

FSSDK to GoPxL SDK / API Migration and Mapping

Common Function APIs Changes	1
Difference in Theory of Operations	2
Data Acquisition	2
Units of Operations	3
Detailed Configuration Mapping	4
Appendix / Miscellaneous	9

Common Function APIs Changes

FSSDK method/function	Purpose	GoPxL equivalent
Open(&camera_count, &camera_ids, OPEN_TIMEOUT)	Retrieve a list of connected cameras	GoDiscoveryClient.InstanceList
_Connect(camera_ids[0], EXAMPLE_IP_ADDR)	Connect to sensor and assign IP address	GoRestClient.Connect , GoSystem.Connect and GdpClient.Connect (depending on use case) To change sensor / scanner settings use GoRestClient.Connect. For data access GdpClient.Connect Note that IP address is typically already assigned. Can be programmatically changed via GoInstance.SetIpAddress
_SetIntParameter(char *cameraId, char *paramName, int value)	Update/set an integer parameter on camera. The most used parameters seem to be: - PARAM_TRIGGER_ZERO_SOURCE - PARAM_TRIGGER_SOURCE - PARAM_SENSOR_DATA_IN_FLASH - PARAM_PEAK_Y_UNIT	GoRestClient.Update General function to update sensor resources with content. Not type specific. Not that many specific parameters may not be supported by G5 anymore or replaced by different parameters.

	• PARAM_PEAK_X_UNIT • PARAM_PEAK_ENABLE • PARAM_REG_PULSE_FREQ	
<code>_GetIntParameter(char *camerald, char *paramName, int *value)</code>	Retrieve integer parameter from camera. The most used parameters are: Chi Ho: Need a mapping of the parameters to GoPXL SDK	GoRestClient.Read General function to read sensor resources. Can be cast to specific data types at read time.
<code>_SetStringParameter(char *camerald, char *paramName, char *value)</code>	Update/set a string parameter on camera. The most used parameters seem to be: • PARAM_SENSOR_CALIBRATION_FILE • PARAM_SENSOR_X_CALIBRATION_FILE	GoRestClient.Update General function to update sensor resources with content. Not type specific.
<code>_SetLineSortingOrder(char *camerald, int sortingOrder)</code>	Set the line sorting order for the data layers.	GoRestClient.Update ⚠ Not yet implemented!
<code>_SetLineCallback(char *camerald, int layerid, LineCallback callback)</code>	Assigns camera id and layer id to the line callback function.	GoRestClient.sub Subscribe to the resource to receive lines asynchronously. May require a formatting handler.
<code>_AdjustRoiAndFps(char* camerald, int operation, int inValue, int* outValue)</code>	Adjust and retrieve region of interest and FPS information. Note that function is used to set values as well as to retrieve values.	GoRestClient.Update and GoRestClient.Read
<code>LineCallback()</code>	Line callback function that captures z values, intensity, line length, x step size and meta information per line.	GoRestClient.sub Subscribe to the resource to receive lines asynchronously. May require a formatting handler.

Difference in Theory of Operations

Data Acquisition

GoSDK is based on a streaming design, where data is received synchronously using `GoGdpClient::ReceiveDataSync()` or asynchronously through `GoGdpClient::ReceiveDataAsync()` data handlers. Memory is allocated when the handlers are called and the user is responsible for freeing the data when they are done. Users could hold onto the data in other threads until the data is no longer needed.

Metadata of the received data is passed as part of each data receive stamp or as part of the headers. This allows the metadata, such as resolution, to be different between each data frame

Units of Operations

GoSDK always operates in real-world units (mm), except when receiving video images.

CPU Usage

G55xx compresses the data using the sensor hardware before streaming to the PC on a 1Gbps link, whereas the LCI sensor sent uncompressed data and required a 5Gbps link. Under typical conditions, the total PC CPU usage used to pass data into user applications is comparable between G5500 and LCI sensors.

Below is the CPU load and Ethernet bandwidth of two typical cases.

