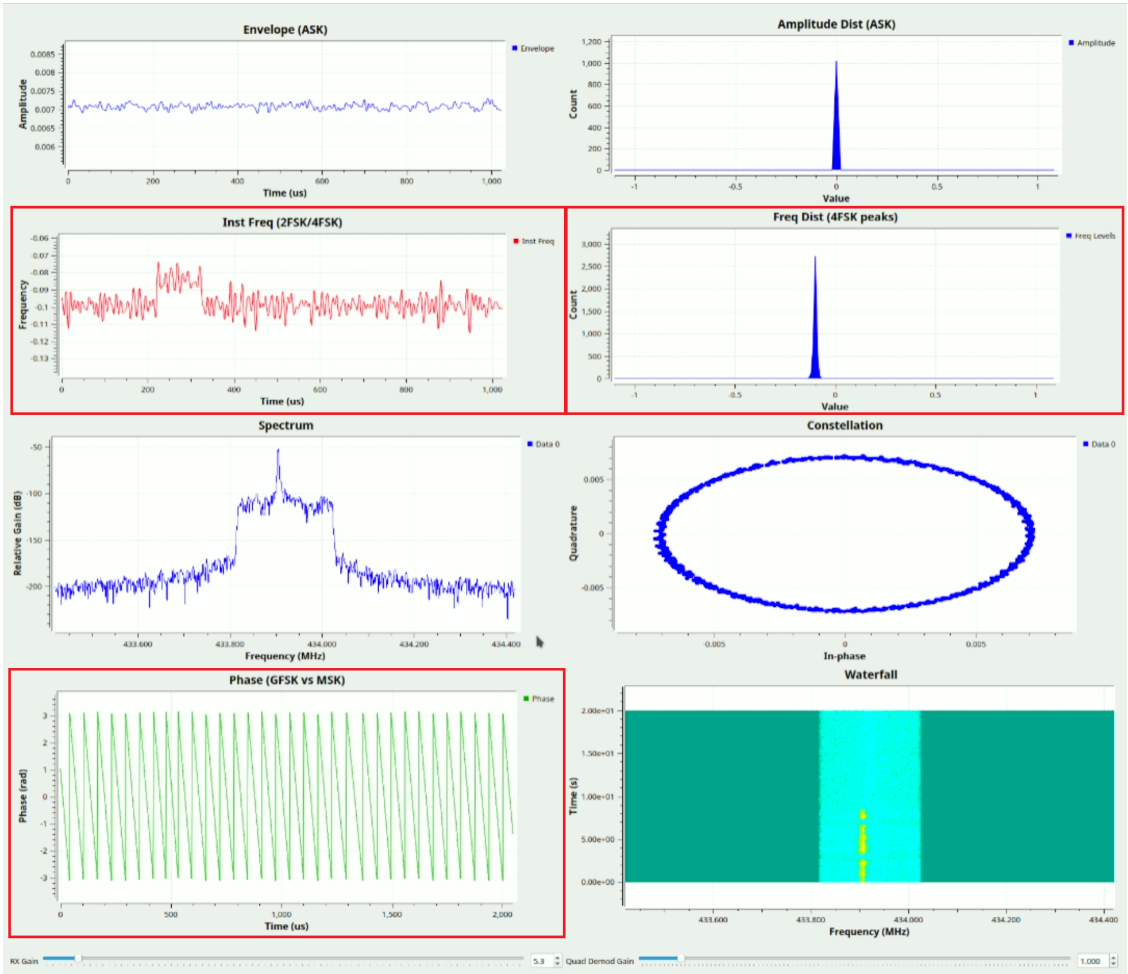

Figure 6: SDR visualization of MSK modulation showing (a) frequency histogram with two peaks at the MSK-constrained deviation, (b) instantaneous frequency display with continuous transitions, (c) phase trajectory with distinctive linear ramp changes, the definitive MSK identifier, and (d) waterfall display demonstrating spectral containment.

