

How to retrieve Cortex-M4 logs after crash

Stable: 11.02.2019 - 13:21 / Revision: 22.01.2019 - 08:55

Contents

1 Introduction	1
2 Prerequisite	1
3 Getting an M4 coredump file	1
4 Reading the logs from the coredump file	2

1 Introduction

When the Arm[®] Cortex[®]-M4 firmware crashes, Linux[®] running on Arm[®] Cortex[®]-A7, can generate an M4 coredump file. This M4 coredump records the states of the working memory of the Cortex-M4 firmware which permits to assist in diagnosing errors.

If some Cortex-M4 firmware logs have been output to the [trace buffer declared in the resource table](#), they can be retrieved from the M4 coredump.

2 Prerequisite

An M4 coredump file is generated under the following conditions:

- the Cortex-M4 informs the Cortex-A7 when it crashes. This is the case for instance if the Cortex-M4 firmware implements the watchdog detection (see [WWDG internal peripheral](#)),
- the [remoteproc node of the Linux device tree](#) defines the *recovery* property,
- a Linux service allowing to capture coredump files is running. In the STM32MPU Embedded Software distribution this is implemented by the *m4-dump.rules* udev rule.

3 Getting an M4 coredump file

When the Cortex-M4 crashes, Linux running on the Cortex-A7 is informed and stores a timestamped M4 coredump file in `/var/crash`.

Example: listing M4 coredumps

```
Board $> ls -l /var/crash
total 1188
-rw-r--r-- 1 root root 608500 Nov 30 16:02 m4-fw-error_2018-11-30_16-02-21.dump
-rw-r--r-- 1 root root 606420 Nov 30 16:07 m4-fw-error_2018-11-30_16-07-26.dump
```

4 Reading the logs from the coredump file

Start with getting the offset address of the trace log:

```
PC $> readelf -l m4-fw-error_2018-11-30_16-02-21.dump
Elf file type is CORE (Core file)
Entry point 0x10003129
There are 6 program headers, starting at offset 52

Program Headers:
Type           Offset      VirtAddr    PhysAddr    FileSiz MemSiz  Flg Align
LOAD           0x0000f4   0x00000000  0x00000000  0x10000 0x10000 RWE  0
LOAD           0x0100f4   0x30000000  0x30000000  0x40000 0x40000 RWE  0
LOAD           0x0500f4   0x10000000  0x10000000  0x40000 0x40000 RWE  0
LOAD           0x0900f4   0x10040000  0x10040000  0x02000 0x02000 RWE  0
LOAD           0x0920f4   0x10042000  0x10042000  0x02000 0x02000 RWE  0
LOAD           0x0940f4   0x100200e4  0x100200e4  0x00800 0x00800 RWE  0
```

The trace log (assuming that the resource table defines a trace buffer) is stored in the last memory segment. In this example, its offset is 0x0940f4.

You can read the M4 binary coredump file from this offset to get the logs or you can use the following command to output them (use **-c +N** option where N is *log offset + 1*):

```
PC $> tail -c +$(0x940f4 + 1) m4-fw-error_2018-11-30_16-02-21.dump
[00000.000][INFO ]Starting WWDG
[00001.422][INFO ]WWDG now!
```