FIRMWARE SPECIFICATION — Noor StoryBox V1

Product: Children's StoryBox (Audio Only — No Screen)

NO screen → all feedback is audio-based.

Microcontroller: ESP32-S3-WROOM-1-N16

General Vision of the Noor Firmware

Audio CODEC: TLV320AIC3204IRHBR

Controlled via I2C

Audio stream via I2S

Class-D I2S Amplifier: MAX98357AETE+T

Headphone Jack: SJ2-3589-4B-SMT-TR with HP_DET (headphone detection)

Firmware implication: audio routing and headphone priority must be managed via HP_DET (e.g. speaker mute, codec / speaker switching, etc.).

MicroSD card slot

Clickable rotary encoder (wheel)

Physical buttons: Play/Pause, Back/Home, Volume+, Volume—

NO screen → all feedback is audio-based.

OVERALL GOAL

Develop a complete firmware for Noor, a screenless children's StoryBox that supports:

- full audio-based navigation
- hierarchical Theme → Sub-theme → Story structure
- audio feedback for every interaction
- story playback from microSD
- a personalized welcome message from parents
- an interactive educational mode (Q&A)
- story file transfer via USB Mass Storage or Bluetooth
- OTA firmware updates
- physical buttons for essential controls

Because the product has no screen, audio feedback is provided only during

content selection and decision-making phases, including:

- navigation through Themes, Sub-themes, and Stories using the rotary encoder

- selection and validation of answers in interactive Q&A stories

Audio feedback is NOT required for all button presses.

Buttons such as Play/Pause, Home/Back, and Volume may operate silently

unless explicitly defined otherwise.

1. SYSTEM ARCHITECTURE

The firmware must:

- operate 100% through audio (no screen)
- rely on a FAT32 microSD card containing:
- theme folders
- sub-theme folders
- audio titles
- story audio files
- interactive story JSON definition files
- a global index (index.json)
- load this structure on boot

- respond instantly to user input
- gracefully handle SD-card removal or corruption

---

2. PERSONALIZED WELCOME MESSAGE (PARENT-RECORDED)

User-facing behavior

- When the StoryBox powers on, it must automatically play:

/system/welcome.wav

- This file is recorded by parents on their phone, then transferred to the StoryBox:

1. via USB Mass Storage

- The StoryBox does not record audio itself.

Technical requirements

At boot:

if /system/welcome.wav exists:

play it immediately

- Playback must be instant (no noticeable delay).
- Volume matches the master system volume.

3. AUDIO NAVIGATION USING ROTARY ENCODER

3.1 Navigation Levels

- Level 1 — Themes

Example:

1. Stories of the Prophets

2. Islamic Education (Qur'an, Hadith)

- Level 2 — Sub-themes

Example inside "Stories of the Prophets":

- Prophet Adam
- Prophet Nûh
- Prophet Ibrahim
- Prophet Muhammad
- Level 3 — Stories

Example inside Prophet Muhammad:

- Story 1 — "Birth of the Prophet"
- Story 2 — "Revelation in Hira"
- Story 3 — "The Hijra"

3.2 Rotary Encoder Behavior

- Scrolling is circular:

- With 3 items → 1 → 2 → 3 → 1 → 2 → 3
- Scrolling left reverses the cycle.
- At each encoder tick, the StoryBox plays an audio title:
- /titles/theme_01.wav
- /titles/muhammad_01.wav
- etc.
- Pressing the encoder → enter next level.

3.3 Requirements

- Low-latency encoder handling
- Minimal debounce
- No audio gaps when switching files
- Title audio volume = master system volume

---

4. STORY PLAYBACK

Audio format

- WAV, 16-bit, 44.1 kHz mono

Structure

Each story consists of:

- a title file → xxx_title.wav
- a story file → xxx_story.wav
- optionally: an interactive script JSON (see below)

Behavior

- Selecting a story and pressing the encoder → play xxx_story.wav
- Pause/Resume with the Play/Pause button

- Exiting with Home button must stop playback cleanly

5. INTERACTIVE EDUCATIONAL MODE (Q&A)

5.1 User experience

Certain educational stories include Q&A steps:

1. The narration plays normally.

2. A question is read:

"What color was it: blue or pink?"

