Porting openMMC to STM32 Microcontrollers for Flexible AMC Development

Michael Stubbings, Emilio Perez Juarez, Laurence Stant, Diamond Light Source, Didcot, UK
Adam Wujek, CERN, Geneva, Switzerland

1. Background

MicroTCA:

- Advanced Mezzanine Cards (AMCs) provide interchangeable user functionality in a MicroTCA system[1].
- A Module Management Controller (MMC) is required to negotiate power and communications between the AMC and MicroTCA shelf.
- Diamond Light Source is interested in designing custom AMCs for signal conditioning and interlock support.
- However, recreating MMC behaviour is a challenge due to its complex interactions and hardware dependant code.

openMMC:

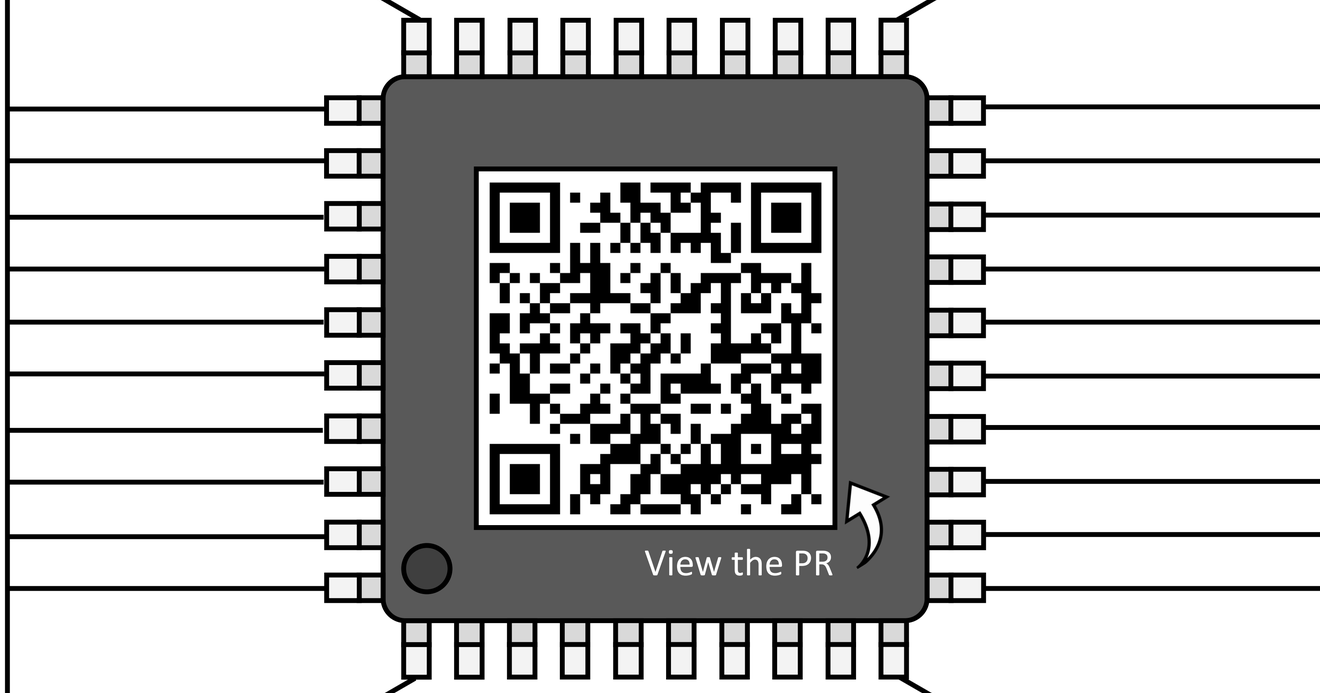
- openMMC is an open source firmware for replicating MMC behaviour in a MicroTCA system[2].
- It uses a modular architecture for flexible configuration of the sensors, communications and controller used in a design.
- However, Its microcontroller support is limited to the NXP LPC17xx family of chips.

Context:

- CERN created a fork of openMMC that ports it onto STM32 controllers, but damaged the portable architecture in the process.
- Diamond started this project to reform the CERN fork into a new STM32 support branch that is compatible with openMMC.