3.5.3 Learning Outcomes

After this lab, students will understand continuous-phase modulation and its spectral efficiency advantages. The mathematical relationship between deviation and bit rate in MSK becomes clear during configuration. The lab shows the trade-offs in modulation selection. MSK has excellent spectral efficiency but imposes system design constraints and requires more complex receiver implementations for coherent detection.

Having students compare this exercise with the previous labs reinforces understanding of the modulation taxonomy: 2-FSK and 4-FSK are discontinuous-phase schemes with abrupt frequency transitions at symbol boundaries, GFSK uses pulse shaping to smooth transitions, and MSK achieves continuous phase through precise parameter selection.

3.6 Lab 5: ASK/OOK Modulation

3.6.1 Session Configuration

Lab 5 uses a session configuration with the modulation type set to ASK/OOK. A unique preamble byte and address value are assigned. Unlike FSK-based labs, the deviation is set to 0 and ignored since ASK modulates amplitude rather than frequency. The maximum data rate is 250 kBaud.

3.6.2 Description

Amplitude Shift Keying (ASK) and On-Off Keying (OOK) modulate carrier amplitude rather than frequency. In OOK, the carrier is transmitted at full power (logic 1) or suppressed (logic 0).

OOK is the simplest digital modulation scheme, requiring only carrier switching. This simplicity reduces implementation costs, making OOK common in garage door openers, key fobs, and doorbells. Its prevalence allows students to non-invasively analyze real-world signals.

Operating at 433.92 MHz with a 1.2 kBaud data rate, the lab uses specific preamble and address values. The deviation is set to 0 as it is inapplicable. The carrier frequency remains constant while amplitude varies.

Despite its prevalence, OOK has significant disadvantages. Amplitude-based modulation is more susceptible to noise than frequency-based schemes since receivers detect amplitude levels. OOK also suffers from poor spectral efficiency and jamming vulnerability. These trade-offs limit it to cost-sensitive applications.

Configuration 1 focuses on register configuration for amplitude modulation. Students configure the format and then observe the library's power amplifier adjustments.

Configurations 2 and 3 use envelope detection for classification. ASK/OOK is identifiable from the `Envelope Time` display, showing obvious amplitude variations. This pattern is absent in FSK variants. The amplitude histogram confirms this modulation identification with two peaks. This check allows students to rule out ASK before FSK analysis. Students can also capture real OOK signals from their own consumer devices, extending the lab to practical exploration.

3.6.3 Learning Outcomes

Upon completion of this lab, students will understand the distinction between amplitude-based and frequency-based modulation. The lab demonstrates the practical trade-offs of each approach: ASK offers implementation simplicity but sacrifices noise immunity, while FSK provides robustness at the cost of increased complexity.

Through configuration and signal analysis, students learn why different applications use different modulation schemes. OOK's prevalence in consumer devices despite inferior noise performance shows that engineering decisions involve cost, complexity, and reliability trade-offs beyond technical optimality.

For Configurations 2 and 3, the lab provides skills in distinguishing amplitude and frequency modulation from captured signals. The ability to analyze real-world OOK signals from consumer

