Nova Watchdog Timer (Dusky-Petrel Extension)

This document defines the **Watchdog Timer** peripheral for Dusky-Petrel. The watchdog provides a safety mechanism for detecting stalled or unresponsive software. If the timer expires, it performs a configured action (interrupt, halt, reset, or none). The watchdog begins life *inactive* and only counts when explicitly armed. To keep the watchdog happy, the guest software may periodically **pet** it—reloading the timeout and reassuring it that the system is still healthy.

Configuration (Host Side)

```
[devices.watchdog]
enabled = false          ; default: inactive until armed by guest or monitor
device_code = 0o14       ; configurable Nova I/O channel
timeout_ms = 5000        ; default timeout
repeat = false           ; one-shot by default
action = "interrupt"    ; interrupt | halt | reset | none
```

Registers and Flags (Guest View)

```
Register A — Timeout (milliseconds)
  DOA → write new timeout value (low 16 bits)
  DIA → read current configured timeout

Register B — Control / Mode
  bit 0: ENABLE        (1 = watchdog may run when armed)
  bit 1: REPEAT        (1 = periodic / repeating)
  bits 2–3: ACTION     (0=none, 1=interrupt, 2=halt, 3=reset)
  bit 4: PET           (1 = reload timer and keep the dog happy)
  bit 5: CLEAR_FIRED   (1 = clear FIRED/DONE state)

Register C — Status (read■only)
  bit 0: FIRED   (1 if watchdog expired and not yet cleared)
  bit 1: ACTIVE  (1 if currently counting)
  bit 2: REPEAT  (mirror of control)
  bits 3–4: ACTION (mirror of control)
```

Busy / Done Semantics

```
BUSY = 1 while the watchdog is armed and counting.
BUSY = 0 when disabled or expired.
DONE = 1 when the watchdog has fired (expired) and not yet cleared.
DONE is cleared by CLEAR_FIRED or device Clear signal.

Skip behavior:
  SKPBN → skip while BUSY=1 (timer running)
  SKPBZ → skip when BUSY=0
  SKPDN → skip once DONE=1 (expired)
  SKPDZ → skip while not yet expired
```

Signals (NIO Start / Clear / Pulse)

```
Start → arm the watchdog if ENABLE=1 and timeout_ms>0.
Clear → disarm and clear BUSY/DONE/FIRED.
Pulse → optional: force an immediate fire event.
```

Internal Behavior (Summary)

```
* The watchdog is inactive on reset (ENABLE=0, ACTIVE=0, FIRED=0).
* Arming reloads the countdown and sets BUSY=1, ACTIVE=1.
* When the deadline passes, the watchdog FIRES:
    - BUSY=0, DONE=1, FIRED=1
    - Perform configured ACTION (interrupt / halt / reset / none)
* If REPEAT=1 and ENABLE=1, it automatically re■arms after firing.
```

Petting the Watchdog

```
To reassure the watchdog that the system is still healthy, the guest periodically <b>pets</b> it by writ

Example:
  LDA   0, #0b00010101   ; ENABLE=1, REPEAT=0/1 as configured, PET=1
  DOA   WDT B            ; pet the watchdog — resets the countdown
```

Example (Nova Assembly)

```
; Configure 5■second watchdog and pet it in the main loop
        LDA     0, #5000          ; timeout in ms
        DOA     WDT A
        LDA     0, #0b00001101    ; ENABLE=1, ACTION=reset, PET=1 (arm + first pet)
        DOA     WDT B

MAIN:
        ; ... do useful work ...
        LDA     0, #0b00010001    ; PET=1, ENABLE=1 (same mode)
        DOA     WDT B             ; pet to keep the dog happy
        JMP     MAIN
```

Compatibility Notes

```
* The watchdog device is simulator■specific and disabled by default.
* Legacy Nova programs run unchanged unless they access the watchdog channel.
* Behavior is deterministic and suitable for interrupt■driven systems.
```