Three weapon fencing scoring box	Firmware
STM32F401 front end	Doc 11 Rev B 2026-08-29

Files

File	Runs on	What it is
scoring_logic.h / .cpp	both	All scoring decisions. Plain C++, no Arduino dependency, so it can be tested on a PC
test_scoring_logic.cpp	PC	25 unit tests over the state machine
scoring_box_f401.ino	Black Pill	Plumbing: ADC, drive phases, lamps
hit_simulator_nano.ino	Nano	Generates contacts of exact duration

Putting the decisions in a plain C++ file is the point. Fencing logic is all about microsecond timing, and timing bugs are close to impossible to find by waving a sword at a box. On a PC the same code runs against synthetic contacts of exact length, in a second, repeatably.

Build and run the tests with: `g++ -std=c++17 -O2 -o test test_scoring_logic.cpp scoring_logic.cpp && ./test`
No g++ installed? The zig toolchain from pip works identically: `pip install ziglang` then `python -m ziglang c++ -std=c++17 -O2 -o test test_scoring_logic.cpp scoring_logic.cpp`

Sabre wiring reality — the Rev B correction

Rev A treated a sabre weapon line joined to its own C line as an equipment fault and as a reason to suppress hits. That was wrong, and it made sabre mode unusable with a real weapon. The FIE Material Rules settle it:

> “The two sockets of the bodywire plug must be in direct contact with the body of the guard, making a closed electrical circuit through the bodywire, the spool and the cable connecting the spool to the scoring apparatus.” — FIE Material Rules, m.24.4 (sabre guard)

So on a healthy sabre the B and C lines are permanently commoned through the guard, at *both* weapons. Three consequences, all in the Rev B code:

Change	Why
Sabre fault is now a **floating** weapon line	W at the rail means the cord, socket or weapon is disconnected. W joined to own C is the normal resting state, never a fault.
toOwnGround and toOppGround no longer veto sabre hits	Own ground is always joined; blade-to-blade contact joins the opponent's B *and* C at once. Cuts delivered during blade contact are decided by the whip-over window, not a blanket veto.
LEVEL_LOW_MAX lowered from 900 to 600	The whip-over pile-up (drive + own C + opponent's B, C and lamé) is one pull-up against four pulldowns = 1/5 rail ≈ 819 counts, which the old 900 floor read as “not joined”. Untouched sense lines read near zero, far below 600.

What the tests establish — 25 checks, all passing

Test	Result
Mid-band sanity: sabre pile-up level (819) joined; near-ground (300) not	2 checks pass
Foil 13 ms on-target contact / 16 ms contact	rejected / registers
Foil floor hit, hit on opponent guard	no light at all
Foil point on insulating cloth / 8 ms cloth slide onto lamé	off target / ON target only
Epee 1 ms contact / 4 ms contact	rejected / registers
Epee floor hit on a metal strip	no light
Epee second hit at 30 ms / at 60 ms	double / locked out
Sabre 1.2 ms lamé contact, with B-C commoned throughout	registers
Sabre at rest (W joined own C) 4 s / floating weapon line 4 s	no fault, no light / fault indicated
Sabre whip-over at 8 ms / whip landing 1 ms after blades part / cut after 25 ms	suppressed / suppressed / still scores
Sabre hit on opponent guard, blade on piste	no light
Foil / epee / sabre lockout gap vs spec (300 / 45 / 170 ms)	error 0 µs, all three
Epee stuck point held 4 s	fault indicated

Three deliberate departures from the common designs

First, the piste and both C lines are read. In the widely copied Arduino sketch the ground pins are declared and wired but never sampled, so every epee hit on a metal strip scores. Here a floor hit produces nothing in all three weapons.

Second, lockout is measured from the moment a hit registers rather than from the start of contact. Measuring from contact start shortens the window by the contact time, which for foil is 14 ms of a 300 ms budget.

Third, nothing blocks. The reference sits in delay() for 3.1 seconds after every hit and is deaf during that time; this latches the lamps and keeps scanning.

Whip-over: mostly confirmed, one residual interpretation

The Material Rules (sabre annexe) confirm what Rev A only assumed: hits through the blade register between 0 and 4 ms of blade contact and are prevented between 4 and 15 ms (+5 ms), **measured from the start of blade contact** — exactly how the code measures it. The one remaining interpretation is the 2 ms tail that keeps rejecting a whip landing just after the blades part. Compare against a known good box before trusting it.

Bring-up order

- [ ] Run the PC tests. They should report 25 passed before you touch hardware.
- [ ] Flash the F401 with nothing connected. Watch the serial scan rate; expect tens of thousands per second. If it reads a few thousand, the fast ADC path is not working.
- [ ] Short a weapon node to a sense node with a jumper and confirm both read mid rail on the serial output before trusting any weapon logic.
- [ ] Wire the Nano simulator through its 2.2 k resistors (share grounds), hold the box's mode button while powering it up to freeze the drive phase, then run the foil suite from the simulator and watch the lamps.
- [ ] Run the sabre suite. The first test holds the commoned B-C resting state for three seconds: no lamp and no fault may appear.
- [ ] Put the logic analyzer on the lamp pins and the simulator trigger pin, and measure the real lockout. It should match the numbers above to within one scan period.
- [ ] Only then connect real weapons.