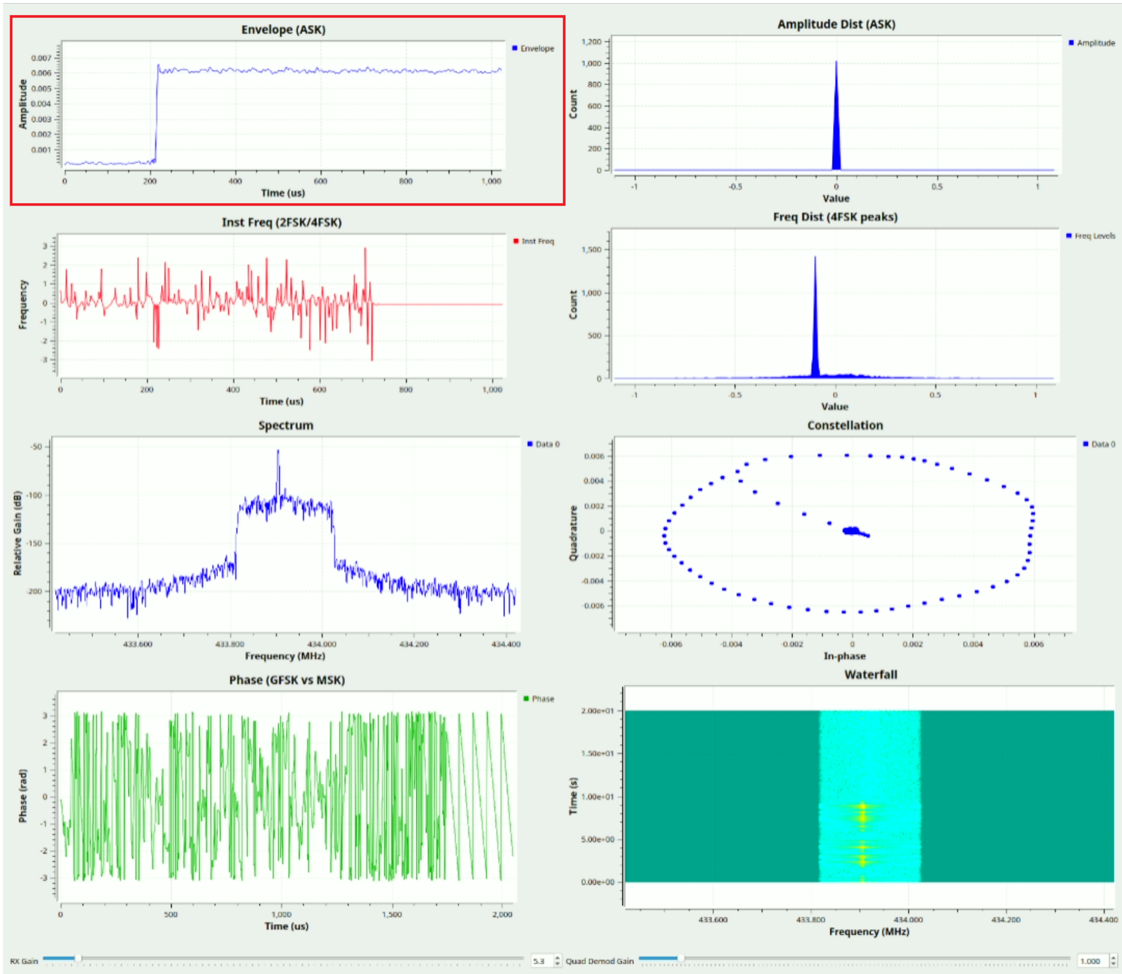

Figure 7: SDR visualization of ASK/OOK modulation showing (a) envelope time display with clear amplitude variations between on and off states (the primary OOK identifier), (b) amplitude histogram with two distinct peaks corresponding to the two power levels, (c) flat instantaneous frequency (no frequency modulation), and (d) waterfall display showing the characteristic on-off carrier pattern.

devices extends the laboratory experience beyond controlled exercises to practical RF exploration.

3.7 Optional Complexity Extensions

Each core lab can be extended by included firmware configuration options. These extensions introduce advanced concepts using the same hardware. Educators can enable extensions to adjust difficulty and complexity. The extensions are cumulative, and enabling multiple options creates challenging exercises requiring mastery of combined concepts.

The receiver firmware includes configuration flags. Educators must modify settings to enable validation, and students must implement the corresponding encoding or calculation to illuminate the LEDs.

3.7.1 Checksum Validation

Error detection is fundamental to working with realistic RF signal modulation, and checksum validation provides core implementation experience. When enabled, the receiver unit rejects packets with mismatched checksums, forcing students to perform proper calculations.