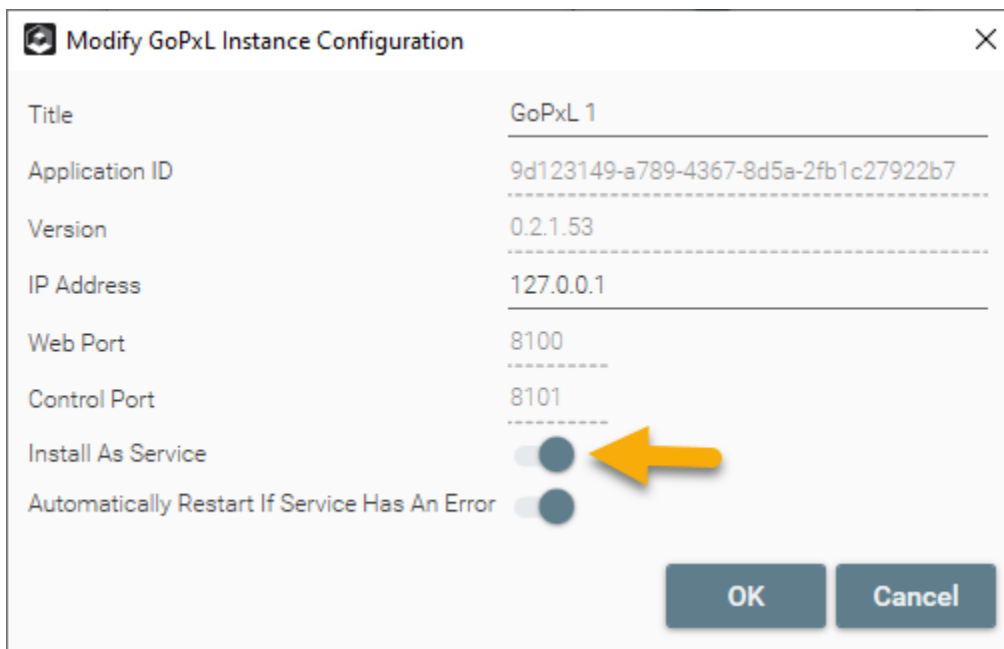
Scenario	G55xx		LCI-1220	
	Total CPU Usage	Ethernet Bandwidth	Total CPU Usage	Ethernet Bandwidth
8kHz, single layer	47%	28MB/s	58%	110 MB/s
3kHz, 3 layers	52%	36MB/s	60%	104MB/s

**CPU load is based on Intel(R) Core(TM) i7-6820HQ CPU @ 2.70GHz, 2701 Mhz, 4 Core(s), 8 Logical Processor(s)*

GoPxL Application

G55xx needs to be accelerated with the GoPxL application running on a PC (to uncompress the data as mentioned above), even if the user is only using GoPxL SDK in their application and not the GoPxL user interface.

GoPxL can be automatically launched as a service in the background with Windows startup. When adding a GoPxL instance, enable the “Install As Service” option as shown below. This will register the instance with Windows as a service and set it to automatically start.



Modify GoPxL Instance Configuration	
Title	GoPxL 1
Application ID	9d123149-a789-4367-8d5a-2fb1c27922b7
Version	0.2.1.53
IP Address	127.0.0.1
Web Port	8100
Control Port	8101
Install As Service	☒
Automatically Restart If Service Has An Error	☒
OK Cancel	

GoPxL Manager can also be used to make any necessary modifications to the service after it has been added.

The configuration for each instance is stored in id.xml under
ProgramData\LMI\GoPxLService\Instances\[app id]\.

When a new instance is created in GoPxL Manager, an instance specific id.xml file is created. id.xml is updated whenever an instance is modified in the GoPxL Manager or vice versa.

Detailed Configuration Mapping

Reference: FsSDK based on VevoParametersDefinitions.h

Reference: [G5 Scanner Confluence Page](#), [System Level Resource Page](#)

GoPxL resource paths:

- Sensor: /scan/engines/g5xxx/scanners/(scannerId)/sensors/(sensorId)
- G5 Sensor: /scan/engines/g5xxx/scanners/(scannerId)/sensors/sensorID/parameters
- G5 Layer N (Nth layer): <G5 Sensor Parameters>/layerSpecificParams/n

