eSim Semester Long Internship - Autumn 2026 Tasks

Google Form : **Link**
INSTRUCTIONS

- This internship task has been divided into 7 Screening Tasks, out of which the participants need to submit at least one task.
- For any queries, email to contact-esim@fossee.in with the subject line as **"eSim Semester Long Internship - Autumn 2026 - Query"**.

RESOURCES

- To download and install eSim, click **here**.
- Click **here** for resources under eSim.

Task 1. eSim Research Migration(Electronics and related fields):

Instructions:

Propose circuit simulations you wish to implement in eSim v2.3 or eSim v2.4 or eSim v2.5 under Research Migration Project.

1. Please make sure that the proposed circuits are not available in the eSim Website, and the examples provided with eSim.
2. Your proposal will be reviewed and approved/disapproved (within one working day). Only after receiving the approval of your proposal through email, can you proceed to attempt the task.
3. The proposed circuits may be both Analog and Digital.
4. **To propose the circuits click here and follow the procedure.**

Technical Requirements:

Knowledge of analog and digital electronics. Familiarity with eSim and its features.

Procedure to Submit:

- Check the procedure here to submit.
- Inform us by sending a mail to contact-esim@fossee.in with the subject line **"eSim Semester Long Internship - Autumn 2026 Submission Task 1"** with the proposal details.

Evaluation Criteria:

The following criteria will be considered while evaluating the circuit simulation:

- complexity of the circuit
- accuracy of simulation results obtained

- number of the circuits successfully completed

Task 2. Communication Protocol Modelling and Verification Using eSim :

Design, model and verify a communication protocol at the circuit level using only eSim.

Instructions:

1. Choose a communication protocol
2. Model the transmitter and receiver
3. Connect and simulate the system
4. Verify the data transfer
5. Can include analog,digital or mixed signal together as per the user analysis
6. To propose the circuits, click here and follow the procedure.

Requirements:

- Basic knowledge of digital electronics
- Understanding of communication protocols (UART, SPI, I²C, etc.)
- Familiarity with eSim software

Procedure to Submit:

- Check the procedure here to submit.
- Inform us by sending a mail to contact-esim@fossee.in with the subject line **"eSim Semester Long Internship - Autumn 2026 Submission Task 2"** with the proposal details.

Evaluation criteria:

The following criteria will be considered while evaluating the circuit simulation:

- Successfully modelled communication protocol in eSim
- Clear demonstration of correct data flow from transmitter to receiver
- Proper verification of timing, synchronization, and errors (if applicable)

Task 3. Complete IoT System Design Using Only eSim (Circuit-Level)

Design and simulate a **complete IoT system at circuit level** using only **eSim**, without relying on external IoT platforms or cloud services

Instructions:

1. Selection and modeling of sensors (e.g., temperature,light, humidity) and

required blocks using available eSim components or equivalent circuit models
2. Connect all blocks into a **single complete system schematic** and simulate the overall system
3. Verify and document that the **system output behaves as expected** under different conditions
4. The system can include analog,digital or mixed signal together as per the user analysis
5. **To propose the circuits, click here and follow the procedure.**

Technical Requirements:

- Knowledge of analog digital electronics and IoT .
- Familiarity with eSim and its features.

Procedure to Submit:

- Check the procedure here to submit.
- Inform us by sending a mail to contact-esim@fossee.in with the subject line **"eSim Semester Long Internship - Autumn 2026 Submission Task 3)"** with the proposal details.

Evaluation Criteria:

The following criteria will be considered while evaluating the circuit simulation:

- Successfully **simulated circuit-level IoT system** in eSim
- Clear demonstration of input to output signal flow

Task 4. eSim Upgradation(CSE and related fields):

As the screening task to get selected, you need to figure out the various problems faced while installing eSim-2.5 on Ubuntu 25 and above systems. Please ensure that you use the technologies mentioned. Please follow the submission procedure and go through the evaluation criteria carefully while submitting the task.

Description:

The task involves to figure out the dependency issues while installing eSim 2.5 on Ubuntu 25.04 and fix at least one of the issues.

Steps:

1. You need to have Ubuntu 25.04 installed on your system(You may use a Virtual Box).
2. Download and try installing eSim-2.5 on Ubuntu 25.04 machine from here.
3. List down the problems you would be facing while installing eSim, for example

you may get stuck while eSim installs llvm.
4. Look into the install-eSim.sh script and try making some changes to fix an error.
5. If the problem is not fixed, you may comment out some part of the script so that you may move on to the next bug
6. You may need to refer to the various resources
7. Create a well documented report of the problems you have faced while installing eSim and mention the steps in detail and the measures you have taken to fix them

Note: Fixing at least one issue is desirable

Technologies to be used:

1. Bash Script
2. Python
3. eSim
4. Spoken Tutorial on Git
5. Ubuntu 25.04 OS

Submission Procedure:

1. Create a GiHub Account
2. Fork the eSim Repository
3. Checkout to the installer branch
4. Commit the code files here
5. Share back the link for your forked repository and the documented report by the below-mentioned deadline by mail to contact-esim@fossee.in. Inform us by sending a mail to contact-esim@fossee.in with the subject line **"eSim Semester Long Internship - Autumn 2026 Submission Task 4"** with the submission task details.

