Memfault

How Bond Home Built In-House Observability & Why They Switched to Memfault

30th October 2024

Speakers and Agenda

Agenda

- Memfault Intro
- Building Observability with Bond Home
- Q & A

Tyler Hoffman

Co-Founder and VP of Developer Experience

Memfault

Chris Merck

CTO & Co-Founder

bond

Building embedded devices is hard…

Finding it hard to identify when faults occur on devices in the field?

Don’t have the data you need to root cause issues effectively when they are discovered?

Fleet wide update rollouts feel stressful and risky?

Building **reliable** embedded devices is **really hard**.

You are not alone…

50.3%
of organizations take **more than a week** to find defects in the field.

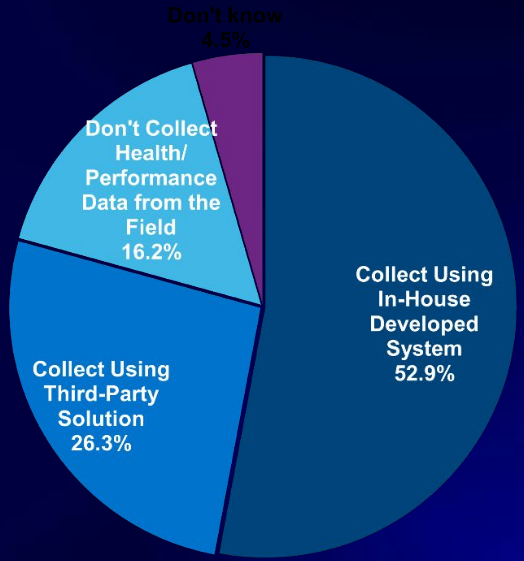
Up to 3 months
engineering time spent **fixing bugs** per year.

44.1%
of organizations take **more than a week** to deploy fixes in the field.

***VDC Research**

And so teams build tools to collect data from their devices…

3/4> of teams collect performance and reliability data from devices in the field.



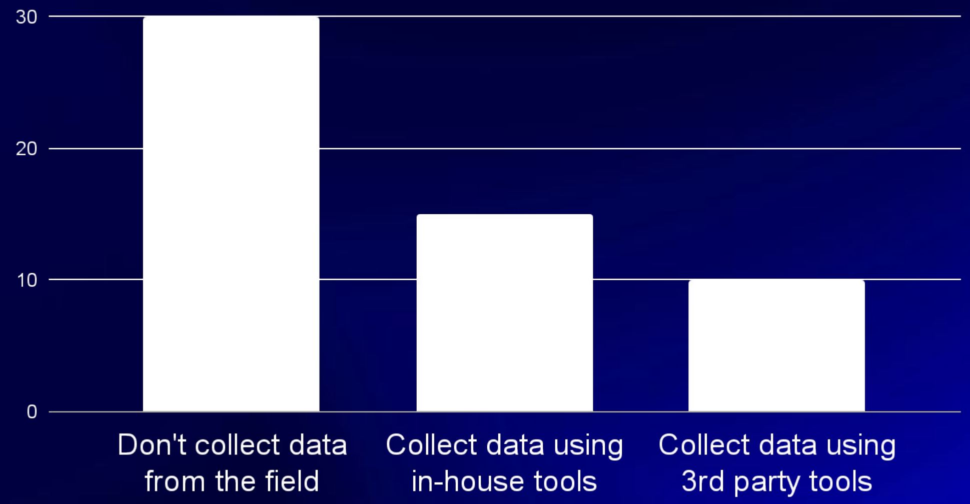
***VDC Research**

Memfault

And they save a lot of time fixing issues…

50%> faster to fix issues happening in the field.