The platform supports three algorithms: SUM, XOR, and CRC8. Byte summation (SUM) computes the arithmetic sum of data bytes modulo 256. This detects most single-byte errors but misses errors preserving the sum. Students implement this by accumulating values:

```
checksum = (data[0] + data[1] + ...  + data[n-1]) & 0xFF
```

Exclusive-or (XOR) computes the bitwise XOR of data bytes. It detects odd numbers of bit errors in a position but misses even numbers. This implementation requires bitwise operations:

```
checksum = data[0] ^ data[1] ^ ...  ^ data[n-1]
```

CRC8 implements an 8-bit Cyclic Redundancy Check. It detects all single-bit errors, double-bit errors, odd numbers of errors, and burst errors shorter than the CRC length. The platform uses the CRC-8-CCITT polynomial (0x07). Students implement this using bit-by-bit division or lookup tables.

For Configurations 2 and 3, students demodulate the signal to extract data bytes, then reverse-engineer which checksum algorithm is in use by analyzing the relationship between data and checksum fields across multiple captured packets.

3.7.2 Manchester Encoding

Manchester encoding (phase encoding) is a line coding technique. Unlike NRZ (Non-Return-to-Zero) encoding where bits are voltage levels, Manchester encoding represents each bit as a transition. Logic 1 is a low-to-high transition; logic 0 is high-to-low.

The main advantage of Manchester encoding is self-clocking; every bit contains a transition, allowing the receiver to extract timing information directly. This improves robustness to clock drift. In this lab, students observe that Manchester-encoded transmissions maintain synchronization with minimal preamble.

Another benefit is that Manchester encoding also eliminates DC bias. Each bit period contains equal time at high and low levels, ensuring zero average DC component. The cost of this improvement is bandwidth. Manchester encoding requires twice the bandwidth of NRZ for the same data rate. Transmitting at 1.2 kbps results in an effective symbol rate of 2.4 kBaud. Students observe this bandwidth expansion in spectral plots.

When this configuration option is enabled, students implement the encoding algorithm. For each data bit, the encoder outputs two bits: logic 1 becomes ”01”, logic 0 becomes ”10”. Errors result in garbled reception.

In Configuration 1, students configure their units to enable Manchester encoding.

For Configurations 2 and 3, Manchester encoding complicates analysis. Students must recognize the doubled symbol rate and transition patterns to decode the data. Universal Radio Hacker

(URH) supports Manchester decoding, but identifying it from raw signals is beneficial to student learning outcomes.

3.7.3 Data Whitening

Data whitening addresses spectral dependency on transmitted data. Repetitive patterns such as long sequences of zeros or ones produce spectral lines that concentrate energy at specific frequencies instead of spreading it across the channel bandwidth. This can cause interference and degrade synchronization.

The whitening process XORs the data with a pseudo-random binary sequence. The sequence makes whitened data appear random, spreading the spectrum. The receiver then performs an XOR operation on received data with the same sequence to recover the original data.

The CC1101 uses a 9-bit linear feedback shift register (LFSR) [22]. In this configuration, each data bit is XORed with the LFSR output. Students implement this LFSR in their transmitter code when whitening is enabled.

Whitening lets students explore pseudo-random sequences. In these sequences, the data appears random, but is deterministic. Any receiver with the correct LFSR polynomial and seed can recover the original data with 100% accuracy.

In Configuration 1, students enable the default whitening used by CC1101 that will match the other CC1101 used by the receiver.

For Configurations 2 and 3, whitening complicates analysis. The demodulated data appears random, and students must apply proper the de-whitening technique. Advanced exercises might require determining the polynomial from captured packets.

3.7.4 Forward Error Correction

Forward Error Correction (FEC) shifts focus to error correction. Unlike checksums, FEC adds redundancy for error identification and correction without retransmission. This is necessary when retransmission is impractical.

The CC1101 implements a rate 1/2 convolutional code with constraint length $m = 4$ and interleaving [22]. The rate 1/2 doubles transmitted data volume. The constraint length determines encoder memory, and interleaving spreads burst errors.

