Seekware™ LinuxSDK

User Guide
SDK v2.9
August 13,
2018

Seekware™ Linux SDK

Table of Contents

Welcome

The Seekware™ SDK was created for developers who want to use Seek Thermal cameras in their own projects. The SDK is designed to be simple to use while also providing access to key capabilities of the camera. We offer the Seekware™ SDK for multiple platforms with a common the API.

Supported Cameras

Compact

CompactPro

J1 Platform Core

J2 Platform Core

J3 Starter Kit

Camera		Image	Speed	SDK OS
Compact	PIR-206	206 x 156	<9Hz	Linux
CompactXR	PIR-206	206 x 156	<9Hz	Linux
Compact Pro	PIR-320	320 x 240	<9Hz	Linux
J1 Platform Core	PIR-206	206 x 156	<9Hz	Linux
J2 Platform Core	PIR-206	206 x 156	<9Hz	Linux
J3 Platform Core	PIR-320	320 x 240	<9Hz	Linux

Table 1 - Supported Platforms

NOTE: Starter Kits that run at higher frame rates are available on special request.

Installation Instructions

Linux

Install required packages

Debian-based Systems

sudo apt-get update	(ensures that apt-get pulls the latest packages)
sudo apt-get install libusb-1.0	(installs libusb drivers)
sudo apt-get install libusb-dev	(installs libusb development libraries)
sudo apt-get install libsdl2-dev	(installs SDL2 development libraries)

Fedora-based Systems

dnf update	(ensures that apt-get pulls the latest packages)
dnf install libusb-1.0	(installs libusb drivers)
dnf install libusb-devel	(installs libusb development libraries)
dnf install SDL2-devel	(installs SDL2 development libraries)

Execute the installation script

For ARM v7 or greater targets:

```
./install-arm-linux-gnueabihf-2.9.sh
```

For i686 targets:

```
./install-i686-linux-gnu-2.9.sh
```

For x86_64 targets:

```
./install-x86_64-linux-gnu-2.9.sh
```

For High-Silicon Cortex-A7 targets:

```
./install-arm-hisiv300-linux-uclibcgnueabi-2.9.sh
```

For Rockchip RV1108 targets:

```
./install-arm-rkcvr-linux-uclibcgnueabihf-2.9.sh
```

NOTE: If the install script does not run due to lack of permissions, type ‘sudo chmod +x <script>’. This will set the execute bits on the script file.

Build the sample apps

In each sample directory, type:
“make clean”, then “make”

New Linux SDK Features

The Seekware™ API

This is the definition of all the data structures and callable routines that are available to the maker. The API is based on the C programming language and specifies how software components should interact.

The Seekware™ Linux SDK Libraries

The Seekware™ Linux SDK Libraries currently support Linux® platforms, target armv7 (32-bit only), and aarch64.

API Functions

All functions return with a value of the type `sw_retcode`.

Return Code	Meaning
SW_RETCODE_NONE	No error has been detected
SW_RETCODE_NOTOPENED	Device is not opened
SW_RETCODE_OPENEX	Device is already opened exclusively
SW_RETCODE_BPARAM	Bad parameter
SW_RETCODE_NOFRAME	Frame processing error
SW_RETCODE_ERROR	Generic error
SW_RETCODE_OFLOW	Some kind of overflow
SW_RETCODE_USBERR	USB had an error; we need to restart the camera
SW_RETCODE_SETONLY	Setting is write only
SW_RETCODE_GETONLY	Setting is read only

Table 2 – API Return Codes

Error Recovery

After finding a camera and opening it for use, if a function returns a value other than `SW_RETCODE_NONE`, the software should call `Seekware_Close`, then `Seekware_Open` to properly recover the connection to the camera. This SDK has been designed to gracefully recover from errors when this process is employed.