Engineering hours to remediate each SW defect



***VDC Research**

Memfault

But teams face a choice - build it yourself or use a 3rd party

Building & Buying Observability

some considerations for build-vs-buy from experiences at Bond Home

Chris Merck — Co-founder & CTO

bond

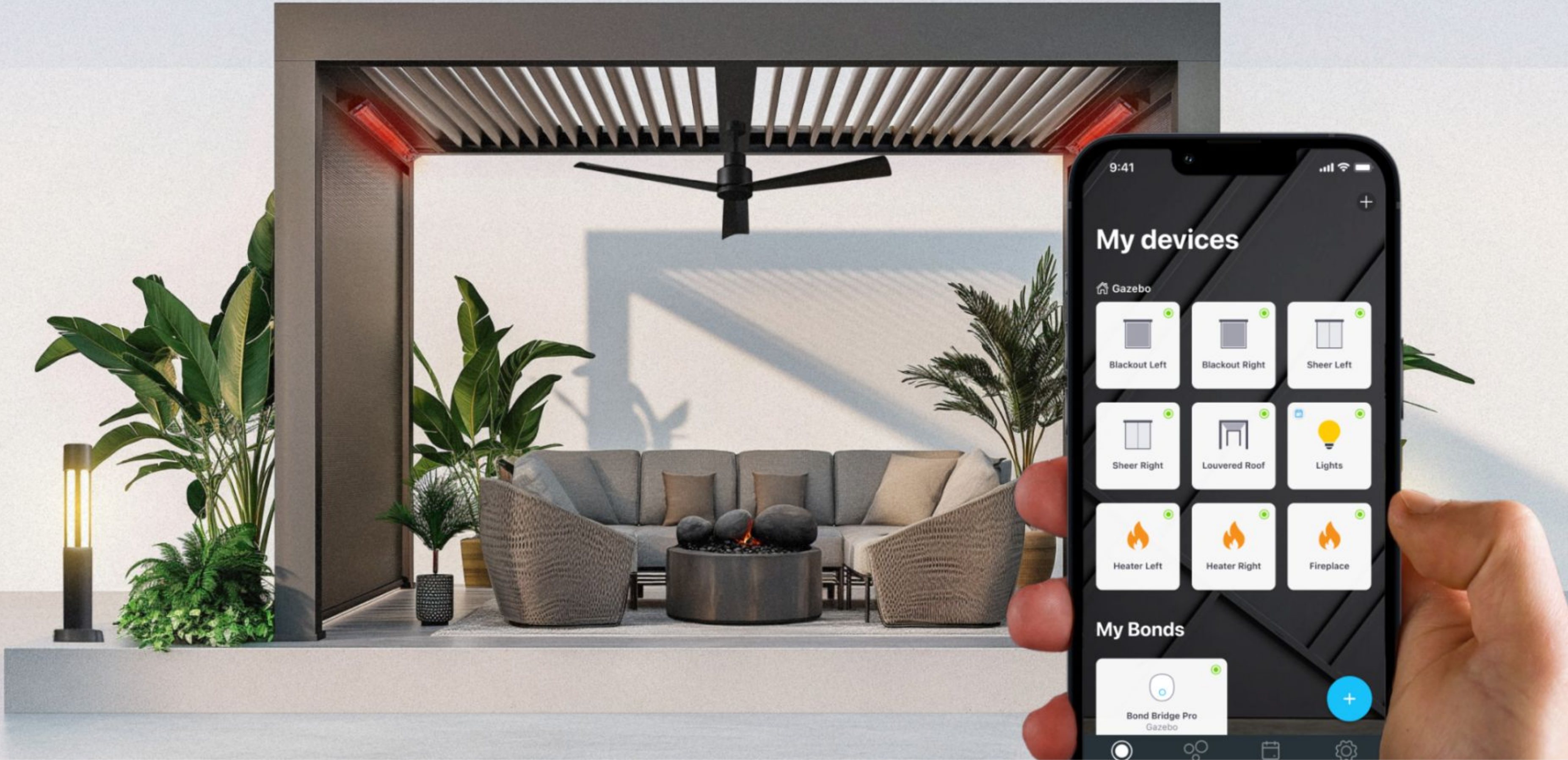
bondhome.io

Who is Bond?

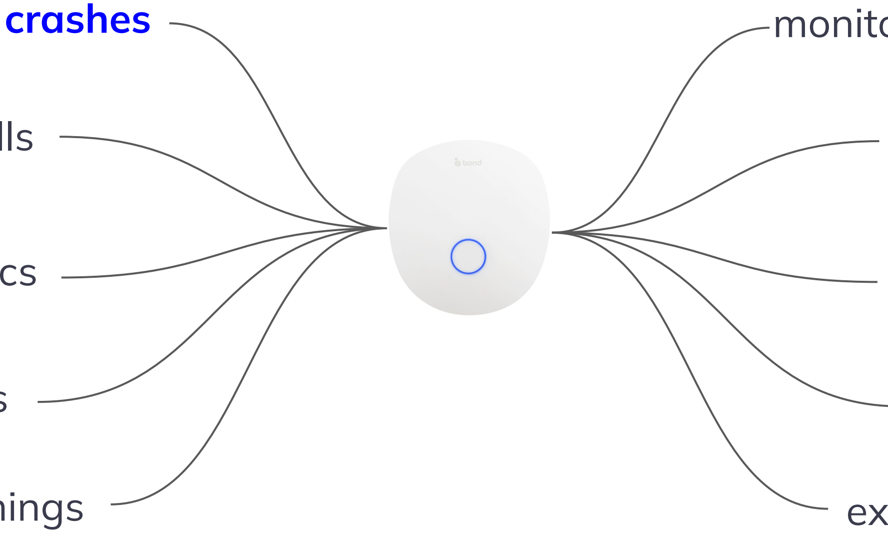
- Headquartered in New Jersey.
- ~12-person technical team in Santa Catarina, Brazil.
- *thoughtfully connecting under-appreciated appliances*
- We build:
    - RF-to-WiFi bridges
    - we power smart ceiling fans for most USA brands
    - (coin-cell-powered) smart remote controls
    - (battery-powered, RF-connected) motorized shades
    - control systems for outdoor living spaces
- single firmware codebase for all products
- mix of STM32 & ESP32 platforms

bondhome.io

most firmware crashes → user notices a "glitch"