Evaluation criteria:

1. Quality of documentation

(For example – On visiting your repository, any user should understand the details of the bugs/issues, methodology, steps involved in fixing the bugs, etc.)

2. Number of issues reported
3. Number of issues fixed
4. The difficulty level of the problems fixed(For example: A dependency issue which interrupts the installation of the main GUI of eSim will be given higher weightage than those which interrupts the installation of smaller blocks like NgVeri, etc.)

Task 5. Tool manager(CSE and related fields):

Problem Statement:

eSim is an open-source EDA tool for circuit design, simulation, analysis, and PCB design. It integrates several external tools and libraries to provide a seamless environment for engineers and researchers. Managing these tools manually can be tedious, especially when considering updates, compatibility, and user-specific configurations.

The goal of this task is to design an **"Automated Tool Manager"** for eSim(**preferably in python**) that automates the installation, configuration, update, and management of external tools and dependencies.

Below are the requirements for the screening tasks. The task submission needs to meet any 2 of the requirements listed below.

Requirements:

1. **Tool Installation Management:**
    - The manager should be able to download and install required external tools (e.g., Ngspice, KiCad, etc.) automatically.
    - Ensure compatibility with the system environment (Linux/Windows).
    - Handle version control to ensure the correct versions of tools are installed.
2. **Update and Upgrade System:**
    - Design a system that checks for updates for the external tools and libraries.
    - Implement functionality for the user to update these tools with minimal manual intervention.
3. **Configuration Handling:**
    - Automate the configuration of the installed tools based on user settings or predefined settings.
    - Handle path management and environment variable settings so that the tools work seamlessly with eSim.
4. **Dependency Checker:**
    - Ensure that all necessary dependencies for the installed tools are checked and managed.
    - Provide feedback to the user if certain dependencies are missing or incompatible.
5. **User Interface:**
    - Provide a simple, user-friendly command-line or graphical interface for managing the tools.
    - Allow the user to view installed tools, versions, and any updates available.
    - Provide a log of actions taken (e.g., installations, updates, errors).
6. **Additional Features (Optional):**
    - Cross-platform support (Linux/Windows/Mac).
    - Integration with popular package managers (e.g., apt, Homebrew, Chocolatey) to streamline tool management.

Deliverables:

1. **Design Document:** A detailed document describing the architecture of the Automated Tool Manager. This should include the overall design, module breakdown, and how the different components (installation, updates, configuration, etc.) interact.
2. **Code Implementation:**

    - A basic prototype or proof-of-concept code that demonstrates at least tool installation and version checking.
    - You may use any language or platform for the implementation (Python, Shell script, etc.).
3. **Instructions for Execution:** Provide clear steps on how to run and test the prototype. This should include installation instructions for any dependencies required for your tool.
4. **Presentation (Optional): A short video or presentation (5-10 minutes) explaining your approach, design choices, and a demo of your tool manager.**

Evaluation Criteria:

- **Functionality:** How well the tool manager achieves the desired functionality (installation, updates, configuration).
- **Design:** Clarity and efficiency of the design, ease of use, and modularity.
- **Documentation:** The clarity of the design document and code documentation.
- **Creativity:** Additional features or unique approaches taken to solve the problem.
- **Code Quality:** Readability, structure, and use of best practices in coding.

Submission Guidelines:

- Submit your design document, code, and instructions as a GitHub repository link.
- Ensure that the created repository is private and give access to the following github id: **https://github.com/Eyantra698Sumanto**
- Share back the link for your repository and the documented report by the below-mentioned deadline by mail to contact-esim@fossee.in. Inform us by sending a mail to contact-esim@fossee.in with the subject line **"eSim Semester Long Internship - Autumn 2026 Submission Task 5"** with the submission task details.

Task 6. KiCad Plugin Development (CSE and related fields):

Instructions:
Propose and implement any plugin for KiCad 9.
The objective of this task is to develop a working base version of a KiCad plugin, which can be further improved and extended during the internship period.
The proposed plugin should involve both Computer Science and Electronics knowledge and must be compatible with the latest version of KiCad.
Only after receiving the approval of your proposal through email, can you proceed to attempt the task.

Technical Requirements:

- Programming skills and ability to write clean, modular code
- Understanding of software architecture and extensible systems
- Familiarity with API usage and integration
- Experience with scripting or application extension frameworks
- Ability to design, test, and debug software components
- Familiarity with version control and documentation practices

Procedure to Submit:

- Prepare a proposal describing the plugin idea and its basic functionality
- Inform us by sending a mail to contact-esim@fossee.in with the subject line **"eSim Semester Long Internship - Autumn 2026 Submission Task 6"** along with the plugin proposal details
- Submit your design document, code, and instructions as a GitHub repository link
- Share back the link for your forked repository and the documented report by the below-mentioned deadline by mail to contact-esim@fossee.in. Inform us by sending a mail to contact-esim@fossee.in with the subject line **"eSim Semester Long Internship - Autumn 2026 Submission Task 6"** with the submission task details.

