LMI TECHNOLOGIES

Gocator Firmware 5.3.33.49 – Release Notes

Firmware Version 5.3.33.49 for G200 sensors
Document Revision A

Compatibility

- Devices Supported
  - Gocator 210, 230, 240 and 250 Multi-Point Sensors
  - Gocator 205 Vision Sensors
- Web Interface
  - The web interface requires Chrome, Firefox or Microsoft Edge version 79 or later.
  - Old Edge versions are only supported in a limited fashion (see known issues for details).
- Backwards compatibility:
  - 5.3.33.49 GoWebScanSDK is compatible with applications using older 5.x SDK and firmware
  - 5.3.33.49 sensor firmware is compatible with older 5.x SDK and applications
  - 5.3.33.49 example application is **not** compatible with older 5.x SDKs
  - For users who have customized the SDK, please compare all files in the SDK with your customized SDK since the changes in this release are extensive.

Improvements

Calibration More Tolerant of Angles and Noise	Possible misalignment in the vision data has been addressed by allowing the calibration bar to have some roll with respect to the sensors. The calibration process is also now more resilient to scratches and artifacts on the calibration bar. This includes improvements to X coverage warnings which are less likely to be triggered by these scratches and artifacts.
Improved Locator Detection	Locator detection during calibration has been made more robust.
Gamma Colour Support	Added support for applying gamma corrections to vision board images. The example program has been updated to demonstrate this functionality, which can be enabled by setting the *visionGammaEnabled* flag to true. As part of gamma support, a number of new utility functions have been added, including *GoWebScanUtils_Array2ToImage* which replaces the function *arrayToImage* in the example program.

Bug Fixes

Fixed Camera Desync Caused by Data Drops	Fixed a firmware bug that could cause cameras to become desynchronized due to data drops resulting in incorrect Y offsets between cameras in the board data.

+1 604 636 1011
+1 604 516 8368
info@lmi3D.com
www.lmi3D.com
HEADQUARTERS
9200 Glenlyon Parkway
Burnaby, BC V5J 5J8
Canada

Data Drops Now Cleared Properly Between Runs	Fixed a bug where lanes were not correctly reset when cleared, which caused data drops from a previous run to interfere with a new run.
Calibration Replay Pipeline Fix	Fixed a bug in which reprocessing a calibration recording would not always complete due to some data not making it through the pipeline.

Known Issues

Translations Incomplete	Not all English text is translated in every language.

Detailed Release Features

These notes detail all changes, improvements and new features for GoWebScanSDK that have been incorporated since the release of GoWebScanSDK 5.3.33.39.

Changes to SDK

- Support has been added to convert images from the linear to gamma colour space. This includes adding several utility functions:
    - GoWebScanUtils_Array2ToImage. Converts a kArray2 to a kImage.
    - GoWebScanUtils_8uTo64f. Converts k8u values to k64f.
    - GoWebScanUtils_Round64fTo8u. Clamps, rounds, and converts k64f values to k8u.
    - GoWebScanUtils_Scale64f. Multiplies each element in an array of k64f values by a scalar.
    - GoWebScanUtils_ApplyExponentTo64f. Raises each element in an array of k64f values to a power using the pow() function.
    - GoWebScanUtils_PopulateGammaLookup. Constructs a gamma correction lookup table in which the value at an index is calculated by normalizing the index (from 0-255 to 0-1), applying the given exponent, and converting back to k8u.
    - GoWebScanUtils_ApplyGammaLookup. Applies a precomputed gamma lookup table to an array of kRgb pixels. Each R, G, and B channel is independently remapped through the lookup table.
    - Added define GO_WEB_SCAN_UTILS_LINEAR_TO_GAMMA_EXPONENT_ESTIMATE (1/2.2), the exponent estimate to convert images from the linear space to the gamma space.
    - Added define GO_WEB_SCAN_UTILS_GAMMA_TO_LINEAR_EXPONENT_ESTIMATE (2.2), the exponent estimate to convert images from the gamma space to the linear space.

