* **Buffer Management Functions**

1. LoadMarkerLibrary
2. InitializeMarkerLibrary
3. ReleaseLibrary
4. InitializeBufferManager
5. GetLECDeviceIDList
6. GetLECDeviceProperties not completed yet
7. DisconnectLECDevice
8. LoadLensData
9. LoadLaserData
10. GetDigitalIO
11. GetAnalogIO
12. ReadRegister
13. WriteRegister
14. GetRegister

* **List Management Functions**

1. GetListSize
2. ExecuteList
3. LoopList
4. AbortList

* **List Commands**

1. Application\_Event 8 don’t know what the parameter means.
2. Begin\_Job\_Event 4
3. Enable\_MOTF 4
4. Enable\_Wobble 4
5. End\_Job\_Event 4
6. Job\_Marker 4 not sure that is the progrss
7. Jump\_Abs 16
8. Laser\_On 8
9. Laser\_Signal\_Off 4
10. Laser\_Signal\_On 4
11. Long\_Delay 8
12. Mark\_Abs 16
13. MOTF\_Reset\_Jump 12
14. Set\_Active\_Correction\_Table 4
15. Set\_Correction\_Table 8
16. Set\_Enable\_Laser 4
17. Set\_Galvo\_DAC\_Config 4
18. Set\_Jump\_Speed 8
19. Set\_Laser\_Enable\_Delay 4
20. Set\_Laser\_Enable\_Timeout 4
21. Set\_Laser\_FPK 8
22. Set\_Laser\_Mod\_Delay 4
23. Set\_Laser\_Mode\_Config 4
24. Set\_Laser\_Off\_Delay 4
25. Set\_Laser\_On\_Delay 4
26. Set\_Laser\_Pipeline\_Delay 4
27. Set\_Laser\_Power 4
28. Set\_Laser\_Power\_Change\_Delay 8
29. Set\_Laser\_Pulse 12
30. Set\_Laser\_Standby 12
31. Set\_Laser\_Timing 4
32. Set\_Mark\_Speed 8
33. Set\_Microvector\_Delay 4
34. Set\_MOTF\_Direction 4
35. Set\_MOTF\_Cal\_Factor 8
36. Set\_MOTF\_Direction 4
37. Set\_MOTF\_Mode 4
38. Set\_PM\_FieldOffset 8
39. Set\_PM\_FieldOrientation 4
40. Set\_PM\_LaserPowerPct 8
41. Set\_PM\_MarkSpeedPct 8
42. Set\_PM\_PulsePeriodPct 8
43. Set\_PM\_PulseWidthPct 8
44. Set\_Servo\_Config 4
45. Set\_Wobble 516 not sure what is the parameter
46. Wait\_For\_IO 12
47. Wait\_For\_MOTF\_Count 8
48. Write\_Analog 8
49. Write\_Digital 8
50. Z\_Abs 8

// -- Buffer Management Functions:

// 1. DisconnectLECDevice

int LANHAL::Abort(bool ABlocking)

{

logging::Log(LanmarkControls::lf\_FPGAData, L"LANHAL::Abort", L"PC");

// Package the priority data

unsigned char\* data;

int size = streamdatapackager::Abort\_Data(&data, LanmarkControls::ar\_AbortJob);

// Create a Transmit Package header to put on the front of the Priority data

unsigned char\* newbytearray = NULL;

int newlistsize = 0;

STREAMPACKAGEHEADER header;

header.ipaddress = 0;

header.reqipaddress = 0;

header.dataid = LanmarkControls::di\_PriorityData;

header.datasize = size;

streamdatapackager::PackageStreamData(header, data, size, &newbytearray, &newlistsize);

//Abort looping

BaseHAL::Abort(ABlocking);

return SendPriorityData(newbytearray, newlistsize, LanmarkControls::pt\_Abort, LanmarkControls::ar\_AbortJob);

}

#define stringify( name ) # name

const char\* errorNames[] =

{

stringify(LanmarkControls::LECResponse::Success),