EMBEDDED HACKING

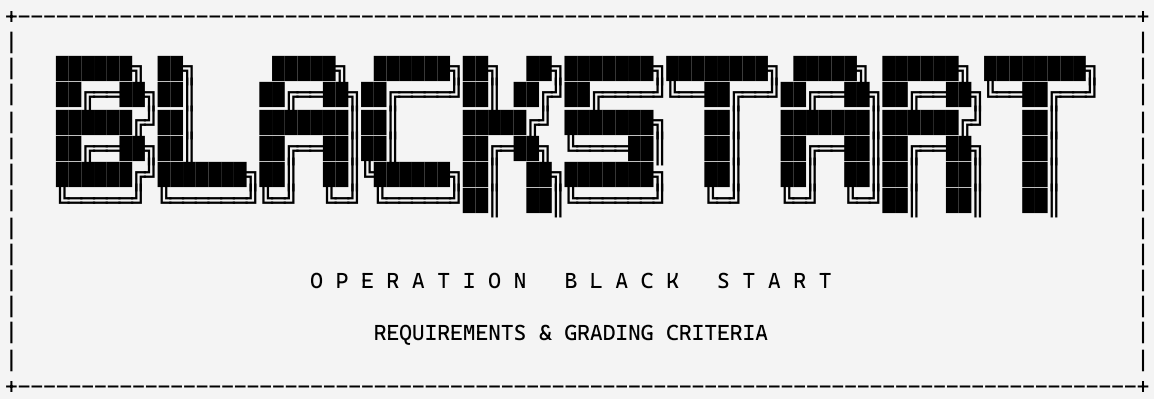
Reverse Engineering the Raspberry Pi Pico 2

CTF Challenge: Operation Black Start

ARM Cortex-M33 • GDB • Ghidra

September 9, 2026

Project Requirements & Grading Criteria



Project Overview

Students are the reverse-engineering reserve team called in after WorldGrid Compact's emergency firmware build shipped a miscompiled safety threshold and a false status string to its GRID-7 relay fleet. Students reverse engineer the supplied RP2350 image with Ghidra, locate two real defects, patch them directly in the binary, export a corrected image, flash it to real hardware, and verify the corrected behavior on a physical Pico 2 — the same workflow used in the FINAL projects.

Students must use only Weeks 1-3 concepts: ARM registers and stack behavior, UART output, GDB, Ghidra static analysis and binary patching, vector tables, reset startup, XIP, and Thumb addressing. No LED, relay, sensor, display, or later-week peripheral-control task is part of this CTF.

The challenge is separate from all FINAL projects and contains no FINAL-project answer, constant, address, bug, or patch.

Learning Objectives

Students will demonstrate the ability to:

1. Capture a clean UART baseline before modifying the binary.
2. Decode an RP2350 vector table and initial stack pointer.
3. Explain the Thumb bit in a reset-handler pointer.
4. Trace a flash string into the UART output call and argument register.
5. Locate a miscompiled immediate value used in **multiple** locations and understand why every occurrence must be patched.
6. Correctly compute a patched immediate value even when the compiler has transformed the original comparison (e.g., `<` optimized to `<=`).
7. Patch a string literal in flash without corrupting adjacent data.
8. Export a patched binary, convert it to UF2, and verify the fix on real hardware.

9. Recover a quarantined data frame from flash as static evidence.

Deliverables Checklist

#	Deliverable	Format	Task
1	Ghidra project screenshot (project name, processor, base address)	PNG/JPG	Task 1
2	`main()`, status-loop, and vector-table address table	Inside `CTF-01-Answers.md`	Task 1
3	Bug #1 analysis: both addresses, original/patched bytes, immediate-value reasoning	Inside `CTF-01-Answers.md`	Task 2
4	Bug #2 analysis: string address, original/patched bytes, character-by-character mapping	Inside `CTF-01-Answers.md`	Task 3
5	Recovered dispatch frame and its address	Inside `CTF-01-Answers.md`	Task 4
6	`CTF-01_fixed.bin` — exported patched binary	BIN file	Task 5
7	`CTF-01_fixed.uf2` — UF2-converted binary	UF2 file	Task 5
8	Verification transcript: corrected UART output on real hardware	Inside `CTF-01-Answers.md`	Task 5
9	Summary table of all patches (address, original bytes, patched bytes)	Inside `CTF-01-Answers.md`	Task 5
10	Written reflection (two short answers)	Inside `CTF-01-Answers.md`	Task 6

Required Tools and Equipment

Tool	Purpose	Required For
Raspberry Pi Pico 2	Isolated target	All tasks
3.3 V USB-UART adapter	UART capture	Tasks 1, 5
Serial monitor	Observe output	Tasks 1, 5
Ghidra	Static analysis and binary patching	Tasks 1-4
Python (`uf2conv.py`)	UF2 conversion	Task 5
`CTF-01.bin` and `CTF-01.uf2`	Supplied artifacts	All tasks

UART settings: **115200 baud, 8 data bits, no parity, 1 stop bit**.

