Update on MRF Timing System

Timing Workshop ICALEPCS 2019

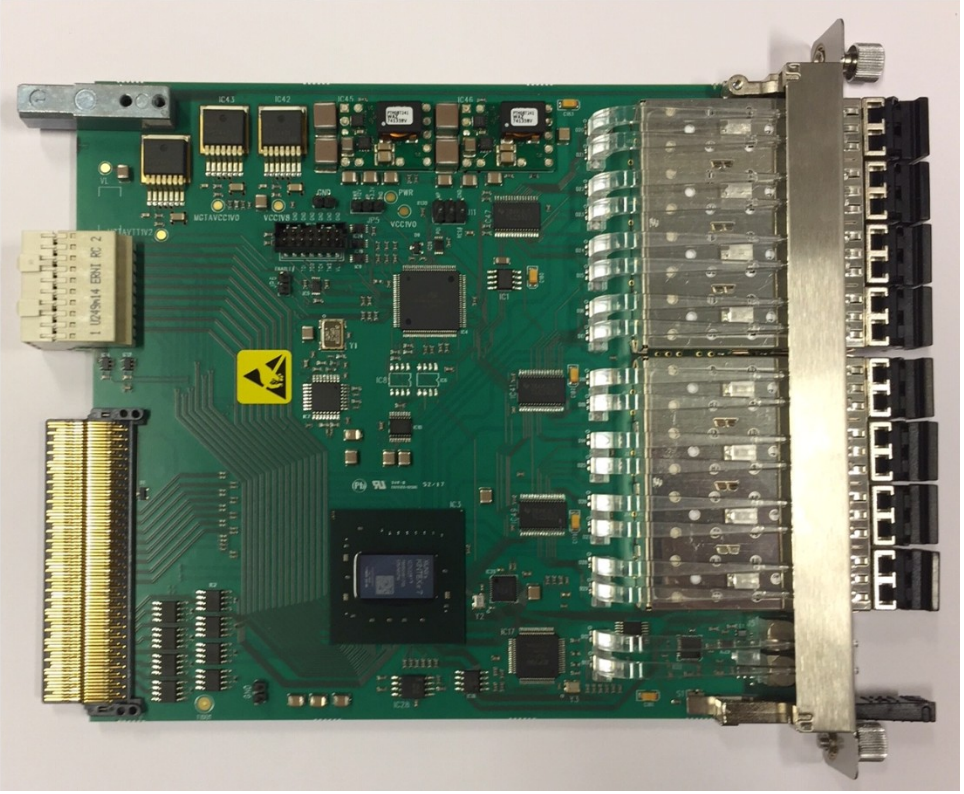
Jukka Pietarinen

Micro-Research Finland Oy

MRF Linux Driver and API on github

- Linux driver https://github.com/jpietari/mrf-linux-driver
  - Support for all MRF PCI and PCIe based hardware
  - Firmware upgrade support
- MRF API on https://github.com/jpietari/mrf-linux-api
  - Functions for programming EVG/EVR in C
  - Shell wrapper functions to control devices from command line
  - Future plan: omit MRF Linux driver and move over to using userspace I/O