3. The child turns the encoder:

- tick → plays audio: "Blue"
- next tick → "Pink"
- loops continuously

4. The child presses to validate the answer.

5. If the answer is correct:

- play “Correct!” audio
- continue the story

6. If wrong:

- play encouragement audio
- repeat the question

5.2 Story scripting (JSON example)

```
{
  “parts“: [
    { “type“: “story“, “file“: “part1.wav“ },
    {
      “type“: “question“,
      “question“: “q1.wav“,
      “answers“: [
```

```
        { "file": "q1_answerA.wav", "correct": true },
        { "file": "q1_answerB.wav", "correct": false }
      ],
      "correctFile": "q1_correct.wav",
      "retryFile": "q1_retry.wav"
    },
    { "type": "story", "file": "part2.wav" }
  ]
}
```

5.3 Requirements

- Full state machine:
- NARRATION → QUESTION → ANSWER → RESULT → NARRATION
- Fast transitions
- Zero delay between audio sequences

---

6. MICROSD CARD FILE STRUCTURE

Example structure:

```
/system/
    welcome.wav
    low_battery.wav

/stories/
    prophets/
        adam/
            adam_01_title.wav
            adam_01_story.wav
```

muhammad/

muhammad_01_title.wav

muhammad_01_story.wav

muhammad_01.json

education/

faith/

lesson_01_title.wav

lesson_01_story.wav

lesson_01.json

/index.json

Firmware responsibilities

- Load /index.json on boot

- Build internal lists of:
- themes
- sub-themes
- stories
- Detect new story packs and update navigation accordingly

---

7. FILE TRANSFER (USB & BLUETOOTH)

7.1 USB Mass Storage Mode (MSC)

- When plugged into a PC:
- Noor appears as a USB drive
- Parents can drag & drop files
- Upon disconnection:
- Firmware rescans SD

- Reloads index.json
- Updates navigation

7.2 Bluetooth File Transfer (via mobile app)

- App sends audio/story packs through Bluetooth
- Firmware writes them to the correct directory
- Must support:
- file integrity verification
- resume on interruption
- automatic integration in index.json

---

8. BATTERY MONITORING & AUDIO WARNING

- Read battery voltage using ADC

- Default threshold: 3.3V

- If below threshold:

- play /system/low_battery.wav

9. OTA FIRMWARE UPDATE

V1 (Factory / initial release): firmware is flashed via USB (factory flashing / recovery).

The firmware must include an OTA-ready partition scheme so that future updates (V2+) can be deployed via Wi-Fi OTA.

Wi-Fi OTA is not required to be user-facing in V1 (can be disabled or hidden), but the OTA infrastructure must be present and validated.

The developer must ensure that V1 is OTA-ready (dual-slot partitioning + rollback safety), enabling future Wi-Fi OTA updates without requiring USB reflashing for end users.

10. BOOT SEQUENCE

1. Initialize audio + SD quickly (goal: <2 seconds)
2. Play welcome.wav if available
3. Load index.json
4. Enter Theme-selection mode with encoder

11. PHYSICAL BUTTON BEHAVIOR (DETAILED)

Play / Pause

- During story playback: toggle pause/resume
- Ignored in menus

Home / Back

- Immediately return to Theme Level
- Stop any audio playback cleanly

Volume + / Volume −

- Adjust global volume:
- story audio
- title feedback
- questions
- correct/wrong answers
- welcome message

---

12. DELIVERABLES

The developer must provide:

- Full source code (.ino / .cpp / .h)

- Full documentation:
- setup
- build instructions
- flashing procedure
- A compiled .bin ready for factory flashing

13. DEVELOPMENT PRINCIPLES

- Clean, modular, well-commented code
- Robust state machine for navigation and Q&A
- Non-blocking code (no long delays)
- Fast encoder interrupt handling
- Reliable SD and USB MSC integration
- Smooth audio transitions (no popping/clicking)

END OF SPECIFICATION — Noor StoryBox Firmware V1