System Management

FsSDK Parameters	Purpose	GoPxL SDK Equivalent
String: PARAM_LOAD_RECEIPE PARAM_SAVE_RECEIPE	Load and save configuration	/jobs/commands/load, /jobs/commands/save
INT: PARAM_SENSOR_DATA_IN_FLASH	Check if sensor data is in flash	Not required for Gocator
STRING: PARAM_SENSOR_CALIBRATION_FILE PARAM_SENSOR_X_CALIBRATION_FILE PARAM_SENSOR_INTENSITY_CALIBRATION_FILE PARAM_SENSOR_Z_COMPENSATION_FILE	Path to Z calibration file	Not required for Gocator
Int: PARAM_SENSOR_TYPE	Return sensor type without LCI pre-fix	<G5 Sensor>/model
Int: PARAM_HEARTBEAT		Not needed. Heartbeat is done by GoPxL. Poll <G5 Sensor>/ConnectionStatus to detect changes in state
Int: PARAM_DEVICE_SERIAL_NUMBER	Sensor serial number	<G5 Sensor>/SerialNumber
Int: PARAM_USER_FLASH_WRITE_TIME		Not needed
Int: PARAM_FREQUENCY_CALCULATION	Enable or disable frequency calculation in the header of the profile. 0 = disabled, 1 = enabled (default) Changes timestamp from frequency	Not supported. Stamp in GoPxL data protocol is always returned in timestamp.

Scan Mode and Triggering

FsSDK Parameters	Purpose	GoPxL SDK Equivalent
Int: PARAM_PEAK_ENABLE	Output video, profile, line or batch	<G5 Sensor>/scanMode
Int: PARAM_INTENSITY_CORRECTION_ENABLE	Intensity correction	Not supported in 0.2, but will be supported in 1.0 release
Int: PARAM_SURFACE_MODE	Surface sorting mode Sensors LCI401, LCI1200, LCI1220 and LCI1620: 0 = disabled (default), 1 = bottom, 2 = top, 3 = brightest, 4 = two two top-most, 5 = two top-most.	Controlled thru surface generation and then layer control. <G5 Layer>/intensityFilterType/layerSortingOrder <G5 Scan Parameters>/Surface/Type
Int: PARAM_REG_PULSE_DIVIDER, PARAM_ENCODER_PULSE_WIDTH	Encoder divider and pulse width for external triggering	<G5 Sensor>/trigger/encoderSpacing
Int: PARAM_REG_PULSE_FREQ	Sensor trigger frequency	<G5 Sensor>/trigger/frameRate. Note: /trigger/source should be set properly
Int: PARAM_TRIGGER_<Control>	Control trigger source	Control via Gocator input settings Trigger debounce is not configurable
Int: PARAM_MOVING_DIRECTION	Moving Direction	Not supported in 0.2, but will be supported in 1.0 release

Data Acquisition

FsSDK Parameters	Purpose	GoPxL SDK Equivalent
Int: PARAM_MTU, PARAM_IFG	Network TCP MTU, interpacket delay	Not available. Internally handled
Int: PARAM_AVERAGE_X_PIXEL_SIZE, PARAM_AVERAGE_Z_PIXEL_SIZE	Average pixel resolution in ROI	Use XYZ resolution and offset from data stream
Int: PARAM_LINE_CALLBACK_MAX_LENGTH, PARAM_LINE_CALLBACK_XRES, PARAM_BATCH_LENGTH, PARAM_BATCH_TIMEOUT	Number of pixels expected from the line callback function. Can be called before grabbing but after initialization	Line callback is equivalent to uniform profile in GoPxL. Batch is equivalent to surface output in GoPxL

Int: PARAM_GRABBING	Start/stop sensor	Refer to GoPxL SDK documentation on how to start and stop sensors
---------------------	-------------------	---

Window Control & Layer Control Parameters

FsSDK Parameters	Purpose	GoPxL SDK Equivalent
Int: PARAM_IMAGE_HEIGHT, PARAM_IMAGE_OFFSETY	Imager active region in Pixel	In Gocator, all units are in mm, not in pixel <G5 Sensor Parameters>/activeArea
Int: PARAM_PEAK_Y_UNIT, PARAM_PEAK_X_UNIT	Window/Output Y in pixel units	Not supported in GoPxL. All GoPxL outputs are in real-world coordinates
Int: PARAM_IMAGE_HEIGHT_ZERO_POSITION	Returns image row z coordinate where height has been tuned nearest to '0' by calibration file on use.	Not supported in 0.2, but will be supported in 1.0. Refer to GoPxL SDK documentation on how to trigger alignment operation
Int: PARAM_IMAGE_<Parameters> PARAM_FLIP_<Parameters>	Image size, offset and orientation parameter	Receive info from Gocator data stamp and data structure in real time Image orientation is NOT supported

