**CK\_Light software architecture**

authors: Yang, Ta, Wang

**Note**: Complicated concepts need examples, maybe in Chinese.

**main quality of the software**:  
Speed, robust, easy to use.

Secure is not the most important thing.

**Basic idea**: the system contains a microskernel, which deal with hardware communications and provide kernel services, which also include saving/loading the data/measurement. The intermediate level (adaptor) corresponds to translate the measurement modes into kernel services.

This microkernel need to deal with hardware level error and exceptions.

**Details**:

1. microkernel:  
   microskernel includes 3 parts: hardware io service, file io service and adaptor. When new hardware is implemented in the microkernel, the layers above work directly, no need to change anything.
   1. hardware driver/interface: hardware driver or interface tends to break. Some hardware is relatively slow in reaction and most of the hardware can deal with only one request at a time (status need to be tracked.). Therefore, multithreading is crucial and proper error handling is also very important. The software should not lock itself, but return to keep on track on the errors.
   2. hardware io service: a standard hardware io service is needed. The idea is to transform all hardware action into a standard service so that new hardware can be added at this level without touching anything at the higher level.   
      Definition of the hardware io service: input/output, analog or digital, analog range or possible digital values, typical response time, if can be triggered, master or slave and ect. need to check my matlab code for more information. Basically, it is somehow overlay with the hardware vector structure.   
      synchronization of the hardware is a bit tricky: Multiple channels can be synchronized and
   3. file io service: Duplicate the parameters and save the data to a temporary file. Then save to one file after measurements and consistency checking of the parameters, default file name, username+ timestamp.ext. Md5 check is needed for data integrity.
   4. adaptor: Translate the measurement parameters into kernel services, which is in the end called by the measurement events. Hardware is associated with the properties of parameters. parameters are element of the parameter list. Each element shares standard properties, which is extendable (property numbers). The properties should be extendable as well. (Check my Matlab parameter list, + hardware typical response time, last response time (time or -1 means failed). + axis can across multiple scanning axis. one function to check the integrity of the scanning. Display of the data should be flexible as well according to the data display settings. Complete scans can be also axis, but is a node, and cannot extended to a normal scan but only a loop like scanning. scanning level, low or high.
2. Measurement mode:

File is there in the github. It is mainly useful for the user interface level.

1. Data output and input,
   1. data output has two modes:
      1. hardware synchronization mode: different channels got there scanning wave beforehand. the hardware action need either a trigger to start or start by software. different hardware channels are synchronized by the hardware connections. Typical example is the NI cards.
      2. software synchronized mode: software sends the stream constantly. while hardware got their paces due to their own resources. therefore, a cache is need for them.
   2. data input or acquisition:
      1. DMA (direct memory access) option should be available. some hardware supports it.
2. data structure, check the one from Picoquant and Abberior,

hardware vector, parameter vector, data.   
**hardware vector** is responsible for hardware initialization (at the beginning and running initialization), status checking (oneline, initialized, last operation time, …), unique for each hardware and exclusive to each other. hardware initialization, some of them are slow, therefore, multi-threaded (to speed up) and error (wait) return, in software initialization.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| properties | value | misc | properties | value | misc |
| 'scale' | 1 | ->hardware | **range** | [-inf inf] | ->hardware |
| channels |  | list | **channelmode** |  | i/o/both |
| hwName | ‘’ | unique |  |  |  |
| portType |  |  | **port** |  |  |
| origin |  |  | **reverse** | false |  |
|  |  |  | **flybackMode** |  | mixed ? |
| t\_last | 0 | ms | **t\_resonse** | 0 | ms |
| callback |  |  |  |  |  |
| hwSpecific |  | cluster |  |  |  |

**parameter vector**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| properties | value | misc | properties | value | misc |
| outputMode | ?? |  |  |  |  |
| 'scanrange' | [0 0] |  | **'range'** | [-inf inf] | ->hardware |
|  |  |  | **'stepsize'** |  |  |
| 'scale' |  | ->hardware | **'flybackMode'** |  | ?->hardware |
| 'scanmode' |  |  | **'outputchan'** |  | ?->hardware |
| 'shift' | 0 |  | **'axis'** | -1 |  |
| 'syncout' | false |  | **'syncchan'** | [] |  |
| 'syncmode' | 0 |  |  |  |  |
| origin |  |  | **target** |  |  |
| repeat | false |  |  |  |  |
| reverse | false |  | **wait** | 0 | ms |
| display | false |  | **axisDisp** |  |  |
| transient | false |  |  |  |  |
| callback |  |  |  |  |  |
|  |  |  |  |  |  |

**data**, compressed

hareware setting and some parameters can be locked while measurement.

End of the file for now. 20191022

~~low level scanning axis, high level scanning axis: low level need additional hardware to sync better. high level scanning axis will be synced in different thread in software.~~

~~A hardware list and then be linked by the parameter list. incase multiple scanscheme need to access the same hardware and get messed up. Hardware status, can be set by different scanscheme and set to different state?~~

~~synchronization: two ways, the best would be a cache is used for the stream to the hardware. Then all the activated hardware need to take the data from the cache. the other process is used to fill the cache to avoid big data junk has to be generated.~~

~~exception handling: Exception can be reported to the company. By a standard email, then a server is needed and keep on track of all the reported exceptions.~~

~~potential functionalities: calibration, camera (should be same as the hardware)~~

~~special for the software~~

~~hardware: laser, shutter (eom aotf), power tuning, scanner, slm, focus lock, spectrometer, TCSPC, scanner, detector gating, fast gating,~~

~~testing code is also required!~~