

Channel Access in Erlang

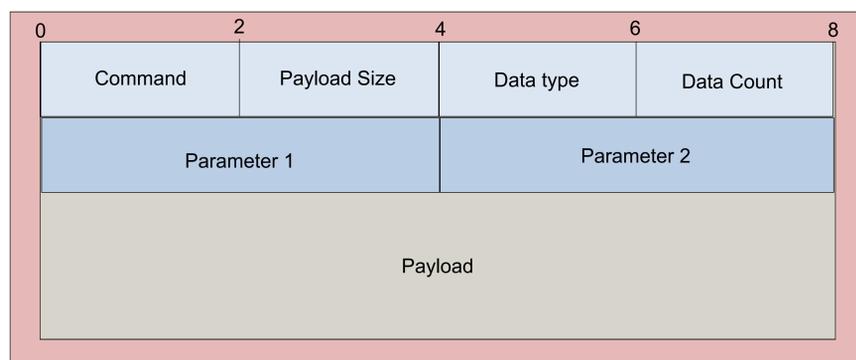
Dennis J. Nicklaus
Fermilab

Highlights

- Pure Erlang Implementation
- Provides a bridge between Acnet Devices and EPICS PVs
- Channel Access client functionality
- Acnet device database stores PV name for corresponding Acnet device
- Searches for EPICS PVs (IOCs)
- Monitors (readings) or settings
- No C code or libraries to link with
- Erlang byte-code is very portable with no recompilation required
- Functional language opens up new applications
- No erlang server (IOC) code yet

Channel Access Details

All Channel Access communications are with standard messages as shown below (there is also an extended message we don't support yet)



Sample Code

Here is some sample lower level code which implements the PV search multicast

```
%%% Create a standard CA Message with the provided parameters
encode_any(Cmd,Psize,Dtype,Dcount,P1,P2,Payload) ->
  <<Cmd:16,Psize:16,Dtype:16,Dcount:16,P1:32,P2:32, Payload/binary>>.

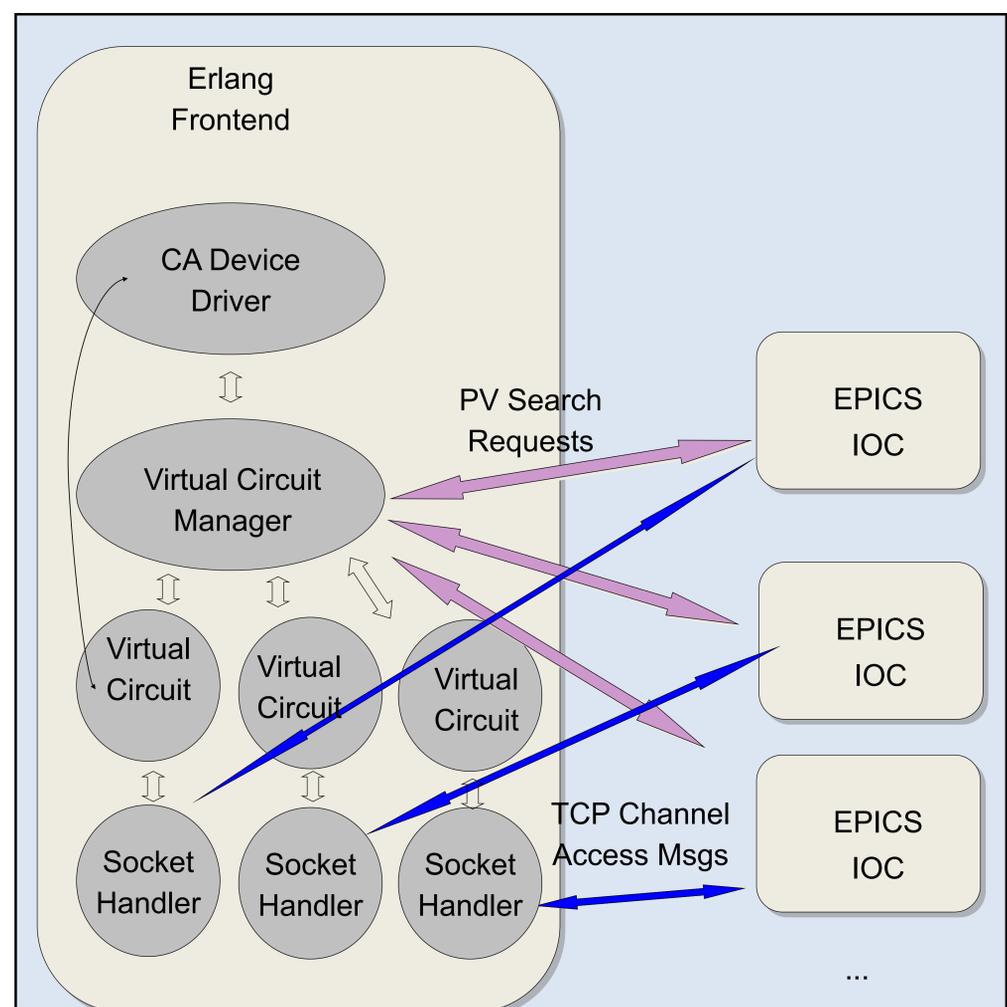
%%% encode a PV name as a payload (pad out to an even multiple of 8 bytes)
name_to_payload(Name) ->
  Padbytes = (8 - (length(Name) rem 8)) * 8,
  Bname = list_to_binary(Name),
  <<Bname/binary, 0:Padbytes>>.

%%% Create the CA message used in UDP PV search commands
encode_search(Name,CID) ->
  Payload = name_to_payload(Name),
  encode_any(?CMD_SEARCH,size(Payload),0,?PROTO_VERSION,CID,CID,Payload ).

%%% Send out the search packet to a list of addresses (i.e. EPICS_CA_ADDR_LIST)
send_out_ca_search(Sock,UDP,Name,NextCID) ->
  SearchPacket = ca:encode_search(Name,NextCID),
  lists:map(fun(OneUDP) ->
    ok = gen_udp:send(Sock, OneUDP, ?CA_PORT,SearchPacket)
  end, UDP).
```

Architecture

- ◆ Channel Access device driver plugs into Erlang front-end framework
- ◆ This device driver maintains list of Acnet device to EPICS PV translation (these are downloaded at start-up time)
- ◆ Virtual Circuit Manager manages PV searches, caches IOCs found, maps between PV read requests and virtual circuits (IOCs)
- ◆ Virtual Circuit manages monitors for one IOC, tracks state information of messages required when opening a new channel and manages the TCP socket
- ◆ TCP Socket Handler performs lowest level TCP communication, passes messages received up to the Virtual Circuit



Abstract

We have developed an Erlang language implementation of the Channel Access protocol. Included are low-level functions for encoding and decoding Channel Access protocol network packets as well as higher level functions for monitoring or setting EPICS process variables. This provides access to EPICS process variables for the Fermilab Acnet control system via our Erlang-based front-end architecture without having to interface to C/C++ programs and libraries. Erlang is a functional programming language originally developed for real-time telecommunications applications. Its network programming features and list management functions make it particularly well-suited for the task of managing multiple Channel Access circuits and PV monitors.

Acknowledgements

Thank you to Cosylab for the Channel Access protocol documentation and to the tech-talk community for their continued help.