2. Objectives

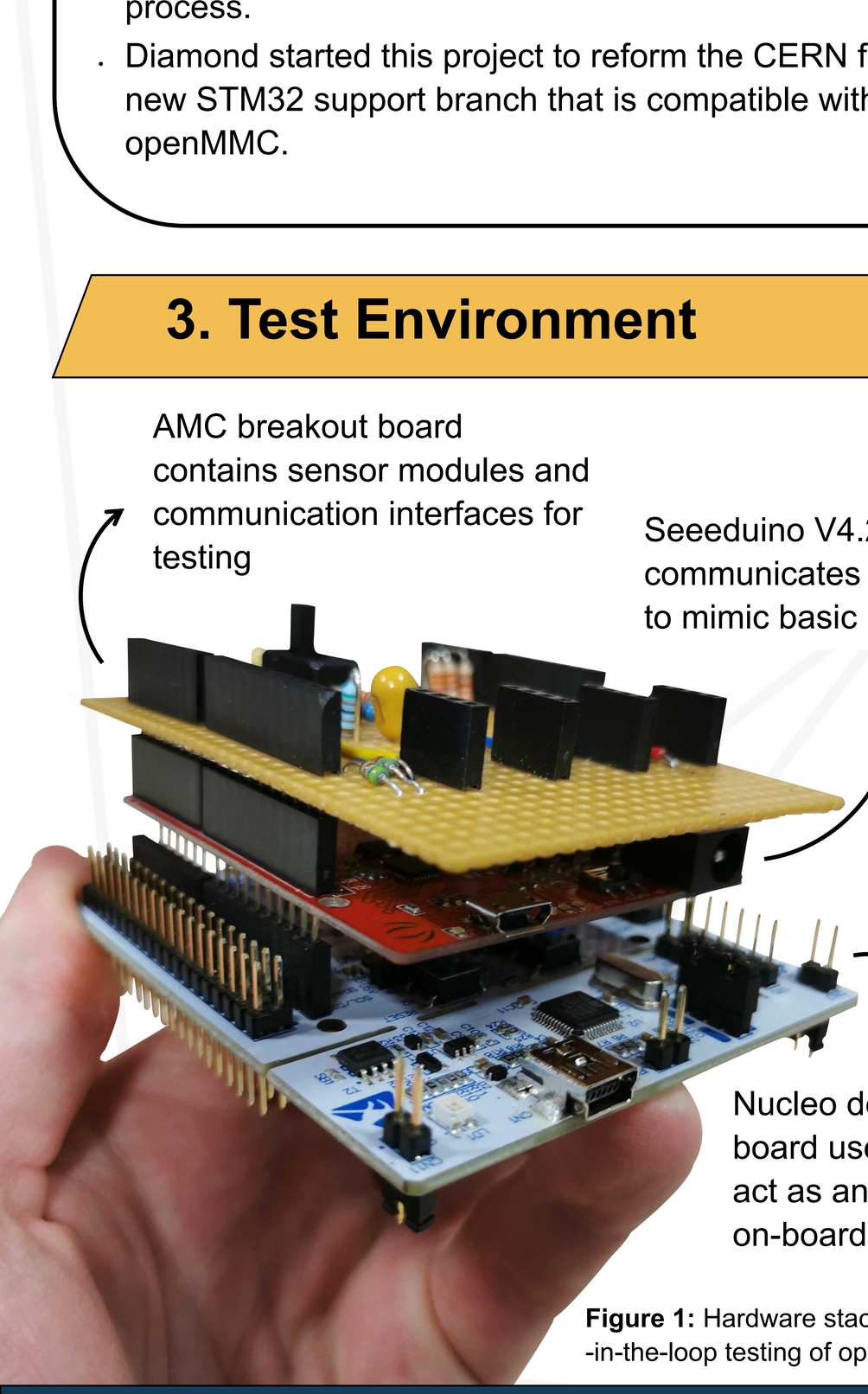
- To provide a stable platform for the testing and further development of openMMC firmware at Diamond Light Source.
- To complete the integration of a CERN fork, bringing STM32 support to openMMC.



3. Test Environment

AMC breakout board contains sensor modules and communication interfaces for testing

Seeeduino V4.2 communicates with a host PC to mimic basic MCH behaviour



Nucleo development board uses openMMC to act as an MMC using its on-board STM32 chip

Figure 1: Hardware stack used for hardware -in-the-loop testing of openMMC

4. Code Integration

A complete analysis and reform of the CERN fork was performed to create the STM32 support branch:

- 154 commits were identified from the common ancestor between the fork and upstream/main.
- Commits were reviewed then modified, dropped, broken down or kept based on their contents.
- I2C initialisation was rewritten for greater flexibility over which port could be used for IPMB communications.
- Code implementing CERNs enhanced Rear Transition Module (eRTM) was dropped from all commits.
- Incompatible licensing on source code was regenerated.
- Code history was rewritten for the new support purpose.
- Coding best practices were applied during modifications.

5. Conclusions

- openMMC is an effective non-commercial solution for designing in-house AMCs and replicating MMC behaviour.
- Introducing STM32 support creates new opportunities for the firmware to be used by a wider audience
- The methodologies and experiences outlined in this project provide a useful asset for those seeking to add new microcontroller support in future.

References:
[1] VadaTech MicroTCA Overview, https://www.vadatech.com/media/pdf_MicroTCA_Overview.pdf
[2] openMMC, https://github.com/lnls-dig/openMM

For more information please visit www.diamond.ac.uk
or contact Michael.stubbings@diamond.ac.uk