

Aniket Giri

PROPER INSTRUCTIONS TO IMPLEMENT THE CODE

1. Save the Program:

- Save the provided C program to a file, e.g., led_control.c.

2. Compile the Program:

- Open a terminal (or command prompt) and navigate to the directory where led_control.c is saved.
- Use a C compiler to compile the program. For instance, if you are using gcc(GNU Compiler Collection), you can compile it with the following command:
- bash
- Copy code
- gcc -o led_control led_control.c
- This command will produce an executable file named led_control.

3. Run the Program:

- In the terminal, execute the compiled program:
- bash
- Copy code
- ./led_control
- You should see the output showing the initial and updated LED group statuses.

For Embedded Development (Microcontroller):

1. Prepare the Environment:

- Install the appropriate toolchain for your microcontroller. For example, if you are using an ARM-based microcontroller, you might use the ARM GCC toolchain.
- Install an Integrated Development Environment (IDE) such as Keil MDK, STM32CubeIDE, MPLAB X, or similar, depending on your microcontroller.

2. Create a New Project:

- Open your IDE and create a new project for your microcontroller.
- Configure the project settings according to your microcontroller specifications (e.g., clock settings, peripherals).

3. Add Source Files:

- Add the led_control.c file to your project. This is usually done through the project explorer or a file manager within the IDE.
- Ensure that you also include any necessary header files or libraries.

4. Configure the Compiler:

- Set up the compiler and linker settings according to the microcontroller's requirements. This typically involves configuring paths, optimization levels, and memory settings.

5. Compile the Program:

- Use the build or compile option in your IDE to compile the code. The IDE will invoke the toolchain to generate a binary or hex file suitable for the microcontroller.

6. Upload to Microcontroller:

- Connect your microcontroller to your development environment via a programmer or debugger (e.g., ST-Link, J-Link).
- Use the upload or flash option in your IDE to transfer the compiled program to the microcontroller.

7. Test the Program:

- Once the program is uploaded, you can use the IDE's debugging tools to step through the code and monitor variables.

- If your microcontroller has peripherals connected (such as LEDs), observe their behavior according to the program logic. Otherwise, use serial output or debugging interfaces to verify the program's functionality