What & When to Observe



What

crashes

API calls

heath metrics

serial logs

errors & warnings

When

monitoring an update

specific support ticket

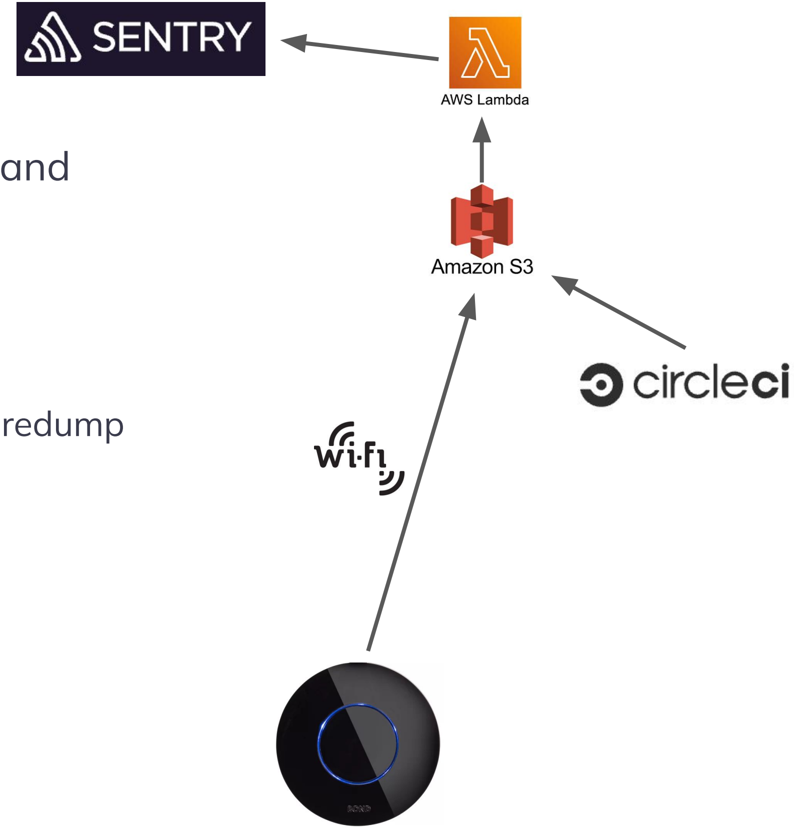
frequent crash

automated test failures

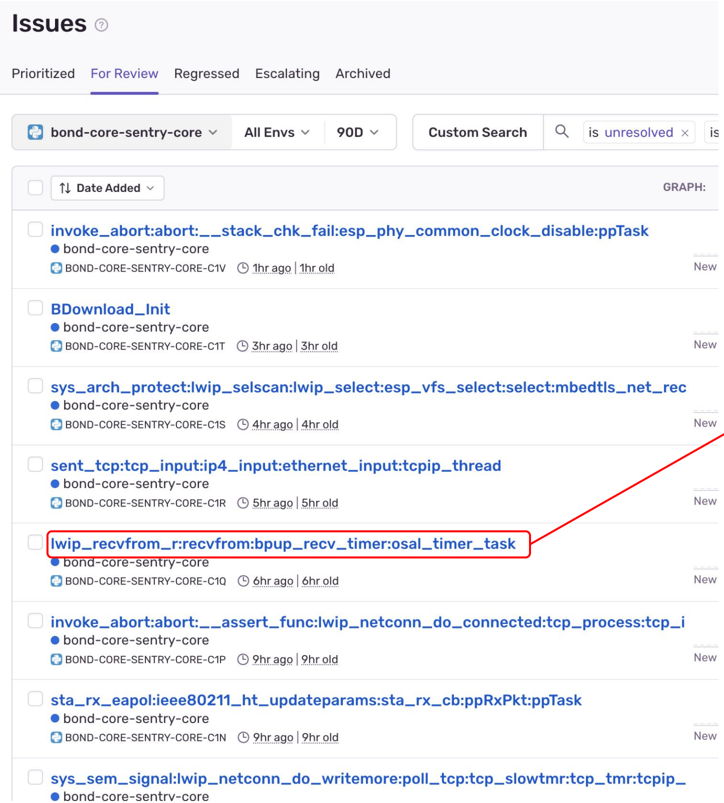
exploring qualityland

Homegrown Crash Analysis

- upload all debug symbols for every build from CI (and developer machines)
- upload coredump to S3
- trigger a Lambda function which:
  - runs modified esp python script to extract actual coredump
  - loads symbols and coredump into xtensa gdb
  - print backtrace of running thread
  - process it down to something repeatable
- upload to Sentry



It Works -- but is it useful?



Issues

Prioritized | **For Review** | Regressed | Escalating | Archived

bond-core-sentry-core | All Envs | 90D | Custom Search | is unresolved × | is

□ Date Added | GRAPH:

□ **invoke_abort:abort:__stack_chk_fail:esp_phy_common_clock_disable:ppTask**
● bond-core-sentry-core
BOND-CORE-SENTRY-CORE-C1V | 1hr ago | 1hr old | New

□ **BDownload_Init**
● bond-core-sentry-core
BOND-CORE-SENTRY-CORE-C1T | 3hr ago | 3hr old | New

□ **sys_arch_protect:lwip_selscan:lwip_select:esp_vfs_select:select:mbedtls_net_rec**
● bond-core-sentry-core
BOND-CORE-SENTRY-CORE-C1S | 4hr ago | 4hr old | New

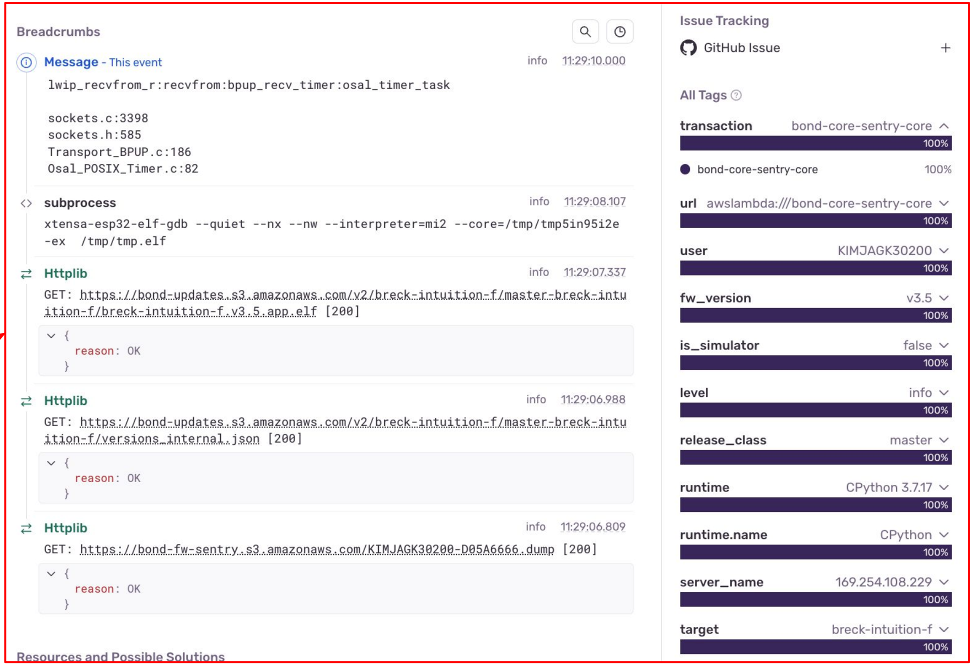
□ **sent_tcp:tcp_input:ip4_input:ethernet_input:tcpip_thread**
● bond-core-sentry-core
BOND-CORE-SENTRY-CORE-C1R | 5hr ago | 5hr old | New

