

# Java Driver Implementation for the Ethernet Flash ADC

John McKisson, Thomas Jefferson National Accelerator Facility

- Multi-platform library for Ethernet Flash ADC (EFADC) and Ethernet Trigger Supervisor (ETS)
- Easy to use API
  - Simple DAQ system running in ~15 lines of code
  - Usable by 3<sup>rd</sup> party DAQ software (ie Kmax) for larger systems
  - Coincident trigger setup
- OO abstraction of the ETS/EFADC connectivity tree
- Event aggregation

# Poster

Jefferson Lab  
Thomas Jefferson National Accelerator Facility

## Java Driver Implementation for the Ethernet Flash ADC

John McKisson<sup>1</sup>

<sup>1</sup>Thomas Jefferson National Accelerator Facility, Newport News VA

**Abstract:**  
 An EFADC is a data acquisition module designed to provide multichannel flash ADC data collection with a synchronous timing method that allows coincidence determination across as many as seven separate units to within 4 ns. Each EFADC unit provides 16 individual flash ADC converter channels with storage memory and multiple EFADCs can be managed by an Ethernet Trigger Supervisor (ETS), to provide a single network connection in a tree hierarchy. The architecture of the EFADC triggering is separated into 4-channel modules, with each module being capable of generating triggers. Each trigger section can be configured with varying length integration windows, individual thresholds, and can provide coincident trigger conditions for use in applications requiring coincidence such as PET.

**User Interfaces for Applications:**  
 We use Kmax (<http://www.sparrowcorp.com>) as a graphical environment to interact with the EFADC. Toolsheets are written in Java which allows the use of any Java library, allowing us to use our EFADC driver as a back-end. We implement a ClientHandler in the Kmax toolsheet as the interface to an EFADC/ETS unit and are then able to run image and spectra generation algorithms with the provided data.

EFADC module

ETS with 2 EFADCs

2x2 PET configuration using 4 Romanuscu HB500 PMTs

3x3 coincident data points

**The EFADC Driver:**  
 The interface to the EFADC/ETS system for both data extraction and control is implemented over a physical layer of 1000baseT Ethernet, using UDP protocol. The driver software is capable of handling a connection to a single EFADC unit or multiple through an ETS. The driver software is implemented in Java making use of the Netty (<http://netty.io>) library package which provides a high-level event driven network stack.

**Object-Oriented Design:**  
 The software provides an intuitive object-oriented interface modeled after the hardware's inherent tree connectivity, and is able to determine and provide the proper driver configuration depending on the hardware's physical configuration.

**Master Mode**  
 Serial links #1-8  
 EFADC16 #1  
 ETS master mode  
 User Console (PC)  
 EFADC16 #16

Single Tree Configuration

**Master/Slave Mode**  
 Serial links #1-8  
 EFADC16 #1  
 ETS slave mode #1  
 Serial link #9  
 ETS master mode  
 User Console (PC)  
 EFADC16 #748  
 ETS slave mode #8  
 EFADC16 #256

Extended Tree Configuration

ETS Maximum Connectivity