HardFault Debugging

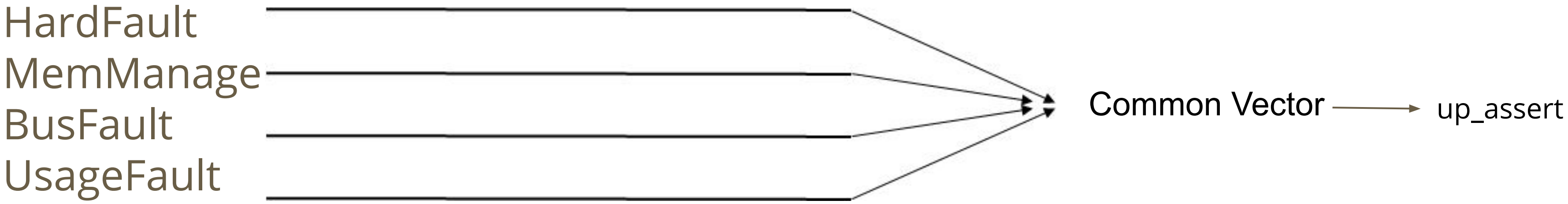
David Sidrane

http://www.nscdg.com

Zurich 06/20/2019

What is a HardFault?

Within NuttX, all roads lead to up_assert via the common vector



Common causes:

- Both software and hardware can cause HardFaults
- Hardware accessing a peripheral that is not enabled -BusFault
- Executing a pure virtual function (AKA: null pointer execution)
- Dereferencing a null pointer
- Stack crash (AKA: stack smashing) or wild pointer corrupting data used downstream

Scale of difficulty debugging a HardFault

Simple to debug:

(Repeatable occurence of HardFault)

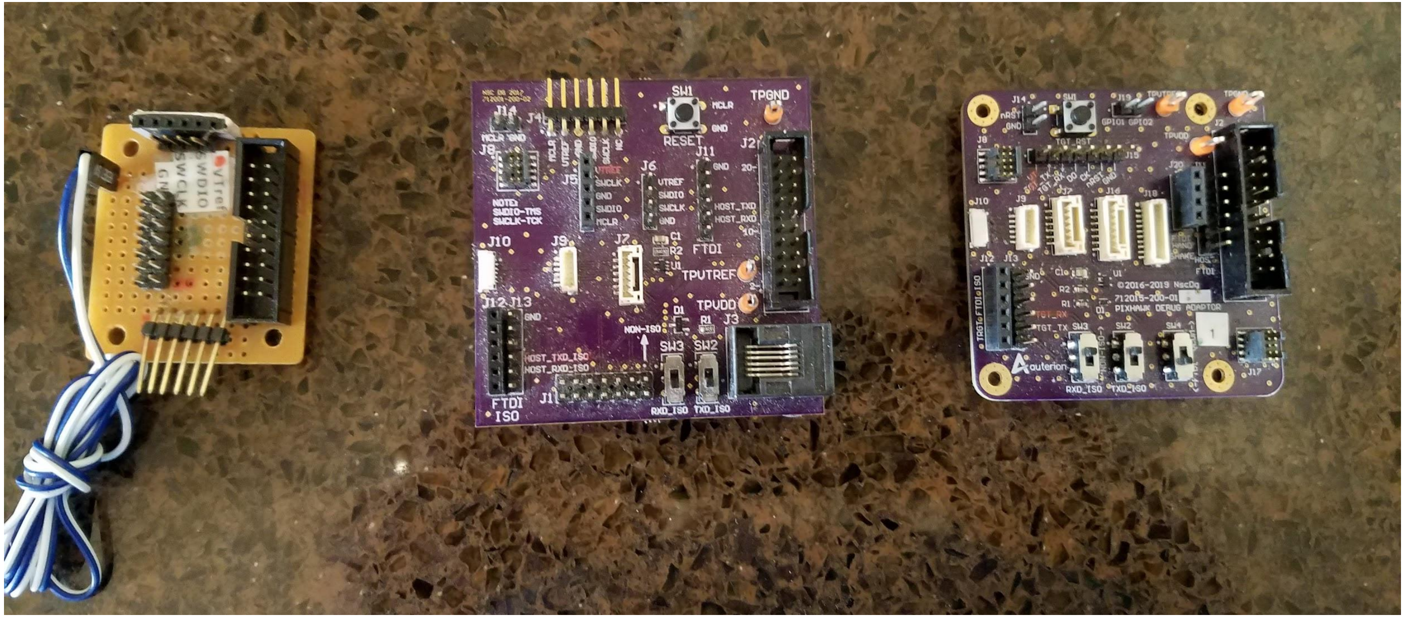
- Hardware accessing a peripheral that is not enabled
- Executing a pure virtual function
- Dereferencing a null pointer

Complex to debug:

(random occurence of HardFault)

- Stack crash or wild pointer corrupting data used downstream
- Inappropriate hardware interrupt priority settings

The Evolution leading to the Pixhawk debug adapter



Tools - HardFault debugging is not as difficult as it used to be

The old days:

Bond-out InCurcuitEmulator (ICE)

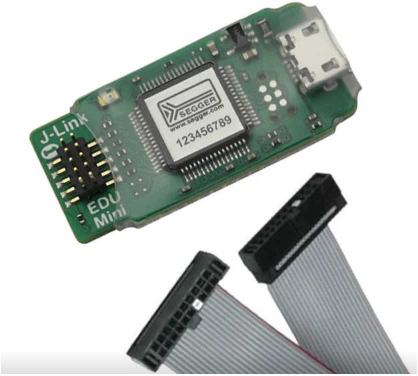
$15,000   USD

=

Current:

JTAG debugger

$20.00 USD



Live and Postmortem Debugging

Live:

GNU ARM → GNU MCU Eclipse!

Set a breakpoint on up_hardfault and up_assert
Set the PC equal to the LR
Select assembly single step
And step to bx lr instruction in do_irq that will return you to the line of code that caused the HardFault

Postmortem:

Reviewing the HardFault log
Choosing addresses in flash
And disassembling at those addresses