Deliverables :

1. **Design Document:** A detailed document describing the architecture of the KiCad plugin. This should include the overall design, module breakdown, and how different components of the plugin (user interface, data processing, interaction with KiCad API, automation logic, etc.) interact with each other.

2. **Code Implementation:**
   A basic working prototype or proof-of-concept KiCad plugin demonstrating core functionality (for example: integration with KiCad, basic automation, rule checking, data extraction, or workflow enhancement depending on the plugin idea).
   You may use any suitable language supported by KiCad plugins (such as Python).

3. **Instructions for Execution:** Provide clear steps on how to install, run, and test the KiCad plugin.

4. **Presentation (Optional):**
   A short video or presentation (5–10 minutes) explaining your plugin concept, design decisions, implementation approach, and a demo of the plugin working inside KiCad.

Evaluation Criteria:
The following criteria will be considered while evaluating the plugin development:

- complexity of the plugin
- correctness of implementation
- integration of Computer Science and Electronics concepts
- compatibility with the latest KiCad version
- number of features successfully implemented
- Documentation: The clarity of the design document and code documentation.

Task 7. PCB Design using eSim

Instructions:

Design any Printed Circuit Board (PCB) using eSim Version 2.3 or Version 2.5.

The objective of this task is to evaluate the candidate's understanding of the complete PCB design workflow using the eSim (KiCad-based) design environment. The PCB should demonstrate proper schematic capture, footprint assignment, PCB layout, routing, and design practices.

The submitted design must be fully compatible with eSim Version 2.3/2.5 and should open without any missing libraries or project errors.

Technical Requirements:

- Ability to create clean and organized schematics
- Proper selection and mapping of PCB footprints
- Knowledge of PCB layout techniques and routing practices
- Understanding of multilayer PCB design

Design Requirements:

- Design any functional PCB of your choice.
- The PCB must contain at least two (2) copper layers. (designs with a higher number of layers may receive additional consideration during the evaluation process).
- The complete project must be created using eSim Version 2.3 or Version 2.5.
- The design should be well-organized and properly documented.

Mandatory Design Flow

The PCB design must be completed using the following workflow:

1. Create the schematic using eSchema. (eSim EDA)
2. Assign appropriate component footprints.
3. Generate the PCB using the Sch to PCB Update functionality.
4. Route all required PCB tracks following good PCB design practices.
5. Perform Design Rule Check (DRC) and resolve any critical errors.
6. Generate and demonstrate the PCB in the 3D Viewer available in eSim (KiCad).

Procedure to Submit

1. Prepare a proposal describing your PCB design idea and its basic functionality.
2. Fork the official GitHub repository: https://github.com/FOSSEE/eSim_PCB_Design_Project_Files
3. Add your PCB design files, design document, and any supporting files to your forked repository, and commit your changes.
4. Create a Pull Request (PR) from your forked repository to the official FOSSEE/eSim_PCB_Design_Project_Files repository.
5. Inform us by sending a mail to contact-esim@fossee.in with the subject line "eSim Semester Long Internship - Autumn 2026 Submission Task 7" with the submission task details.
6. In your email, include:
    - Your PCB design proposal.
    - The link to your forked GitHub repository.
    - The link to your Pull Request (PR).

Note: YEnsure that the created repository is private and give access to the following github **id: https://github.com/Eyantra698Sumanto & https://github.com/OfficialVarad**

Deliverables

Submit a detailed PDF report containing:

- Project overview
- Circuit description
- Schematic screenshots
- Footprint assignment details
- PCB layout screenshots
- Routing methodology
- DRC results
- 3D View screenshots
- Design considerations and conclusions

eSim Project Folder

Submit the complete eSim project workspace containing all project files, including:

- Schematic files (eg. title.kicad_sch
- PCB layout design (eg. title.kicad_pcb, title.kicad_prl, title.kicad_pro
- Libraries (if any)
- Project configuration files

The project should open directly in eSim Version 2.3/2.5 without requiring additional modifications.

Presentation (Optional)

A 5 minute presentation or demonstration video explaining:

- PCB design objective
- Schematic creation
- PCB layout process
- Routing strategy
- DRC verification
- 3D visualization
- Overall design decisions

Evaluation Criteria

The PCB design will be evaluated based on:

- Complexity of the design
- Proper footprint assignment
- PCB layout quality
- Routing quality and adherence to PCB design practices
- Successful DRC validation
- Compatibility with eSim Version 2.3/2.5
- Documentation quality
- Overall project organization and presentation