The instructor-issued artifact hashes are:

```
CTF-01.bin  6FD296F7A85F243FB26BF6BFFCBEAB26815FD8915101A81F72069063A5635E5A
CTF-01.uf2  980F04369C23AD32A063DFE18DE5AF08DF3830138FC7E7898B1F011B4F5E1D9D
```

Task 1: Setup and Initial Analysis — 15 points

Criterion	Points	Full Credit	Partial Credit	No Credit
Ghidra project setup	3	Screenshot shows correct project name, ARM Cortex 32-bit little endian, base `0x10000000`	One item off	Not set up
`main()` and status-loop addresses	4	Both addresses correctly documented	One correct	Neither found
Vector table	4	Correct base, initial SP, reset pointer	One missing	Not found
Thumb addressing	4	Correctly clears bit 0 to identify the real instruction address	General explanation	Incorrect

Task 2: Find and Patch Bug #1 — Miscalibrated Safety Threshold — 30 points

What to find: the frozen reading is compared against a miscompiled safety constant at **two separate addresses** — once for the operator-facing status and once for the automated dispatch decision.

Criterion	Points	Full Credit	Partial Credit	No Credit
Found location A	5	Correct address and original instruction/bytes documented	Address off	Not found
Found location B	5	Correct address and original instruction/bytes documented	Address off	Not found
Correct immediate-value reasoning	8	Explains the `<` vs `<=` compiler transform and derives the correct patched immediate	Correct value, no reasoning	Wrong value
Patched location A	4	Grader verifies the byte change	Wrong byte	Not patched
Patched location B	4	Grader verifies the byte change	Wrong byte	Not patched
Explained why both must be patched	4	Clear explanation of duplicated/independent comparisons	Vague	Missing

Task 3: Find and Patch Bug #2 — The False Signal Banner — 20 points

Criterion	Points	Full Credit	Partial Credit	No Credit
Found the string	5	Correct address, found via Ghidra Defined Strings or hex inspection	Approximate	Not found
Patched correctly	8	All required bytes changed, string length preserved, grader verifies boot output	Correct text but wrong bytes documented	Wrong length or corrupted data
Character-by-character documentation	4	Original vs. patched bytes for every changed character	Partial	Missing
Explained the danger of a hardcoded status word	3	Clear, specific reasoning	Generic	Missing

Task 4: Recover the Quarantined Dispatch Frame — 10 points

Criterion	Points	Full Credit	Partial Credit	No Credit
Found the hidden frame	6	Correct address and full recovered text	Partial text	Not found
Explained why it is not transmitted	4	Clear static-analysis explanation	Vague	Missing

Students must **not** patch this value; it is evidence only.

Task 5: Export and Verify — 20 points

Criterion	Points	Full Credit	Partial Credit	No Credit
Exported patched binary	4	Valid `CTF-01_fixed.bin` submitted	Corrupted	Not submitted
Converted to UF2 correctly	4	Valid `CTF-01_fixed.uf2`, correct base/family flags	Wrong flags	Not submitted
Hardware verification	8	Grader confirms corrected UART output: `GRID STATUS: CRITICAL`, `DISPATCH PATH: HELD`, corrected signal word	Only some lines corrected	No verification
Summary table of all patches	4	Complete table with addresses and before/after bytes	Missing entries	No table

Task 6: Written Reflection — 5 points

Criterion	Points	Full Credit	Partial Credit	No Credit
"Rushed build" is not an excuse	2	Specific, grounded reasoning	Generic	Missing
Engineering practice per bug	3	Names one concrete, relevant practice for **each** bug	Names one for only one bug	Missing

Recommended Answer File Structure

```
# Operation Black Start – Incident Report
## 1. Scope and Artifact Integrity
## 2. Ghidra Setup and Boot/Vector Table
## 3. Bug #1 – Miscalibrated Safety Threshold
## 4. Bug #2 – False Signal Banner
## 5. Recovered Dispatch Frame
## 6. Export and Hardware Verification
## 7. Patch Summary Table
## 8. Written Reflection
```

⚠ Common Pitfalls

Pitfall	Consequence	Avoidance
Patching only one of the two threshold locations	Half the fleet's telemetry still lies	Search for **every** occurrence of the wrong immediate
Assuming the threshold immediate equals the safety limit directly	Off-by-one patch, wrong behavior	Check whether the compiler used `<` or `<=` semantics
Replacing a string with a different length	Corrupts adjacent flash data	Count bytes before patching
Treating an odd vector address as invalid	Thumb analysis fails	Explain bit 0
Skipping hardware verification	Patch is unproven	Flash and capture real UART output
Modifying the quarantined dispatch frame	Destroys required evidence	Recover it, do not patch it

Reference Memory Map

Region	Address	Purpose
Bootrom	`0x00000000`	Immutable boot code
Flash/XIP	`0x10000000`	Vector table, code, constant strings
SRAM	`0x20000000`	Stack and writable state

Safety and Academic Integrity

Use only the supplied Pico 2 and firmware. Do not connect the exercise to an operational grid, water plant, public network, military system, or third-party device. This is a controlled, isolated educational exercise.