```
function [sim, finished, outlist, userdata]=experiment(sim, ev, inlist, userdata)
    // Do the experiment; collect e.g. data to a shared memory
    AccGyro = inlist(1);
    [sim] = ld_printf(sim, 0, AccGyro, "Collecting data ... ", 6);
    // ...
    outlist=list(out);
endfunction

function [sim, outlist, userdata]=whileComputing(sim, ev, inlist, userdata)
    // While the computation is running this is called regularly
    [sim, out] = ld_const(sim, ev, 0);
    outlist=list(out);
endfunction

function [sim, outlist, userdata]=whileIdle(sim, ev, inlist, userdata)
    // When no calibration or computation is active
    AccGyro = inlist(1);
    [sim, out] = ld_const(sim, ev, 0);
    outlist=list(out);
endfunction

function [sim, CalibrationOk, userdata]=evaluation(sim, userdata)
    // Will run in a thread in background execution mode. Only one time step is executed here.
    // ...
    // Embedded e.g. a Scilab script that will be called once to perform the calibration
    [sim, Calibration] = ld_scilab2(sim, 0, in=CombinedData, comp_fn=scilab_comp_fn, include_scilab_fns=list(),
                                    scilab_path="BUILDIN_PATH");
    // ...
    // Tell ld_AutoExperiment that the calibration was successful
    [sim, oneint32] = ld_constvecInt32(sim, 0, vec=1)
    CalibrationOk = oneint32;
endfunction

[sim, finished, outlist] = ld_AutoExperiment(sim, ev, inlist=list(AccGyro, Ts), insizes=[6,1], outsizes=[1], ...
                                intypes=[ORTD.DATATYPE_FLOAT,ORTD.DATATYPE_FLOAT], ...
                                outtypes=[ORTD.DATATYPE_FLOAT], ...
                                ThreadPrioStruct, experiment, whileComputing, evaluation, whileIdle);
```