□ **lwip_recvfrom_r:recvfrom:bpup_recv_timer:osal_timer_task**
● bond-core-sentry-core
BOND-CORE-SENTRY-CORE-C1Q | 6hr ago | 6hr old | New

□ **invoke_abort:abort:__assert_func:lwip_netconn_do_connected:tcp_process:tcp_i**
● bond-core-sentry-core
BOND-CORE-SENTRY-CORE-C1P | 9hr ago | 9hr old | New

□ **sta_rx_eapol:ieee80211_ht_updateparams:sta_rx_cb:ppRxPkt:ppTask**
● bond-core-sentry-core
BOND-CORE-SENTRY-CORE-C1N | 9hr ago | 9hr old | New

□ **sys_sem_signal:lwip_netconn_do_writemore:poll_tcp:tcp_slowtmr:tcp_tmr:tcpip_**
● bond-core-sentry-core



Breadcrumbs

Message - This event | info | 11:29:10.000

```
lwip_recvfrom_r:recvfrom:bpup_recv_timer:osal_timer_task

sockets.c:3398
sockets.h:585
Transport_BPUP.c:186
Osal_POSIX_Timer.c:82
```

subprocess | info | 11:29:08.107

```
xtensa-esp32-elf-gdb --quiet --nx --nw --interpreter=mi2 --core=/tmp/tmp5in95i2e
-ex  /tmp/tmp.elf
```

Httplib | info | 11:29:07.337

```
GET: https://bond-updates.s3.amazonaws.com/v2/breck-intuition-f/master-breck-intu
ition-f/breck-intuition-f.v3.5.app.elf [200]
{
  reason: OK
}
```

Httplib | info | 11:29:06.988

```
GET: https://bond-updates.s3.amazonaws.com/v2/breck-intuition-f/master-breck-intu
ition-f/versions_internal.json [200]
{
  reason: OK
}
```

Httplib | info | 11:29:06.809

```
GET: https://bond-fw-sentry.s3.amazonaws.com/KIMJAGK30200-D05A6666.dump [200]
{
  reason: OK
}
```

Resources and Possible Solutions

Issue Tracking

GitHub Issue +

All Tags

transaction	bond-core-sentry-core	100%
● bond-core-sentry-core		100%
url	awslambda:///bond-core-sentry-core	100%
user	KIMJAGK30200	100%
fw_version	v3.5	100%
is_simulator	false	100%
level	info	100%
release_class	master	100%
runtime	CPython 3.7.17	100%
runtime.name	CPython	100%
server_name	169.254.108.229	100%
target	breck-intuition-f	100%

Challenges **building** the tool

- new fw crashes, old fw uploads, does not decode
    - Espressif's coredump (at that time) did not include a unique build ID for the crashing image!
    - so we had to augment the coredump format to include user data, especially a unique build ID
    - this required wrapping / modifying their coredump scripts, and writing tricky code in the crash handler
- security
    - non-trivial embedded resources required to perform and HTTPS upload in parallel to the MQTT/TLS connection, so it is tempting to do it unsecured
    - breaking into chunks over MQTT was considered but would have expanded the project considerably (where to store them while they are being assembled?)
- cross-disciplinary: embedded & backend skills required
    - collaboration is good for teambuilding, but it is also a resource drain as multiple team members are needed to debug and maintain indefinitely

Challenges **using** the tool

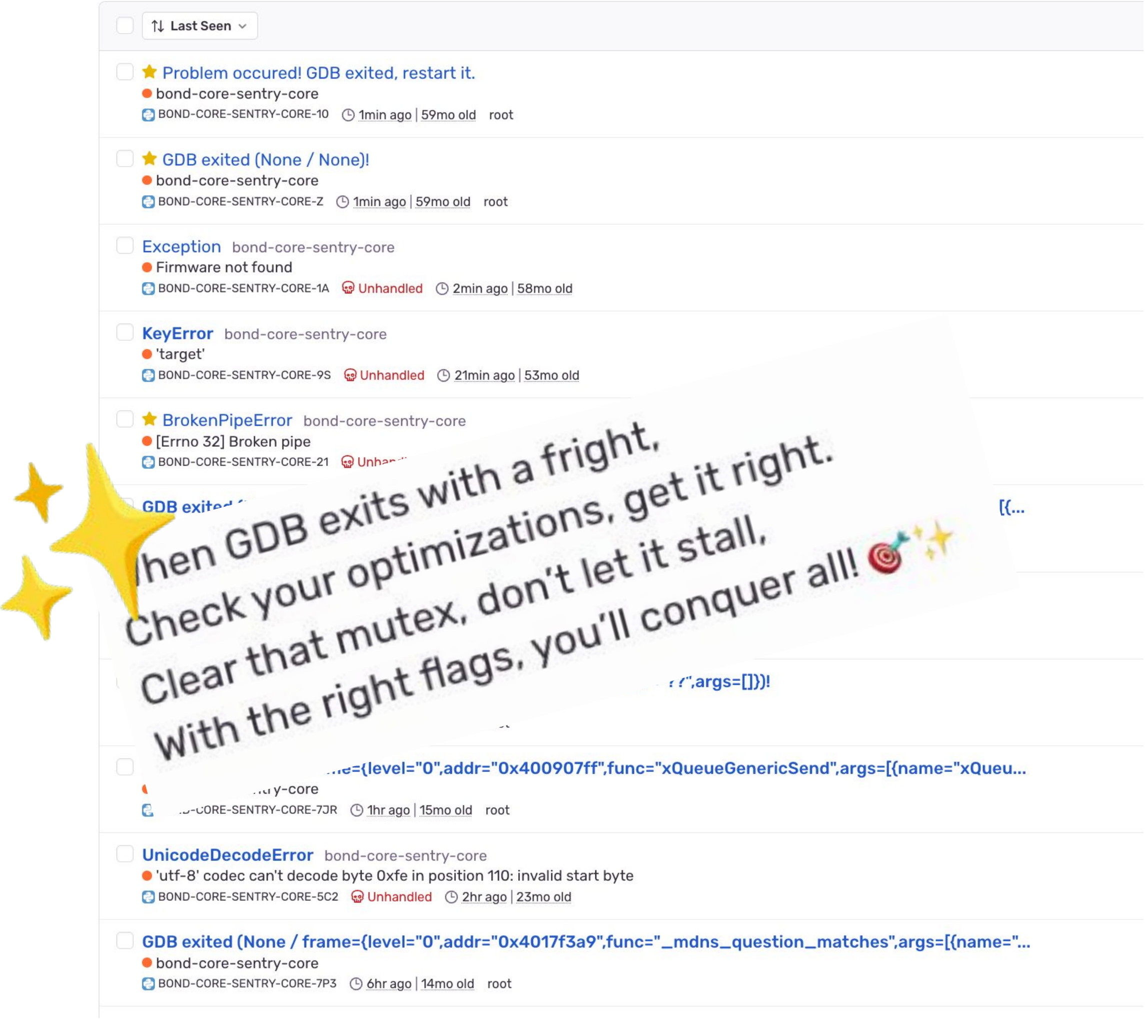
- can only see active thread
- cannot see local / global variables
- loading into gdb manually required for non-trivial investigations
  - high activation energy!
- lacks context
  - recent serial logs
  - API calls
  - errors & warnings
  - we built ways of observing these, but they are all disperate
- single platform
  - we were in the process of transitioning from a Linux/MIPS platform, and we could have benefitted from having observability on the existing fleet.
  - adding a second platform would be a whole layer of abstraction and more embedded work