Seekware_Find

```
sw_retcode Seekware_Find(psw *pswlist[], int length, int *numfound)
```

Description

Search the target environment for all connected devices. Then, starting at index zero-fill the `psarray` with a pointer to a device structure for each connected device up to `length`, then set `numfound` to the number of devices found. If there are more than `length` devices connected, fill the array, set `numfound` to length and return `SW_RETCODE_OFLOW`. The `sw` structure contains `frame_rows` and `frame_cols` fields which indicate the rows and columns of the attached camera.

Parameter(s)

`pswlist[]`	A pointer to an array of `psw` pointers allocated by the caller.
`length`	The length of the caller-supplied pointer array.
`numfound`	The number of devices found in the target environment.

SW Structure Definition

This structure contains camera specific information to describe attached devices. It also contains OS specific information that is used to manage devices and their use.

```
    typedef struct sw {// Device information
         uint16_t model,
         char serialNumber[13];
         char modelNumber[17];
         char manufactureDate[33];
         uint8 fw_version_major;
         uint8 fw_version_minor;
         uint8 fw_build_major;
         uint8 fw_build_minor;
         uint16 frame_rows;
         uint16 frame_cols;
    #ifdef __linux__
         // Linux Fields
         libusb_device_handle * lusb_dev_handle;
         libusb_transfer_status * lusb_status;
    #endif
         sw_retcode retcode; // Latest return code
         void *sdkPrivate;   // SDK Private Data
    } sw, *psw
```

The `model` field is of the sw_model type given below:

```
    typedef enum sw_model {
        SEEK_MODEL_206_WFOV = 0,
        SEEK_MODEL_206_WFOV_FF,
        SEEK_MODEL_206_NFOV,
        SEEK_MODEL_206_NFOV_FF,
        SEEK_MODEL_320_WFOV,
        SEEK_MODEL_320_WFOV_FF,
        SEEK_MODEL_320_NFOV,
        SEEK_MODEL_320_NFOV_FF
    } sw_model;
```

The `serialNumber` field contains a null terminated string with the12-digit camera serial number.
The `modelNumber` field contains a null terminated string with camera model number.
The `fw_version/build` fields report the camera firmware version and build numbers.
The `frame_rows` and `frame_cols` fields report the image data rows and columns.

Seekware_Open

```
sw_retcode Seekware_Open(psw id)
```

Description

Open the device for use, allocate memory, begin acquiring thermal data then return. Open devices are available exclusively to the instance of the SDK that opened them. A call to this function on a device that is already open should return `SW_RETCODE_OPENEX`.

Parameter(s)

`id`	A pointer to a Seekware device structure.

Seekware_Close

```
sw_retcode Seekware_Close(psw id)
```

Description

Close the device, release memory and terminate any events.

Parameter(s)

`id`	A pointer to a Seekware device structure.

Seekware_GetSdkInfo

`sw_retcode Seekware_GetSdkInfo(psw id, sw_sdk_info *info)`

Description

Returns a structure containing information about the SDK.

Parameter(s)

id	A pointer to a Seekware device structure.
info	A pointer to a sw_sdk_info structure.

SDK Info Structure

The sw_sdk_info structure contains data necessary to uniquely identify the SDK and internal components.

```
typedef struct sw_sdk_info {
     uint8 sdk_version_major,      // SDK version number
     uint8 sdk_version_minor,
     uint8 sdk_build_major,
     uint8 sdk_build_minor,
     uint8 lib_version_major,      // Library version number
     uint8 lib_version_minor,
     uint8 lib_build_major,
     uint8 lib_build_minor
} sw_sdk_info;
```

Seekware_GetImage

```
sw_retcode Seekware_GetImage(
     psw id, uint16_t *binary, float *temperature, uint32_t *display
)
```

Description

This function grabs the next available image from the camera in one or more of the available formats. If any of the output formats is not necessary, the caller may supply a NULL pointer for that format and that parameter shall be ignored.

If a buffer is supplied, then it must be appropriately sized based on the data type and the number of pixels in the image. If a buffer is not supplied, then the computations and memory allocation required for that function shall not be performed.