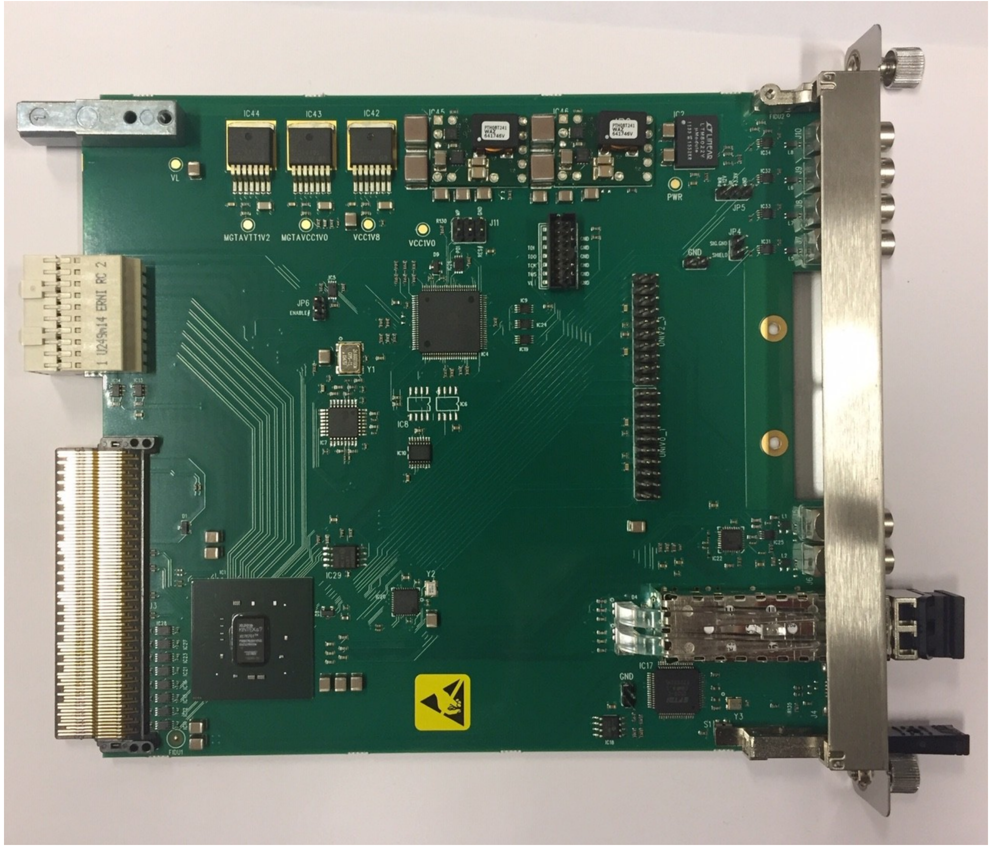
mTCA-EVM-300

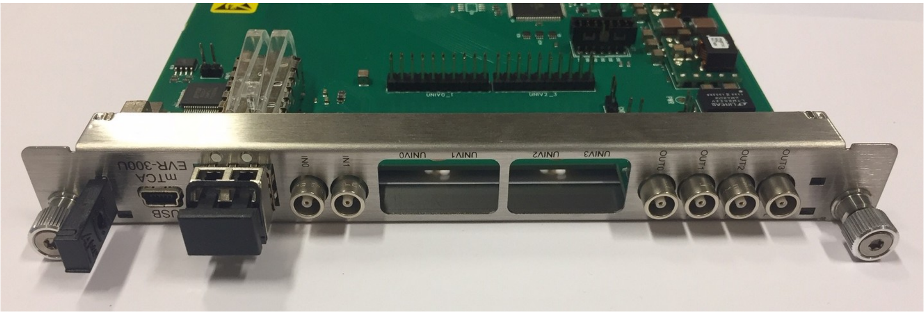


- Event Generator (EVG)
  - 2 sequencers with maskable events
    - Max. 2047 events/sequence
    - 32 bit timestamp
  - 8 multiplexed counters
  - data buffer up to 2k bytes
  - segmented data buffer
    - 127 segments, 16 bytes each
- 7-Way Fan-Out/Concentrator
  - +4 backplane ports
- Two Event Receivers (EVR)
  - 8 internal pulse outputs
  - one sequencer
- Event rate conversion (with some data buffer related limitations)
- Front panel input phase monitoring/select features
- Distributed bus phase selection
- RF input monitoring