When enabled, data expands during encoding. An 8-byte packet requires 16 bytes. The receiver uses appropriate decoding to recover the original data from the encoded transmission, correcting errors that occurred during propagation.

Configuration 1 students enable the default FEC used by CC1101 that will match the other CC1101 used by the receiver.

For Configurations 2 and 3, FEC creates analysis challenges. The doubled data rate provides a clue for students. Identifying convolutional code parameters is an advanced exercise.

FEC provides experience with information theory. In this lab, students see that error correction costs bandwidth. They also get exposure to the concept that encoding is simple while decoding is complex.

3.7.5 Combining Extensions

All labs have additional configurations to extend the complexity of the exercise. These configurations can be combined to create exercises that will challenge students. For example, enabling Manchester encoding and data whitening requires first enabling data whitening then Manchester encoding. The receiver reverses this: demodulate, Manchester-decode, de-whiten, verify checksum, and validate.

A lab configuration with all extensions approximates commercial protocols. With this level of challenge, students gain understanding of the full protocol stack and wireless communication complexity.

4 Student Learning Outcomes

Student outcomes for this series of labs focuses on hands-on experiential learning of RF technology and modulation techniques through a series of labs with increasing exposure to more advanced techniques. The sequence of labs, as presented in this work, are designed to address the following 2025-2026 ABET Student Outcomes for Engineering Programs [23]:

1) An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.

6) An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.

7) An ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

Other learning outcomes can be addressed through classroom instruction, written or oral assignments, team assignments, and relation to real-world concepts or exercises. These labs are designed with consideration towards adaption and implementation in existing courses or curricula.

Upon completing this series of labs, students will be able to approach complex engineering problems systematically, applying the engineering design process to meet specifications within design constraints. Some constraints are determined by the lab, but others can be adjusted according to the desired instructor difficulty. By working with these hands-on modules, students will be exposed to simple RF hardware to send and receive wireless signals. This allows students to gain transferable experience to industry standard equipment and advanced wireless applications. Student understanding of wireless communication protocols and the importance of modulation on modern communications is particularly transferable across course curricula.

Although the assignment assessment methodology is left up to the instructor, this work provides a framework for lab implementation and outcomes. Student learning outcomes for each lab are described and paired with example results for comparison. Results are explained for instructor reference and tailoring, but are not full solutions for the lab.

5 Solutions

Solutions for all five lab modules are available to educators upon request. Educators can contact the authors for access. Access is provided to verified educators to maintain the integrity of the lab exercises for student assessment.

6 Conclusion

This work introduces an interactive platform focused on hands-on lab exercises designed to increase RF education accessibility through low-cost hardware and open-source software. This material bridges the gap between theory and practice with hands-on modules using components under $75, while an advanced configuration with the HackRF One SDR remains accessible at $400 per station. Educators can use preconfigured labs without modification or adapt the material to meet learning outcomes for existing course curricula.

Five pre-built labs cover 2-FSK, 4-FSK, GFSK, MSK, and ASK modulation schemes. Optional features like checksums, Manchester encoding, data whitening, and FEC allow for adjustable complexity. Within each lab, three configurations accommodate different resource availability, ranging from basic transmission exercises to advanced SDR analysis. Visual feedback via the LED array on the instructor's hardware validates student transmission success. Open-source firmware enables customization, while the solution package available to educators on request supports assessment. The presented design methodology allows educators to integrate these exercises into existing curricula without requiring extensive equipment investments or firmware development expertise.

References

[1] D. M. Pozar, *Microwave Engineering*, 4th ed. Wiley & Sons, 2012.

[2] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. New York: McGraw-Hill, 2008.

[3] K. T. Selvan and K. F. Warnick, *Teaching electromagnetics: Innovative Approaches and Pedagogical Strategies*, 1st ed. CRC Press, 2021.

[4] A. Goldsmith, *Wireless Communications*. Cambridge, UK: Cambridge University Press, 2005.