Meta Issues

Over time, more and more of the coredumps failed to decode, cluttering Setry with issues about the tool, rather than the product.

- Python script aborts
- out of memory in Lambda
- gdb crashing...

Thankfully Sentry can provide cute AI couplets with "suggestions".



□ Last Seen

□ Problem occured! GDB exited, restart it.
bond-core-sentry-core
BOND-CORE-SENTRY-CORE-10 1min ago | 59mo old root

□ GDB exited (None / None)!
bond-core-sentry-core
BOND-CORE-SENTRY-CORE-Z 1min ago | 59mo old root

□ Exception bond-core-sentry-core
Firmware not found
BOND-CORE-SENTRY-CORE-1A Unhandled 2min ago | 58mo old

□ KeyError bond-core-sentry-core
'target'
BOND-CORE-SENTRY-CORE-9S Unhandled 21min ago | 53mo old

□ BrokenPipeError bond-core-sentry-core
[Errno 32] Broken pipe
BOND-CORE-SENTRY-CORE-21 Unhan

GDB exite [{...

",args=[]})!

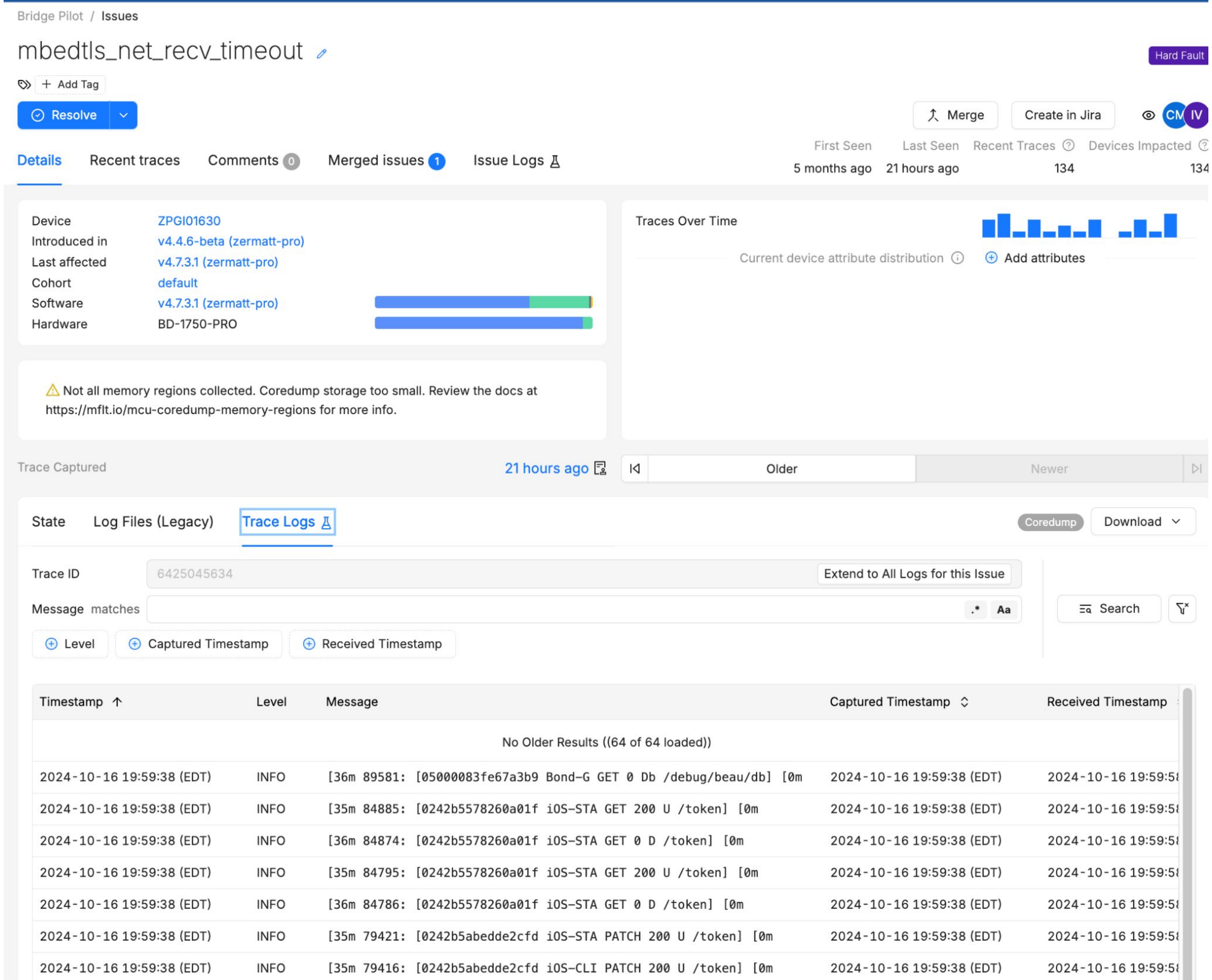
□ e={level="0",addr="0x400907ff",func="xQueueGenericSend",args=[{name="xQueu...
y-core
-CORE-SENTRY-CORE-7JR 1hr ago | 15mo old root