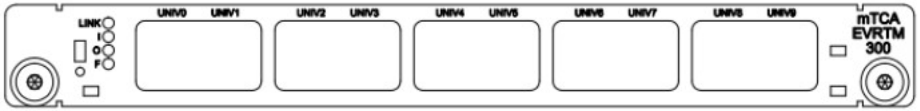
mTCA-EVR-300U

- Event Receiver (EVR)
- Two Universal I/O slots
  - both compatible with -DLY modules
- Four LVTTL outputs
- Two TTL inputs for external triggers
- Backplane triggers
- Can drive TCLKA/TCLKB clocks with GTX logic
- RTM interface, no RTM designed yet





mTCA-EVRTM-300 in development



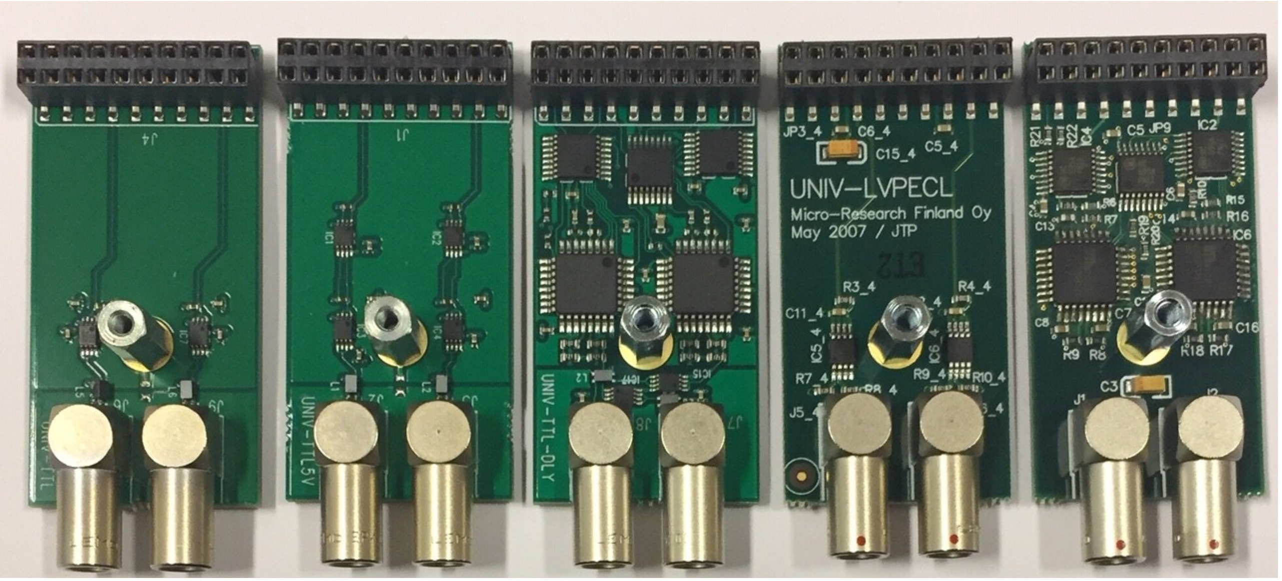
- Rear Transition Module (RTM) for mTCA-EVR and mTCA-EVM
- Five Universal I/O Modules
- Support for -DLY modules in all slots

- Current status (as of Oct. 2019):
  - Changes to MMC code done
  - Firmware support incomplete

Universal I/O Modules

- UNIV-TTL
  - LVTTL output
- UNIV-TTL5V
  - 5V TTL output
- UNIV-TTL-DLY
  - LVTTL output
  - Delay tuning 1024 steps of ~9 ps
- UNIV-LVPECL
  - differential LVPECL output
- UNIV-LVPECL-DLY
  - differential LVPECL output
  - Delay tuning 1024 steps of ~9 ps