[5] S. Haykin and M. Moher, *Communication Systems*, 5th ed. Hoboken, NJ: Wiley, 2014.

[6] E. Tanyıldızı and A. Orhan, "A virtual electric machine laboratory for synchronous machine application," *Computer Applications in Engineering Education*, vol. 17, pp. 187–195, 2008.

[7] H. Kim, N. Rogers, G. York, and P. Leiffer, "Enhanced learning by visualization applying

embedded hands-on in electromagnetics class," *2024 ASEE Annual Conference and Exposition Proceedings*, 2024.

[8] E. Genç, "Integration of software defined radio and add-on board into digital communication education with hands-on applications," *International Journal of Applied Mathematics, Electronics and Computers*, pp. 118–118, 2016.

[9] H. G. Espinosa, T. Fickenscher, N. Littman, and D. V. Thiel, "Teaching wireless communications courses: An experiential learning approach," in *2020 14th European Conference on Antennas and Propagation (EuCAP)*, 2020.

[10] R. Stewart, L. H. Crockett, D. S. W. Atkinson, K. W. Barlee, D. Crawford, I. C. Chalmers, M. McLernon, and E. Sozer, "A low-cost desktop software defined radio design environment using matlab, simulink, and the rtl-sdr," *IEEE Communications Magazine*, vol. 53, pp. 64–71, 2015.

[11] T. F. Collins, R. Getz, D. Pu, and A. M. Wyglinski, *Software-Defined Radio for Engineers*, ser. Artech House Mobile Communications Series. Norwood, MA: Artech House, 2018, illustrated Edition.

[12] D. M. Molla, H. Badis, L. George, and M. Berbineau, "Software defined radio platforms for wireless technologies," *IEEE Access*, vol. 10, pp. 26 203–26 229, 2022.

[13] M. El-Hajjar, Q. Nguyen, R. G. Maunder, and S. X. Ng, "Demonstrating the practical challenges of wireless communications using usrp," *IEEE Communications Magazine*, vol. 52, pp. 194–201, 2014.

[14] K. VonEhr, W. Neuson, and B. Dunne, "Software defined radio: Choosing the right system for your communications course," *2016 ASEE Annual Conference and Exposition Proceedings*, 2016.

[15] M. Inoñan and R. Hussein, "Melody: A platform-agnostic model for building and evaluating remote labs of software-defined radio technology," *IEEE Access*, vol. 11, pp. 127 550–127 566, 2023.

[16] R. G. Ekanthappa, R. Escobar, A. Matevossian, and D. Akopian, "A remote laboratory for usrp-based software defined radio," *SPIE Proceedings*, vol. 9030, p. 903003, 2014.

[17] O. A. Dobre, A. A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET Communications*, vol. 1, pp. 137–156, 2007.

[18] E. Karincic, L. Linkous, and E. Topsakal, "Interactive educational platform for digital modulation recognition and signal analysis," in *2026 United States National Committee of URSI National Radio Science Meeting (USNC-URSI NRSM)*, 2026.

[19] E. Karincic (dollarhyde), "Interactive rf modulation platform." [Online]. Available: https://github.com/Dollarhyde/rf-modulation-platform

[20] M. Furga, "Cc1101: Arduino library for cc1101 transceiver." [Online]. Available: https://github.com/mfurga/cc1101

[21] J. Pohl, “Urh: Universal radio hacker: Investigate wireless protocols like a boss.” [Online]. Available: https://github.com/jopohl/urh

[22] Texas Instruments, *CC1101 Low-Power Sub-1 GHz RF Transceiver*, Texas Instruments, Dallas, TX, 2018, rev. SWRS061I. [Online]. Available: https://www.ti.com/lit/ds/symlink/cc1101.pdf

[23] ABET Engineering Accreditation Commission, *Criteria for Accrediting Engineering Programs, 2025-2026*, ABET, Baltimore, MD, USA, 2025. [Online]. Available: https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2025-2026/