□ UnicodeDecodeError bond-core-sentry-core
'utf-8' codec can't decode byte 0xfe in position 110: invalid start byte
BOND-CORE-SENTRY-CORE-5C2 Unhandled 2hr ago | 23mo old

□ GDB exited (None / frame={level="0",addr="0x4017f3a9",func="_mdns_question_matches",args=[{name="...
bond-core-sentry-core
BOND-CORE-SENTRY-CORE-7P3 6hr ago | 14mo old root

hen GDB exits with a fright,
Check your optimizations, get it right.
Clear that mutex, don't let it stall,
With the right flags, you'll conquer all! 🎯✨

Enter Memfault



mbedtls_net_recv_timeout

First Seen	Last Seen	Recent Traces	Devices Impacted
5 months ago	21 hours ago	134	134

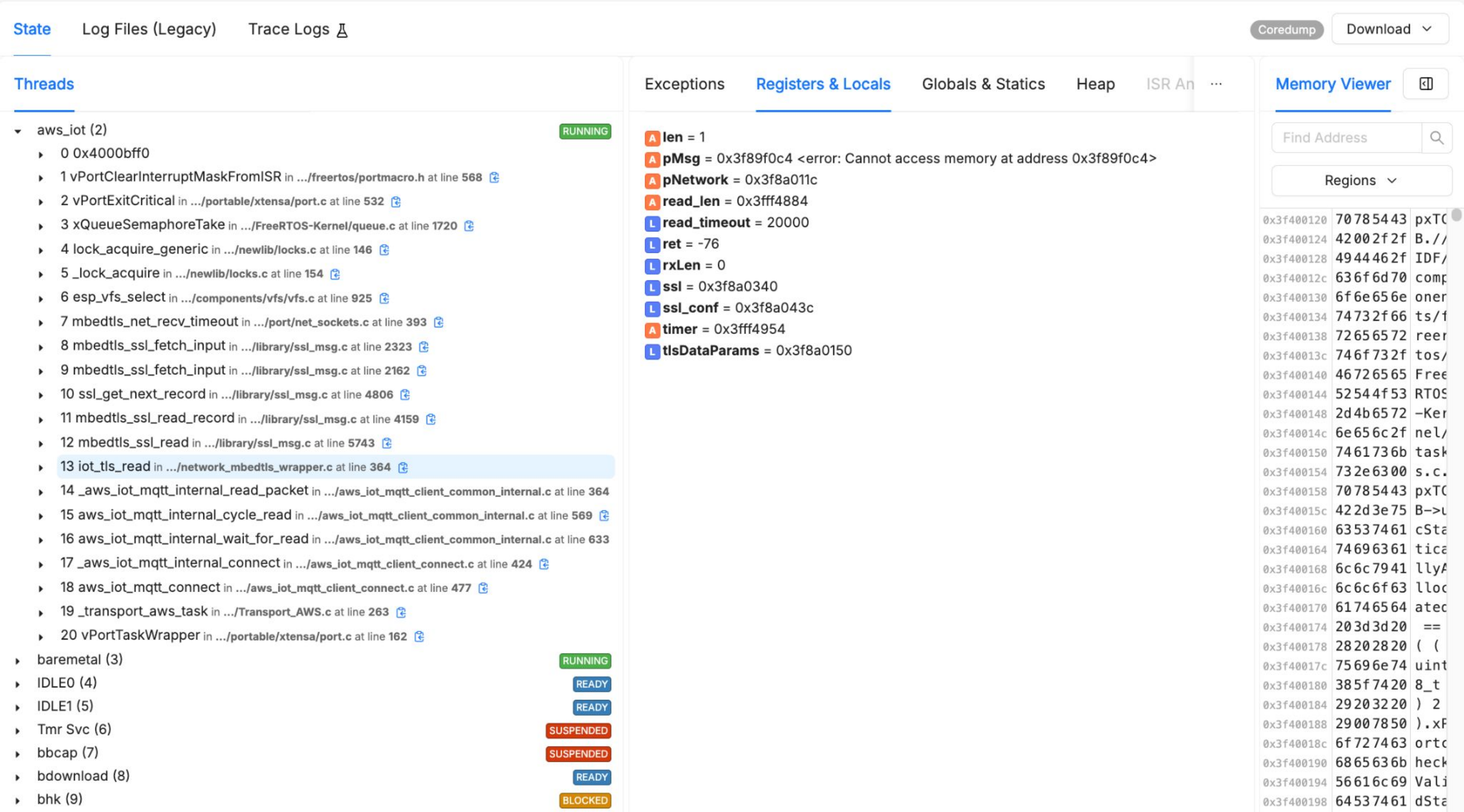
Device	ZPGI01630
Introduced in	v4.4.6-beta (zermatt-pro)
Last affected	v4.7.3.1 (zermatt-pro)
Cohort	default
Software	v4.7.3.1 (zermatt-pro)
Hardware	BD-1750-PRO

⚠ Not all memory regions collected. Coredump storage too small. Review the docs at https://mflt.io/mcu-coredump-memory-regions for more info.

