

Intern Requirements

- Basic knowledge of JavaScript, TypeScript, HTML, and CSS
- Foundational Node.js skills
- Experience with Chrome Developer Tools for debugging
- Familiarity with npm for dependency management
- General knowledge of SvelteKit or React
- Ability to create server-side endpoints using Express.js

Nice-to-Have Skills:

- Experience with SVG manipulation
- Familiarity with Electron for desktop applications

Intern Challenge: SvelteKit Blockly Application for Python Code Generation

Your challenge is demonstrating your creativity and technical skills by building a SvelteKit application that integrates Blockly to generate Python code. Follow these detailed steps to complete the challenge:

Objectives:

- 1. Set Up Blockly in a SvelteKit App:**
 - Create a SvelteKit application and integrate Blockly into it.
 - Ensure the application generates valid Python code from Blockly blocks.
- 2. Create Custom Blockly Blocks:**
 - **LED Control Block:**
 - Design a custom block that generates Python code to turn an LED on and off using `pyFirmata`.
 - **Delay Block:**
 - Create a block that pauses the program execution for a specified number of seconds.
 - Ensure the delay duration is configurable via the block's input.
 - **Button Control Block:**
 - Develop a block that generates Python code to turn the LED on or off based on button input.
- 3. Ensure Code Formatting and Compilation:**
 - Validate that the generated Python code is correctly formatted and free of syntax errors.
 - Confirm the code compiles and works as intended when executed.
- 4. Build a Functional Blockly Workspace:**
 - Arrange the blocks in the Blockly workspace to create a program that:
 - Turns on the LED when a button is pressed.
 - Turns off the LED when the button is released.
- 5. Application Features:**
 - Include a preview or export functionality that shows the generated Python code in a readable format.
 - Provide clear instructions or comments for users to connect the Python script to an Arduino board using `pyFirmata`.

Deliverables:

- A fully functional SvelteKit application with the described Blockly setup.
- Custom Blockly blocks for LED control, delay, and button interaction.
- A sample Blockly program that demonstrates the described functionality (button-controlled LED).

- A short readme or documentation describing the setup process and how to run your application.

This challenge tests your ability to integrate technologies, create custom components, and ensure seamless user interaction. Good luck, and we're excited to see your innovation in action!

Intern Challenge: Electron App for Compiling and Running Python Code on an Arduino

Your challenge is to build an Electron application that compiles Python code, communicates with an Arduino, and controls its functionality. This project will test your ability to create a desktop app with Python-Arduino integration.

Objectives:

1. Set Up an Electron Application:

- Create an Electron app with a user interface that includes:
 - A text area for entering Python code.
 - A dropdown or selector to choose the serial/USB port where the Arduino is connected.
 - A button to upload and execute the Python code.
 - A "Refresh" button to reset the application for new code entry.

2. Arduino Setup:

- Configure your Arduino with **StandardFirmata** firmware.
- Ensure that the Arduino can communicate with your application over a serial connection.

3. Python-Arduino Communication:

- Use **pyFirmata** to establish communication between your Python script and the Arduino.
- Ensure the Electron app can compile and send Python code to execute on the Arduino.

4. LED Blink Functionality:

- Write Python code in the app's text area to blink the internal LED (pin 13) on the Arduino.
- Verify that the code sent to the Arduino performs as expected.

5. Interface Features:

- Provide real-time feedback on the connection status with the Arduino.
- Include error handling for issues like invalid serial port selection or syntax errors in the Python code.
- Allow the user to refresh the text area and recompile code without restarting the app.

Deliverables:

- A fully functional Electron app with:
 - A working text area for Python code entry.
 - Serial/USB port selection and connection management.
 - Buttons for uploading and refreshing code.
- The ability to blink the Arduino's internal LED (pin 13) using Python code entered in the app.

- Clear instructions on how to upload StandardFirmata to the Arduino and run the app.

This challenge combines desktop app development with hardware interaction, allowing you to showcase your skills in creating seamless integrations between software and physical devices. Good luck, and have fun with this project!

Submission Link:

Please submit your task via the link below.

<https://forms.gle/KQzPFfnLKKfgkLSQ7>