UNIV-TTL UNIV-TTL5V UNIV-TTL-DLY UNIV-LVPECL UNIV-LVPECL-DLY



Universal I/O Modules

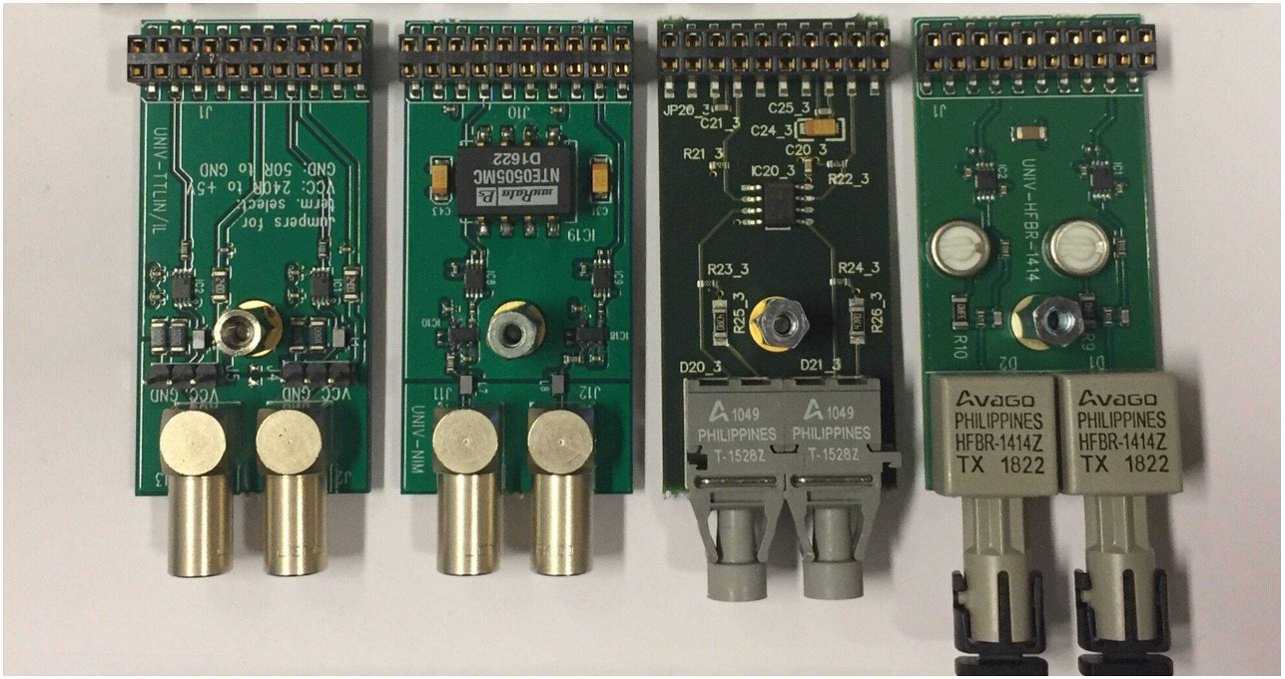
- UNIV-TTLIN
  - TTL input
  - configurable term.
    - 240 ohm to +5V
    - 50 ohm to GND