Trace Captured 21 hours ago

Trace ID: 6425045634

Timestamp ↑	Level	Message	Captured Timestamp	Received Timestamp
No Older Results ((64 of 64 loaded))				
2024-10-16 19:59:38 (EDT)	INFO	[36m 89581: [05000083fe67a3b9 Bond-G GET 0 Db /debug/beau/db] [0m	2024-10-16 19:59:38 (EDT)	2024-10-16 19:59:5
2024-10-16 19:59:38 (EDT)	INFO	[35m 84885: [0242b5578260a01f iOS-STA GET 200 U /token] [0m	2024-10-16 19:59:38 (EDT)	2024-10-16 19:59:5
2024-10-16 19:59:38 (EDT)	INFO	[36m 84874: [0242b5578260a01f iOS-STA GET 0 D /token] [0m	2024-10-16 19:59:38 (EDT)	2024-10-16 19:59:5
2024-10-16 19:59:38 (EDT)	INFO	[35m 84795: [0242b5578260a01f iOS-STA GET 200 U /token] [0m	2024-10-16 19:59:38 (EDT)	2024-10-16 19:59:5
2024-10-16 19:59:38 (EDT)	INFO	[36m 84786: [0242b5578260a01f iOS-STA GET 0 D /token] [0m	2024-10-16 19:59:38 (EDT)	2024-10-16 19:59:5
2024-10-16 19:59:38 (EDT)	INFO	[35m 79421: [0242b5abedde2cfd iOS-STA PATCH 200 U /token] [0m	2024-10-16 19:59:38 (EDT)	2024-10-16 19:59:5
2024-10-16 19:59:38 (EDT)	INFO	[35m 79416: [0242b5abedde2cfd iOS-CLI PATCH 200 U /token] [0m	2024-10-16 19:59:38 (EDT)	2024-10-16 19:59:5



```
aws_iot (2)                                                                    RUNNING
    0 0x4000bff0
    1 vPortClearInterruptMaskFromISR in .../freertos/portmacro.h at line 568
    2 vPortExitCritical in .../portable/xtensa/port.c at line 532
    3 xQueueSemaphoreTake in .../FreeRTOS-Kernel/queue.c at line 1720
    4 lock_acquire_generic in .../newlib/locks.c at line 146
    5 _lock_acquire in .../newlib/locks.c at line 154
    6 esp_vfs_select in .../components/vfs/vfs.c at line 925
    7 mbedtls_net_recv_timeout in .../port/net_sockets.c at line 393
    8 mbedtls_ssl_fetch_input in .../library/ssl_msg.c at line 2323
    9 mbedtls_ssl_fetch_input in .../library/ssl_msg.c at line 2162
    10 ssl_get_next_record in .../library/ssl_msg.c at line 4806
    11 mbedtls_ssl_read_record in .../library/ssl_msg.c at line 4159
    12 mbedtls_ssl_read in .../library/ssl_msg.c at line 5743
    13 iot_tls_read in .../network_mbedtls_wrapper.c at line 364
    14 _aws_iot_mqtt_internal_read_packet in .../aws_iot_mqtt_client_common_internal.c at line 364
    15 aws_iot_mqtt_internal_cycle_read in .../aws_iot_mqtt_client_common_internal.c at line 569
    16 aws_iot_mqtt_internal_wait_for_read in .../aws_iot_mqtt_client_common_internal.c at line 633
    17 _aws_iot_mqtt_internal_connect in .../aws_iot_mqtt_client_connect.c at line 424
    18 aws_iot_mqtt_connect in .../aws_iot_mqtt_client_connect.c at line 477
    19 _transport_aws_task in .../Transport_AWS.c at line 263
    20 vPortTaskWrapper in .../portable/xtensa/port.c at line 162
baremetal (3)                                                                  RUNNING
IDLE0 (4)                                                                      READY
IDLE1 (5)                                                                      READY
Tmr Svc (6)                                                                    SUSPENDED
bbcap (7)                                                                      SUSPENDED
bdownload (8)                                                                  READY
bhk (9)                                                                        BLOCKED
```

```
A len = 1
A pMsg = 0x3f89f0c4 <error: Cannot access memory at address 0x3f89f0c4>
A pNetwork = 0x3f8a011c
A read_len = 0x3fff4884
L read_timeout = 20000
L ret = -76
L rxLen = 0
L ssl = 0x3f8a0340
L ssl_conf = 0x3f8a043c
A timer = 0x3fff4954
L tlsDataParams = 0x3f8a0150
```