Changes to Example Program

- The calibration submenu can now be exited and re-entered without losing the calibration file.

- The calibration warning indicating camera data loss has been updated to mention that it may be ignored only if it occurs while replaying previously recorded data.

Bug Fixes and Improvements

- Fixed a firmware bug that could cause cameras to become desynchronized due to data drops.
- System calibration has been updated to be more robust against noise. System calibration now also allows the calibration bar to have some roll without resulting in misalignment in the vision data.
- System calibration has also been improved to be more robust against noise such as scratches and other artifacts which would have otherwise caused calibration issues.
- Locator detection in system calibration has also been improved.
- The warning for insufficient data during calibration should no longer be triggered if the error for insufficient calibration data already has been triggered.
- Fixed a bug where lanes were not correctly reset when cleared, which caused data drops from a previous run to interfere with a new run.

New Features

This release includes gamma colour support as a new feature.

Gamma Colour Support

Support has been added for converting vision images from the linear sRGB colour space to the gamma sRGB colour space if gamma correction is desired by the user. This includes the addition of members kBool visionGammaEnabled and kArray1 gammaLookup to DataContext in the example application. visionGammaEnabled is a flag which must be set to true in order to enable the conversion to gamma. gammaLookup is a precalculated lookup table for applying gamma to 8-bit values.

There are multiple ways to implement gamma conversion. Since the gamma conversion in the example program is being applied to RGB images in which each channel (R, G, and B) is 8 bits, it is

faster to create a lookup table of gamma values for each integer from 0 to 255. This is what helper function GoWebScanUtils_PopulateGammaLookup does:

- Create a size 256 array of k8u values from 0 to 255.
- Convert this to k64f with GoWebScanUtils_8uTo64f.
- Normalize this by scaling it by a factor of 1 / 255 with GoWebScanUtils_Scale64f.
- Apply the gamma exponent with GoWebScanUtils_ApplyExponentTo64f.
- Denormalize this by scaling it by a factory of 255 with GoWebScanUtils_Scale64f.
- Convert this back to k8u with GoWebScanUtils_Round64fTo8u.

```
C/C++

GoWebScanFx(kStatus) GoWebScanUtils_PopulateGammaLookup(kArray1* output, k64f
    gamma, kAlloc alloc)

{

...

        // Convert to k64f

        kTest(kArray1_Construct(&dataFloat, kTypeOf(k64f), k8U_MAX + 1, alloc));

        kTest(GoWebScanUtils_8uTo64f(gammaData, (k64f*)kArray1_Data(dataFloat),
    kArray1_Count(gammaInput)));



        // Normalize

        kTest(GoWebScanUtils_Scale64f((k64f*)kArray1_Data(dataFloat),
    (k64f*)kArray1_Data(dataFloat), kArray1_Count(dataFloat), 1.0 /
    (k64f)k8U_MAX));



        // Apply linear to gamma correction

        kTest(GoWebScanUtils_ApplyExponentTo64f((k64f*)kArray1_Data(dataFloat),
    (k64f*)kArray1_Data(dataFloat), kArray1_Count(dataFloat), gamma));



        // Denormalize

        kTest(GoWebScanUtils_Scale64f((k64f*)kArray1_Data(dataFloat),
    (k64f*)kArray1_Data(dataFloat), kArray1_Count(dataFloat), (k64f)k8U_MAX));
```


```
        // Convert back to k8u
        kTest(kArray1_Construct(&outputArray, kTypeOf(k8u), k8U_MAX + 1, alloc));
        kTest(GoWebScanUtils_Round64fTo8u((k64f*)kArray1_Data(dataFloat),
   (k8u*)kArray1_Data(outputArray), kArray1_Count(dataFloat)));

        *output = outputArray;
...
    return kOK;
}
```

If visionGammaEnabled is enabled in the example program, the lookup array gammaLookup will be applied to vision data with GoWebScanUtils_ApplyGammaLookup.