

The AI Paradox: Suspicion vs. Utility

Quick Show of Hands / Who's a little suspicious of AI as "cheating"? Immoral?

Addresses the ethical and professional hesitations many programmers feel when integrating AI into their work, questioning intellectual honesty.

Use of AI

Acknowledges that AI is an active tool in the development process.

Personal Anecdote: Treating AI (like ChatGPT) as a cost-effective, trainable "employee" capable of generating significant amounts of code (e.g., "4000 lines of code"). This provides a pragmatic view of AI's current capabilities.

Coding for Constrained and Critical Systems

Coding For Tiny Devices / Resources Constraints:

Focuses on **embedded systems** running on **small microcontrollers** with **limited CPU, RAM, and Flash**.

Notes the challenging environment: **No operating system, no safety net** (often "bare metal" coding).

Critical Applications:

Code runs in **safety-critical aviation**, emphasizing that reliability and verification are paramount due to high-stakes consequences of failure.

AI as an Efficiency Multiplier

Efficient Solution from AI:

Suggested simpler math: AI can optimize logic by suggesting mathematically simpler or computationally less expensive algorithms, critical for constrained devices.

Removed dead code paths: AI helps identify and suggest the removal of unreachable or redundant code, reducing program size (Flash usage).

Pointed out: AI acts as a pair programmer, highlighting inefficiencies or potential issues.

AI tools reduce time spent starting from scratch by generating code suggestions that programmers can refine and verify.

Describes the primary benefit: accelerating development by automating boilerplate or initial drafts.

AI-Supported Coding Workflows & Skills

Problem Explanation Skills:

Clear communication of programming problems becomes crucial for getting useful AI assistance and debugging more efficiently. The key skill shifts toward effective prompt engineering and articulation of constraints.

Critical Code Review and Testing:

The programmer's indispensable role is as the **critical verifier**—thoroughly reviewing, testing, and validating generated code, especially in safety-critical contexts where AI output cannot be implicitly trusted.

Smart AI USAGE TIPS

Active Engagement with AI:

Interact with AI to learn and gain deeper understanding, rather than just copying answers. Using AI as a Socratic partner or tutor to improve personal coding skills and knowledge.