Parameter(s)

`id`	A pointer to a Seekware device structure.
`binary`	A pointer to a buffer to hold raw filtered image data.
`temperature`	A pointer to a buffer to hold full frame temperature data.
`display`	A pointer to a buffer to hold AGC corrected display data.

Seekware_GetImageEx

```
sw_retcode Seekware_GetImageEx (
     psw id, uint16_t *binary, float *temperature, uint32_t *display
)
```

Description

This function is identical to `Seekware_GetImage` except that it appends telemetry data to an extra row immediately following the image data. **Therefore, for this function the `binary` buffer must be sized to include one additional row.** The extra row contains telemetry data supplied by the Seekware Library for each frame. The telemetry data definition follows:

Telemetry Row Index	Value	Description
0	Field Count LSW	The field count is the index of each frame that comes from the sensor. It continues to increment even during shutter closures therefore, it can be used to detect shutter closures because shutter closures cause a discontinuity in the field count sequence.
1	Field Count MSW	
2	Temperature Diode count value	The temp diode count value is an uncalibrated, raw sampling of the temperature diode voltage.
3	EnvTemp LSW	EnvTemp is the estimated environment temperature based on FPA.
4	EnvTemp MSW	

Table 3 - Telemetry Data Definition

Parameter(s)

`id`	A pointer to a Seekware device structure.
`binary`	A pointer to a buffer to hold raw filtered image data + 1 telemetry line.
`temperature`	A pointer to a buffer to hold full frame temperature data.
`display`	A pointer to a buffer to hold AGC corrected display data.

Seekware_GetSetting

```
sw_retcode Seekware_GetSetting (
     psw id, sw_settings index, int *value
)
```

Description

Gets the value of the specified setting.

Parameter(s)

`id`	A pointer to a Seekware device structure.
`index`	The setting index (see Table 4 - Setting Index Values). Must be less than or equal to SETTING_THERMOGRAPHY_VERSION.
`value`	A pointer to the location to write the setting value.

Seekware_GetSettingEx

```
sw_retcode Seekware_GetSettingEx (
     psw id, sw_settings index, void *value, uint32_t bytes
)
```

Description

Writes the requested setting into value.

Parameter(s)

`id`	A pointer to a Seekware device structure.
`index`	The setting index.
`value`	A pointer to the storage location of value
`bytes`	The size of value in bytes.

Seekware_SetSetting

```
sw_retcode Seekware_SetSetting (psw id, sw_settings index, int value)
```

Description

Sets the value of the specified setting.

Parameter(s)

`id`	A pointer to a Seekware device structure.
`index`	The setting index (see Table 4 - Setting Index Values). Must be less than or equal to SETTING_THERMOGRAPHY_VERSION.
`value`	The setting value.

Seekware_SetSettingEx

```
sw_retcode Seekware_SetSettingEx(psw id, sw_settings setting, void
*value, uint32_t bytes)
```

Description

Sets the value(s) of the requested setting using the provided value(s).

Parameter(s)

`id`	A pointer to a Seekware device structure.
`setting`	The setting index.
`value`	A pointer to the setting value(s).
`bytes`	The size of value in bytes.

SW_SETTINGS

Various SDK settings can be queried and changed by calling the Seekware_GetSetting/Ex and Seekware_SetSetting/Ex functions respectively. The settings are selected by the index parameter. The following table lists settings provides descriptions for each settings in the sw_settings enum.

