

Functional specifications

This section provides explanation on the libLDA architecture. The library provide:

- Loading configuration and sending it to hardware.
- Slow control, trigger, and acquisition threads manage by the `running` global variable.
- A read-out call ensuring the completeness of data regard to triggers that provide this data to a callback user function.

```
int main (int argc, char * argv[])
{
    /* Configure */
    LLR_CALICE_CONF::Conf* conf = LLR_CALICE_CONF::parseCommandLine(argc, argv);
    Driver run(conf);
    run.configure();

    /* Start */
    run.startSlowControl();
    run.start();
    run.startSoftTrigger();

    /* Read out */
    while(running) {
        usleep(run.delay_usr);
        run.dump(readCallback);
    }
    while (run.dump(readCallback) >0);
}

return 0;
}
```