Diagram Documentation

Primary Use Cases - (Use Case diagram)

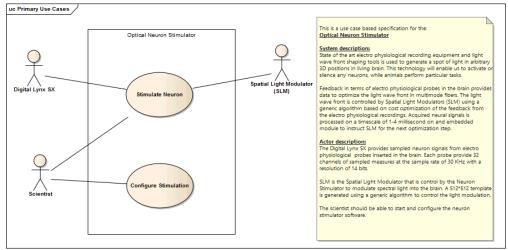


Figure: 1

<u>Logical View</u> - (Class diagram)

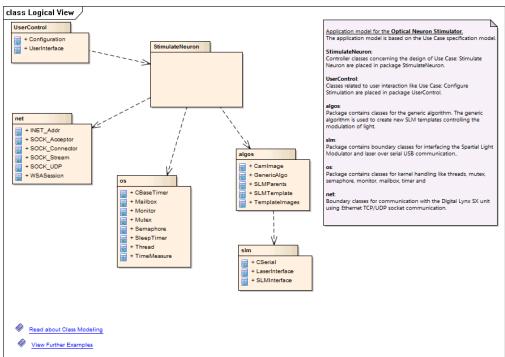


Figure: 2

StimulateNeuron - (Class diagram)

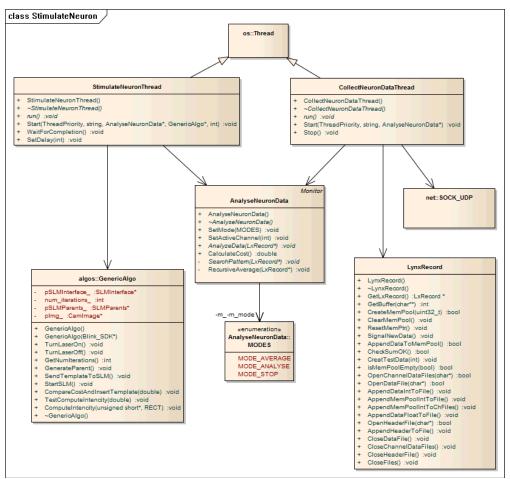


Figure: 3

 $\textbf{Generate Templates and Control SLM} \textbf{-} (Interaction \ diagram)$

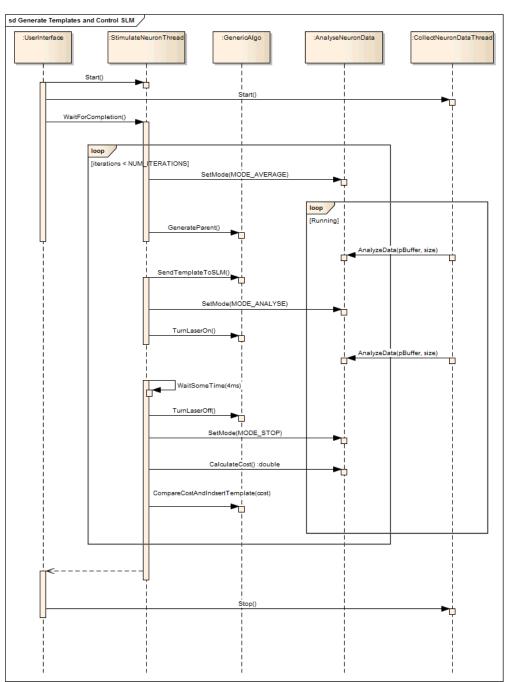


Figure: 4

$\underline{Collect\ and\ Analyze\ Neuron\ Data} \ \hbox{--} \ (Interaction\ diagram)$

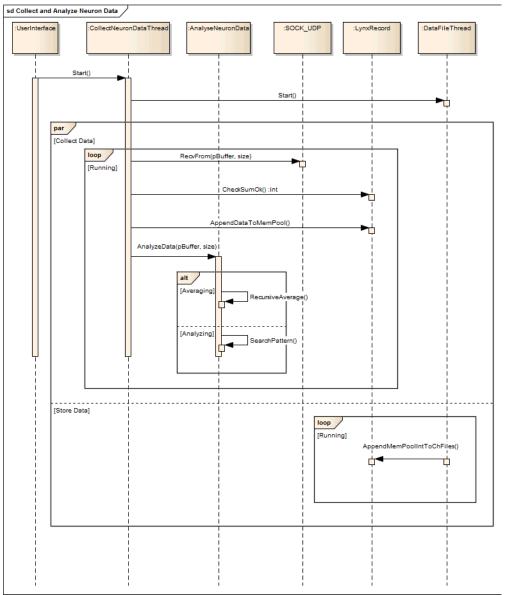


Figure: 5

Neuron Data File Thread - (Class diagram)

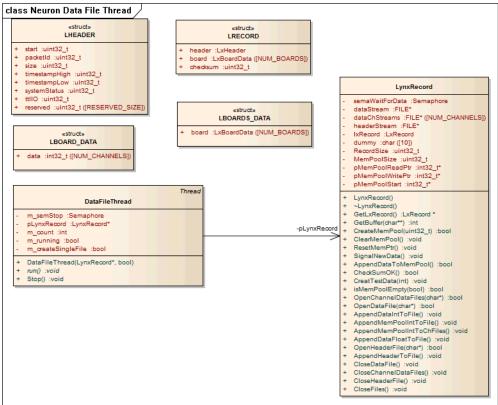


Figure: 6

UserControl - (Class diagram)