Exposure Control

FsSDK Parameters	Purpose	GoPxL SDK Equivalent
Int: PARAM_REG_PULSE_WIDTH_FLOAT	Pulse width (Exposure)	<G5 Sensor>/SingleExposure
Int: PARAM_HDR_ENABLED, PARAM_HDR_KP1_POS, PARAM_HDR_*	HDR mode, Knee points for exposure mode	HDR not supported in 0.2. Expect to be supported in 1.0.
Int: PARAM_X_LIMIT_MIN, PARAM_X_LIMIT_MAX	In pixel. Used for AGC statistics collection	Gocator uses a different dynamic exposure control scheme
Int: PARAM_SATURATION_COUNT	Number of saturated pixels per frame	Not supported in 0.2. Expect to be supported in 1.0
Int: PARAM_AVERAGE_FRAME_INTENSITY	Average intensity of the frame Same question as above	Not available. Is this needed
Int: PARAM_DYN_SENSOR_CTRL_ENABLE	Dynamic sensor control	Not supported
Int: PARAM_GAIN	Digital / analog gain	Digital gain configuration is supported in 0.2. Analog gain will be supported by 1.0. Settings are under <G5 Sensor>/Camera

Int: PARAM_REG_PULSE_WIDTH_FLOAT	Pulse width (Exposure)	<G5 Sensor>/SingleExposure
Int: PARAM_AGC_ENABLED, PARAM_AGC_<Other parameters>	AGC Parameters	Not supported in 0.2. Will be supported in 1.0
Int: PARAM_PULSE_CURRENT_MIDDLE, PARAM_PULSE_CURRENT_EDGES	Pulse current control in the middle section and edge section of the sensor	Intensity control supported for all LEDs, i.e. no separate middle/edge setting LED intensity currently supported up to maximum continuous. By 1.0 will also expect to support high power / low duty cycle operation Settings are under <G5 Sensor>/Intensity
Int: PARAM_INTERLEAVE_<Parameters>	Different exposures for even and odd rows	Not used commonly and not supported

Layer Control Parameters

FsSDK Parameters	Purpose	GoPxL SDK Equivalent
Int: PARAM_REORDERING, PARAM_REORDERING_SPAN, PARAM_REORDERING_DEVIATION,	Layer / line reordering, separation	GoPxL data protocol uses TCP instead of UDP packets for communications. No reordering required.
PARAM_CLUSTERING_DISTANCE_<XYZ>	Layer separation	Clustering support planned for 1.0
Int: PARAM_MAX_POINT_COUNT	Maximum point count per profile Need to be changed before grabbing	Configurable with layer count parameters
Int: PARAM_LAYER_INTENSITY_TYPE, PARAM_LAYER_MAX_THICKNESS, PARAM_LAYER_MIN_THICKNESS, PARAM_LAYER_EFFECTIVE_REFRACTIVE_INDEX	Layer specific settings	See layerSpecificParams

Spot Detection Parameters

FsSDK Parameters	Purpose	GoPxL SDK Equivalent
Int: PARAM_PEAK_THRESHOLD	Intensity data based filtering	<G5 Sensor>/spotDetection/threshold
Int: PARAM_PEAK_X_FILTER,	Peak detector X smoothing parameters	Peak X filter is not supported in Gocator <G5 Sensor>/advanced/SpotDetection
PARAM_TRIM_EDGES,	Trim Edge filter	Edge trimming planned for future release.
INT: PARAM_PEAK_FIR_LENGTH PARAM_PEAK_AVER_FIR_LENGTH	FIR length	<G5 Sensor>/advanced/SpotDetection/edgeWindow and averageWindow
Int: PARAM_SIGNAL_DETECTION_FILTER_LENGTH, PARAM_PEAK_AVERAGE_INTENSITY_FILTER_LENGTH		See advanced spot parameter types <G5 Sensor>/advanced/SpotDetection

Measurements & Outputs

FsSDK Parameters	Purpose	GoPxL SDK Equivalent
Int: PARAM_Z_COMPENSATION_<Options>	Z Compensation controls	Internal use only. Not supported with initial release.
Int: PARAM_THICKNESS_MODE	Calculates thicknesses for layers in line and batch callback functions. The first surface is returned as a world z-coordinate and /// following layers are calculated as thickness values. Note that thickness mode parameter is not saved to the recipe	Require setting up measurement tools in GoPxL
Int: I/O controls	IO control for peak warning LED, input and output states	Achieved thru Gocator output settings Details to be confirmed