# McCoy Speaks — Project Plan (v1.0 Baseline)

## 1. Overview

**Goal:** Create a self-contained playback system that delivers authentic Dr. McCoy voice lines when triggered by a single pushbutton. The system includes optional LED feedback and internal audio playback with reliable power management.

**Design Philosophy:**

* Favor simple, reliable, and repairable design over compactness.
* Maintain clear documentation throughout all design phases.
* Keep GitHub and local project directories as the definitive record.

**Current State:**

* Audio clips collected.
* Hardware selection and procurement underway.
* Software structure defined.

## 2. Hardware

### 2.1 Core Components

| Function | Component | Notes |
| --- | --- | --- |
| MCU | ATtiny85 | Core logic, programmed via PlatformIO. |
| Power | EEMB LiPo 3.7V 280mAh (562030) | With Molex connector; polarity verified. |
| Charging | TP4056 Module | LiPo charging module with protection; micro-USB input. |
| Protection | Polyfuse | In-line on battery positive lead. |
| Power Control | On/Off Switch | Disconnects battery positive to isolate circuit. |
| Control | Pushbutton | Single button (mode/trigger), connected to PB2 (pin 7). |
| Audio | Speaker + Amplifier Module | Amplified output from ATtiny85 PWM pin. |
| Indicator | LED (Optional) | May indicate playback or system activity. |

### 2.2 Electrical Notes

* Use internal pull-up for button input.
* LED tied to PB1 or PB0 depending on pin availability.
* Polyfuse placed near battery connector.
* TP4056 output feeds system VCC directly (through fuse if desired).
* On/Off switch disconnects battery positive from system.
* Ground common across all components.
* Include decoupling capacitors:
  + 100 nF ceramic across ATtiny85 VCC–GND.
  + 10 µF electrolytic on main VCC near amplifier.
  + Optional 470 µF at TP4056 output for surge buffering.
* DFPlayer note: Include 1K–2.2K resistor between ATtiny85 TX and DFPlayer RX to ensure reliable UART communication.

### 2.3 Harness / Wiring Plan

Harness E — Pushbutton (Mode / Trigger)  
# From To Signal Color Gauge Length Notes  
E1 Pushbutton Terminal A ATtiny85 PB2 (Pin 7) Button Input (Active LOW) Blue 26 AWG 20–80 mm Use INPUT\_PULLUP  
E2 Pushbutton Terminal B GND Return Black 26 AWG 20–80 mm ---

### 2.4 To-Do

* Verify all electrical connections and pin assignments.

## 3. Software / Firmware

### 3.1 Platform & Environment

* **Platform:** PlatformIO (VS Code)
* **MCU:** ATtiny85 (using MicroCore or equivalent)
* **Language:** C/C++
* **Repository:** GitHub — primary source of truth for firmware, documentation, and history.
* **Local Development:** VS Code repository mirrors GitHub; local repo used for coding, builds, and testing.

### 3.2 Functionality

* **Short Press:** Play random audio clip.
* **Long Press:** Enter volume control mode.
* **LED (if used):** On during playback; blink during mode selection; fade during volume adjustment.

### 3.3 Repository & File Structure

The project uses a dual-repository structure for consistency and redundancy.

| Repository | Location | Purpose |
| --- | --- | --- |
| **Local VS Code Repo** | Development machine | Active coding, testing, and builds. Commits staged before syncing to GitHub. |
| **GitHub Repo** | Online | Central source of truth for code, documentation, and release tracking. |

#### Directory Layout

/mccoy-speaks/  
├── /hardware/ → EasyEDA design files, wiring lists, images  
├── /software/ → PlatformIO project files, firmware source  
├── /audio/ → Voice clips, metadata, conversion tools  
├── /docs/ → Project plan, notes, markdown exports  
│ └── /history/ → Archived documentation versions  
├── LICENSE  
├── README.md  
└── PROJECT\_PLAN.md → Exported version of this document

### 3.4 Development Notes

* Implement button debounce (software or RC network).
* Measure latency and stability of playback.
* Test current draw in idle and active states.
* Plan for safe power-down between plays.

### 3.5 To-Do

* Move initial code version from ChatGPT to GitHub/PlatformIO.
* Learn to use Copilot in VS Code.

## 4. Audio / Media

### 4.1 Source & Format

* Pre-recorded McCoy voice clips gathered and curated.
* Files stored in DFPlayer-compatible format (e.g., 8-bit mono WAV, 22 kHz).
* Stored in /audio/ directory in GitHub repository.

### 4.2 Future Options

* LED flash synchronized with speech.
* Alternate voices or effects.
* Audio feedback for mode changes.
* McCoy voice used to announce startup.

## 5. Documentation & Version Control

### 5.1 Directory Structure

/hardware/ → Schematics, wire lists, EasyEDA exports  
/software/ → Firmware source, PlatformIO project  
/audio/ → Voice clips, metadata  
/docs/ → Project plan, notes, exports

### 5.2 Versioning Strategy

* Maintain current plan in ChatGPT (canvas).
* Export as PROJECT\_PLAN.md to /docs/.
* Commit each export with version tag (vX.Y) and summary.
* Archive old versions in /docs/history/.

### 5.3 Collaboration Notes

* Local storage and GitHub are authoritative.
* ChatGPT used for drafting, merging, and structured editing.

## 6. Next Steps

* Procure hardware.
* Breadboard initial design.
* Test battery charge system.
* Determine if Hallmark figure requires modification to accommodate hardware.