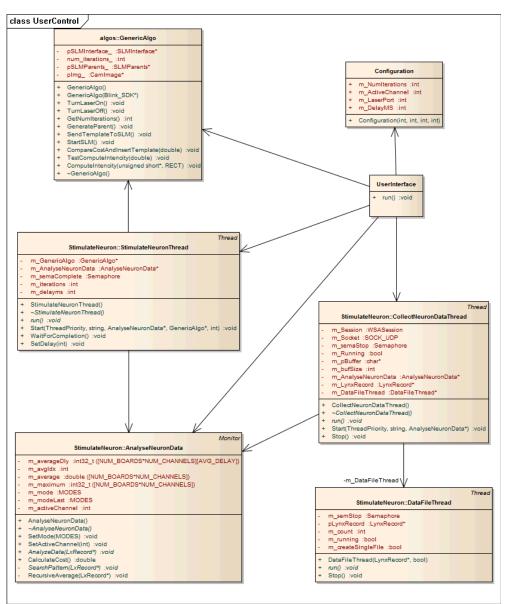


Figure: 7

Generic Algorithm Classes - (Class diagram)

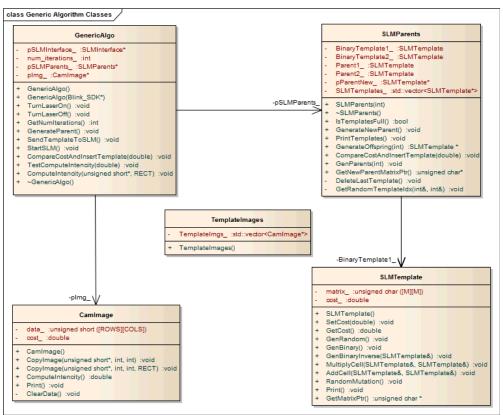


Figure: 8

SLM and Laser Interface Classes - (Class diagram)

class SLM and Laser Interface Classes SLMInterface oldPhase :uchar_vec pSDK_ :Blink_SDK* board_number :int {readOnly} n_boards_found_ :unsigned int constructed_okay_ :bool + SLMInterface(Blink_SDK*) + SLMInterface() ~SLMInterface() + ResetInterface() :bool + SendTestPhase(unsigned char*, int) :bool + SendPhase(unsigned char*) :bool phaseRandom(size_t, size_t, uchar_vec&) :void Consume_keystrokes():void Precalculate_and_loop(uchar_vec&, uchar_vec&, int, Blink_SDK&) :bool LaserInterface m_CSerial :CSerial* m_baudRate :int LaserInterface(int) ~LaserInterface() OpenPort(int) :bool TurnOn() :bool TurnOff() :bool **CSerial** # m_hIDComDev :HANDLE # m_OverlappedRead :OVERLAPPED # m_OverlappedWrite :OVERLAPPED # m_bOpened :bool + CSerial() ~CSerial() Open(int, int) :BOOL Close() :BOOL ReadData(void*, int) :int SendData(char*, int) :int ReadDataWaiting():int IsOpened() :BOOL WriteCommByte(unsigned char) :BOOL

Figure: 9

Kernel Classes - (Class diagram)

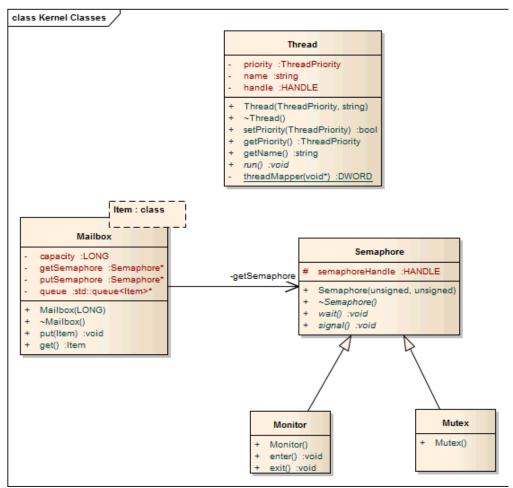


Figure: 10

<u>Timer Classes</u> - (Class diagram)

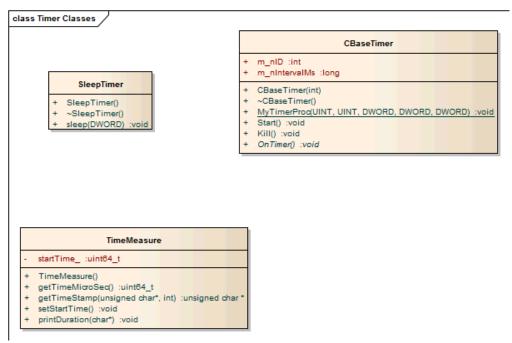


Figure: 11

Network Classes - (Class diagram)

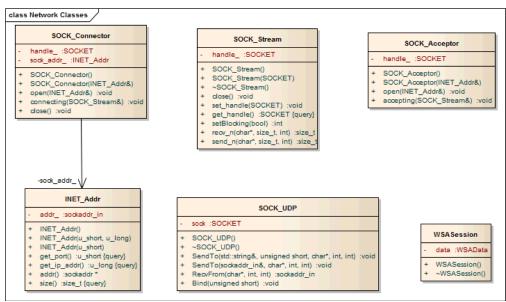


Figure: 12