UART3 Slick Prompt Integration (Feature Branch)

This document summarizes the UART3 user-interaction revamp performed on branch `feature/uart3-slick-prompt`.

Goals (must not regress)

- Precise PWM generation at dynamic duty regimes.
- Robust session handling based on DTR detection on PQ1.
- UART0 (ICDI) remains the diagnostics channel (memory inspection, allocator feedback, stack/heap checks).
- UART3 becomes the final-user interface: stable, colorful prompt, with line editing.

What changed

UART3 interaction moved out of `main.c`

Previously, UART3 used an ISR (`USERUARTIntHandler`) that echoed bytes and accumulated a command line in a buffer in `main.c`, then the foreground parsed the line.

Now:

- `main.c` starts and ends UART3 "sessions" using DTR (PQ1) as before.
- When DTR indicates a session is active, `main.c` calls:
  - `cmdline_init()`
  - `cmdline_run_until_disconnect()`
- The cmdline module prints a welcome and prompt immediately when the DTR session starts (no "blank terminal" on connect).

UART3 slick prompt behavior

The UART3 prompt/CLI is implemented in `cmdline.c` / `cmdline.h`, and is derived from the same prompt-management approach used in `otherC/ESP32_SLICKUART_forREFERENCE.c`:

- ANSI colors for prompt, welcome, OK responses, and errors.
- Prompt de-duplication (prevents repeated > spam).
- Line editing:
  - Backspace / DEL erases characters correctly.
  - Backspace is blocked at the start of the input buffer, so the user cannot delete the prompt prefix.
  - Ctrl-U clears the current input line.
- Uppercase-as-you-type to simplify command matching.

PWM path preserved

- PWM setup and pulse updates remain the same.
- `set_pwm_percent()` was given external linkage so the cmdline module can call it.

UART0 diagnostics preserved

- UART0 (ICDI) remains the diagnostics channel.
- Diagnostic helpers in `diag_uart.c` still use UART0.

Current commands (UART3)

- `PSYN n`
  - Sets PWM duty to `n` percent, with enforcement of range 5..96.
- `HELP` or `?`
  - Prints available commands.

Implementation notes

- UART3 is handled by the cmdline module via polling (UART3 interrupts are disabled in `main.c`).
- `USERUARTIntHandler()` is kept as a safe no-op in case UART3 interrupts get enabled accidentally.
- `cmdline.c` uses the project's `ctype_helpers.h` (`my_isspace`, `my_toupper`) to avoid toolchain/newlib ctype table dependencies.

Build

From the project folder:

- Build:

  ```
  make
  ```

- Flash + run UART capture workflow (if your setup is already configured):

  ```
  make auto
  ```

PDF generation

This markdown is intended to be printable via:

- `tools/md2pdf.sh docs/UART3-Slick-Prompt.md docs/pdfs/UART3-Slick-Prompt.pdf`