# Workflow & Collaboration Guide

## Purpose

This document defines how ChatGPT, VS Code / PlatformIO, and GitHub work together to support the McCoy Speaks project and future related builds.

### 🧭 Working Agreement & Workflow Map

#### 🎯 Purpose

To define how each workspace and tool fits together so development, documentation, and version control stay consistent across the McCoy Speaks project and future character builds.

#### 1. Roles & Responsibilities

| Tool / Space | Purpose | What Happens Here |
| --- | --- | --- |
| **ChatGPT — “Project Plan” Chat** | *Master Document (Canvas)* | Contains the official project plan (Hardware, Software, Media, etc.). Updated only after decisions are finalized. Acts as the single-source master record. |
| **ChatGPT — “Software Plan” Chat** | *Development Lab* | Used for code work, brainstorming, testing software logic, and drafts. When complete, updates are merged into the Project Plan document. |
| **ChatGPT File Uploads** | *Reference Materials* | Upload design zips, schematics, or other projects for pattern extraction. Used temporarily; once extracted, content is documented in GitHub and/or Project Plan. |
| **VS Code / PlatformIO** | *Active Development Environment* | All firmware coding and compilation. Local repo mirrors GitHub repo. Changes staged, built, and tested here. Code updates can later be committed and pushed to GitHub. |
| **GitHub Repository** | *Definitive Source of Truth (Code + Docs)* | Holds current versions of firmware, hardware files, and docs. Public-facing record of releases. Version control, collaboration, and issue tracking. |

#### 2. Workflow Sequence

┌───────────────┐  
 │ Idea / Test │  
 │ (Software Chat) │  
 └──────┬────────┘  
 │  
 ▼  
 ┌───────────────┐  
 │ Decision Made│  
 │ (Finalize Logic) │  
 └──────┬────────┘  
 │  
 ▼  
 ┌────────────────────────┐  
 │ Update Project Plan │  
 │ (Canvas Document) │  
 └──────┬─────────────────┘  
 │  
 ▼  
 ┌───────────────┐  
 │ Implement in │  
 │ VS Code / PIO │  
 └──────┬────────┘  
 │  
 ▼  
 ┌───────────────┐  
 │ Test / Build │  
 │ (Hardware) │  
 └──────┬────────┘  
 │  
 ▼  
 ┌───────────────┐  
 │ Commit + Push │  
 │ to GitHub │  
 └───────────────┘

#### 3. Key Principles

* **ChatGPT = design & documentation space**  
  Use this for structured thinking, explanation, and recording context.
* **VS Code / PlatformIO = implementation space**  
  Code, test, debug, and compile here.
* **GitHub = preservation & version control**  
  Treat GitHub as the “vault” — everything ends up here, tagged and backed up.
* **Exporting from ChatGPT**  
  Export Markdown files from ChatGPT (like this plan) and commit them to /docs/ in your repo with version tags (v1.1, v1.2, etc.).

#### 4. Future Scalability

This workflow is reusable. For each future Star Trek character build:  
- Duplicate this workflow under a new repo (e.g., /spock-speaks/, /kirk-speaks/).  
- Keep ChatGPT as the design & documentation workspace.  
- Keep VS Code as the coding environment.  
- Use GitHub as the archive of record.