- UNIV-NIM
  - NIM output

- UNIV-HFBR-1528
  - Optical Output

- UNIV-HFBR-1414
  - Optical Output

UNIV-TTLIN/IL UNIV-NIM UNIV-HFBR-1528 UNIV-HFBR-1414



Open Source Event Receiver - Introduction

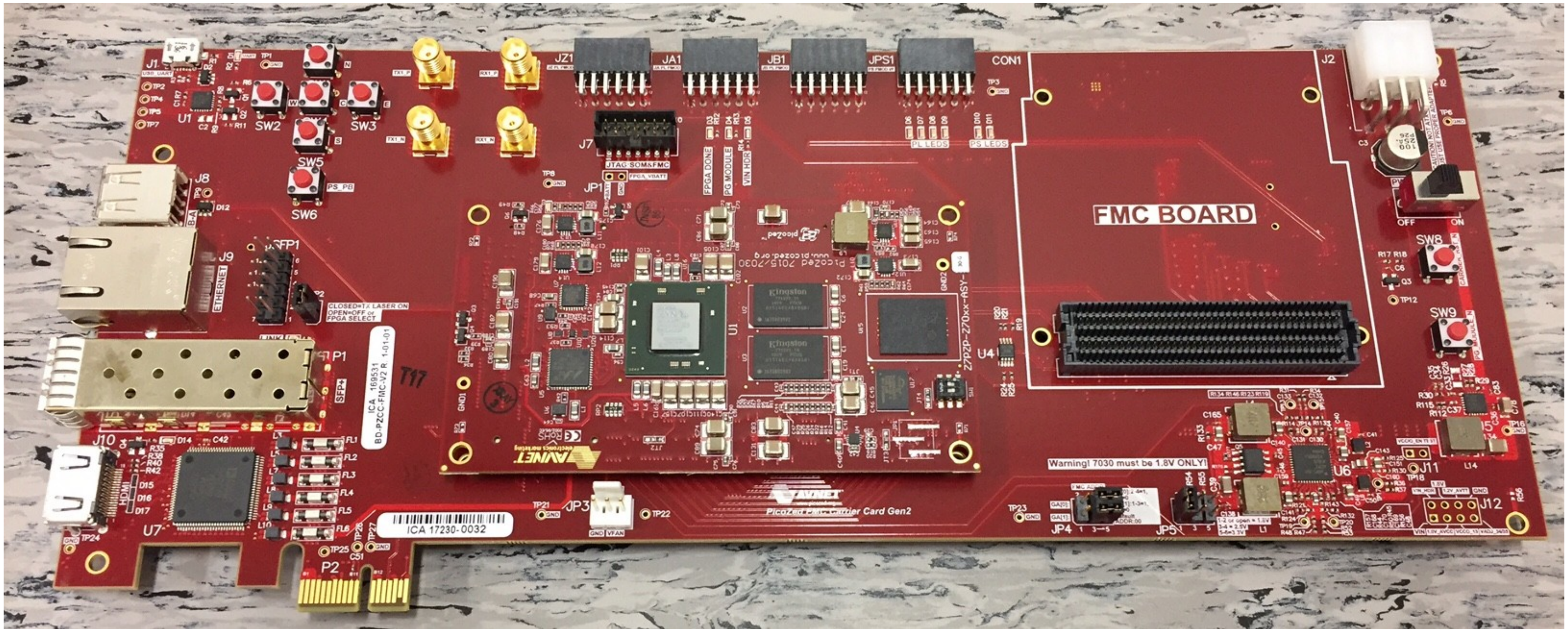
- What is the Open Source Event Receiver
  - Basic building block required to build devices receiving the MRF Timing protocol including but not limited to Delay Compensation capability
- What is it not
  - it is not a replacement firmware for current MRF products
  - it is not a complete Event Receiver with MRF product compatible register map
    - No bus/register interface
    - No pulse generators
    - etc.
- Available on GitHub https://github.com/jpietari/mrf-openevr

Open Source Event Receiver - Requirements

- Hardware
  - Xilinx Kintex-7 based FPGA with GTX transceivers **ONLY!**
  - Zynq 7Z030 is Kintex-7 based
  - SFP Transceiver
  - Reference clock for GTX
  - Example design built for
    - Avnet PicoZed 7Z030
    - Avnet PicoZed FMC Carrier Card V2
- Software
  - Xilinx Vivado 2017.4 (Free WebPack version is sufficient)
- Xilinx programming cable
  - e.g. Platform Cable USB II

Avnet PicoZed FMC carrier with 7Z030 SOM

- Avnet PicoZed AES-Z7PZ-7Z030-SOM-G
- Avnet PicoZed FMC carrier AES-PZCC-FMC-V2-G
  - Zynq 7Z030 incorporates
    - Kintex-based FPGA core
    - Four GTX transceivers
    - Dual-core ARM Cortex-A9
- This kit has everything from the hardware point of view to be used as an event receiver



Kintex-7 Requirement

- GTX transceiver
  - so far acceptable transceivers have been: Virtex-II-Pro, Virtex-5 GTX, 7-series GTX, Lattice ECP2M/ECP3 (limited capability)
- 7-series DCM
  - capable of continuous phase shifting (e.g. Virtex-II-Pro and Virtex-5 do not support this, Lattice ECP2M/ECP3 have 22.5 degree steps)
- It is possible to decode the event stream with almost any FPGA with a gigabit transceiver, if sub event clock period timing accuracy is not required