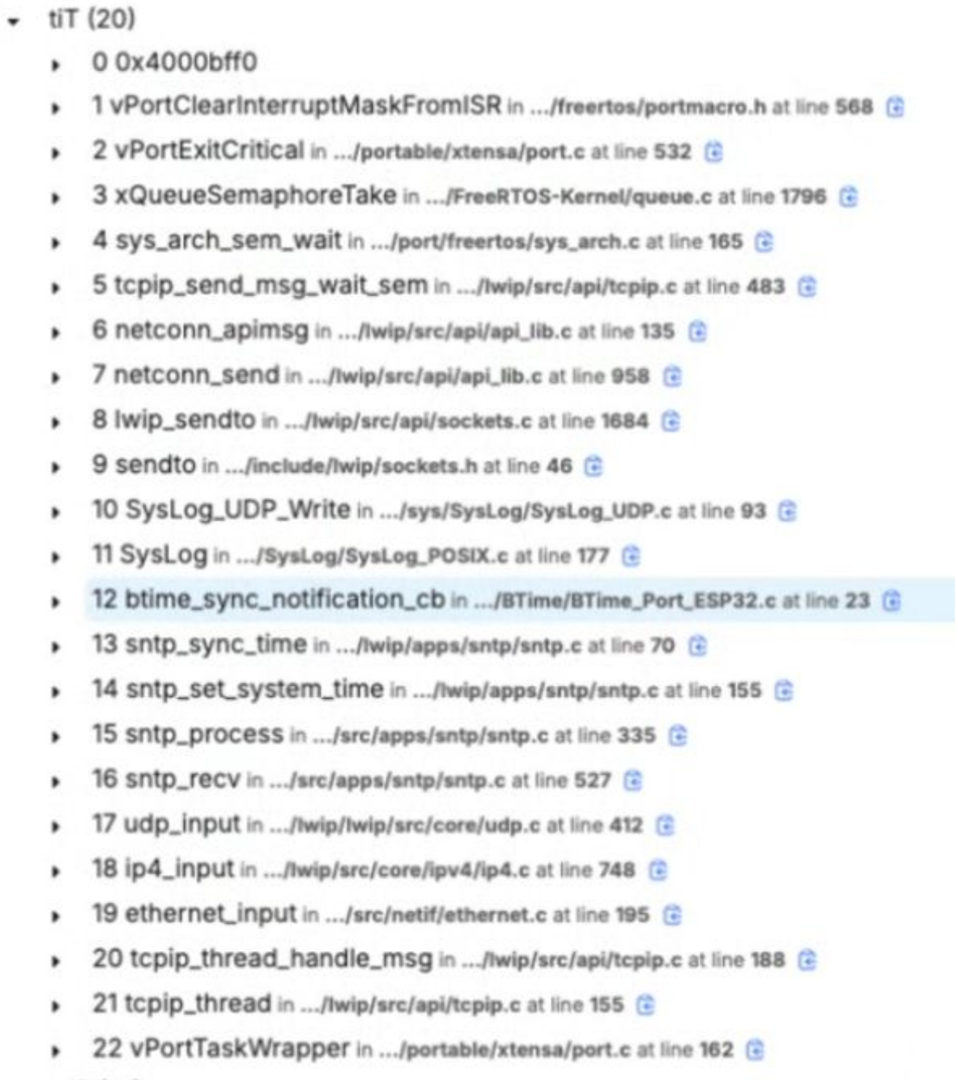
```
0x3f400120 70785443 pxTC
0x3f400124 42002f2f B.//
0x3f400128 4944462f IDF/
0x3f40012c 636f6d70 comp
0x3f400130 6f6e656e oner
0x3f400134 74732f66 ts/f
0x3f400138 72656572 reer
0x3f40013c 746f732f tos/
0x3f400140 46726565 Free
0x3f400144 52544f53 RTOS
0x3f400148 2d4b6572 -Ker
0x3f40014c 6e656c2f nel/
0x3f400150 7461736b task
0x3f400154 732e6300 s.c.
0x3f400158 70785443 pxTC
0x3f40015c 422d3e75 B->u
0x3f400160 63537461 cSta
0x3f400164 74696361 tica
0x3f400168 6c6c7941 llyA
0x3f40016c 6c6c6f63 lloc
0x3f400170 61746564 atec
0x3f400174 203d3d20  ==
0x3f400178 28202820 ( (
0x3f40017c 75696e74 uint
0x3f400180 385f7420 8_t
0x3f400184 29203220 ) 2
0x3f400188 29007850 ).xP
0x3f40018c 6f727463 ortc
0x3f400190 6865636b heck
0x3f400194 56616c69 Vali
0x3f400198 64537461 dSta
```

Deadlock Example

When we have a suspected deadlock on a developer's desk, we send `\n~\n` on the serial port to trigger a crash.

We can then inspect each of the tasks and the cause immediately becomes apparent.

Here we are trying to use BSD sockets API from within the tcpip task. No no!



```
tiT (20)
  0 0x4000bff0
  1 vPortClearInterruptMaskFromISR in .../freertos/portmacro.h at line 568
  2 vPortExitCritical in .../portable/xtensa/port.c at line 532
  3 xQueueSemaphoreTake in .../FreeRTOS-Kernel/queue.c at line 1796
  4 sys_arch_sem_wait in .../port/freertos/sys_arch.c at line 165
  5 tcpip_send_msg_wait_sem in .../lwip/src/api/tcpip.c at line 483
  6 netconn_apimsg in .../lwip/src/api/api_lib.c at line 135
  7 netconn_send in .../lwip/src/api/api_lib.c at line 958
  8 lwip_sendto in .../lwip/src/api/sockets.c at line 1684
  9 sendto in .../include/lwip/sockets.h at line 46
  10 SysLog_UDP_Write in .../sys/SysLog/SysLog_UDP.c at line 93
  11 SysLog in .../SysLog/SysLog_POSIX.c at line 177
  12 btime_sync_notification_cb in .../BTime/BTime_Port_ESP32.c at line 23
  13 sntp_sync_time in .../lwip/apps/sntp/sntp.c at line 70
  14 sntp_set_system_time in .../lwip/apps/sntp/sntp.c at line 155
  15 sntp_process in .../src/apps/sntp/sntp.c at line 335
  16 sntp_recv in .../src/apps/sntp/sntp.c at line 527
  17 udp_input in .../lwip/lwip/src/core/udp.c at line 412
  18 ip4_input in .../lwip/src/core/ipv4/ip4.c at line 748
  19 ethernet_input in .../src/netif/ethernet.c at line 195
  20 tcpip_thread_handle_msg in .../lwip/src/api/tcpip.c at line 188
  21 tcpip_thread in .../lwip/src/api/tcpip.c at line 155
  22 vPortTaskWrapper in .../portable/xtensa/port.c at line 162
```

Memfault Impact on Issue Investigation

Removes activation energy that used to be required before we would open up coredumps. More insight as a standard procedure in issue investigation.

Also now used during development and internal testing.

Currently we only use Memfault on a small subset of our devices, so we are still using a homegrown solution for tracking errors across the whole fleet.

Zero maintenance effort.

Minimal work required to upgrade SDK to access new observability features.

Several serious FW issues have been solved faster than we could have before.

We would not want to give up this level of visibility, and now knowing what it takes to build & maintain it, I would rather not have to do that ourselves again.

Q&A

bond

Memfault