Setting Index	Set/Get	Description
SETTING_ACTIVE_LUT	Set/Get	The active LUT
SETTING_TEMP_UNITS	Set/Get	Temperature units
SETTING_TIMEOUT	Set/Get	Communications timeout
SETTING_CONTROL	Set/Get	Control settings
SETTING_EMISSIVITY	Set/Get	Emissivity
SETTING_BACKGROUND	Set/Get	Background temperature
SETTING_TEMP_DIODE_ROOM	Get	Factory temperature
SETTING_TEMP_DIODE_SLOPE	Get	Slope of thermography diode
SETTING_TEMP_DIODE_OFFSET	Get	Thermography diode offset
SETTING_THERMOGRAPHY_VERSION	Get	Thermography version
SETTING_GLOBAL_THERM_ADJUST	Set	Global temperature offset
SETTING_SCENE_THERM_ADJUST	Set	Temperature offset for a scene
SETTING_ENVIRONMENT_THERM_ADJUST	Set	Temperature offset for an environment
SETTING_SPECIFIC_THERM_ADJUST	Set	Temperature offset for a scene and an environment
SETTING_TRANSIENT_CORRECTION_ENABLE	Set/Get	Transient correction
SETTING_TRANSIENT_CORRECTION_PARAMS	Set/Get	Amplitude and decay for transient correction
SETTING_SMOOTHING	Set/Get	Image smoothing
SETTING_AUTOSHUTTER	Set/Get	Auto shutter
FEATURE_MINMAX	Get	Min/Max with coordinates
FEATURE_OEM	Set/Get	Distortion features

Table 4 - Setting Index Values

SETTING_ACTIVE_LUT

For this setting, Seekware_GetSetting returns a value in the following table and Seekware_SetSetting with an index parameter set to one of the following values will change the LUT used to generate display imagery.

Setting Value	Description
SW_LUT_WHITE_NEW	White hot
SW_LUT_BLACK_NEW	Black hot
SW_LUT_SPECTRA	Rainbow
SW_LUT_PRISM	Modified rainbow
SW_LUT_TYRIAN_NEW	Purple
SW_LUT_AMBER_NEW,	Yellow-orange
SW_LUT_IRON_NEW	Classic Hot Iron
SW_LUT_HI	High temp highlight
SW_LUT_HILO	High and low temp highlight

Table 5 - LUT Values

SETTING_TEMP_UNITS

For this setting, `Seekware_GetSetting` returns a value in the following table and `Seekware_SetSetting` with an `index` parameter set to one of the following values will change temperature units used for temperature data returned from `Seekware_GetImage` and `Seekware_GetImageEx`.

Setting Value	Temperature Units
SW_TEMP_F	Fahrenheit
SW_TEMP_C	Celsius
SW_TEMP_K	Kelvin

Table 6 - Temperature Units

SETTING_TIMEOUT

The USB transaction timeout is determined by the value of this setting. The `value` field is an int containing the timeout period in milliseconds. Thus, a value of 3000 would cause the USB timeout to be 3 seconds. The default timeout is 5000.

SETTING_CONTROL (legacy)

This setting allows control over certain camera binary settings as described in the following table. This setting is a bitmask of the values in the table. To set the setting, the software must bitwise OR the setting on the value sent. To get the setting, the software must mask the value returned to see if the bit is set or not.

Setting Value	Bit Description
SEEKWARE_CTRL_SMOOTHING	Read/Write. Sets/indicates that image smoothing is enabled.
SEEKWARE_CTRL_AUTOSHUTTER	Write only. Mask off to disable auto-shutter.

Table 7 - Control Settings

SETTING_EMISSIVITY

This setting determines the assumed surface emissivity for temperature readings. Since the required value is a floating-point number but the setting must be an integer, this value is set with the intended value x 100. Thus, for a desired emissivity setting of 0.98, this function should be called with a value of 98. Reading returns a number in the same x 100 format. The default is 0.98.

SETTING_BACKGROUND

This setting determines the assumed background temperature for temperature readings. This value must be given and shall be reported in whole degrees Celsius regardless of the temperature units setting. The default background temperature is 25°C.

SETTING_TEMP_DIODE_ROOM

This is a read-only setting that reports factory temperature in order to estimate the environmental temperature.

SETTING_TEMP_DIODE_SLOPE

This is a read-only setting that reports the slope of the FPA thermography diode.

SETTING_TEMP_DIODE_OFFSET

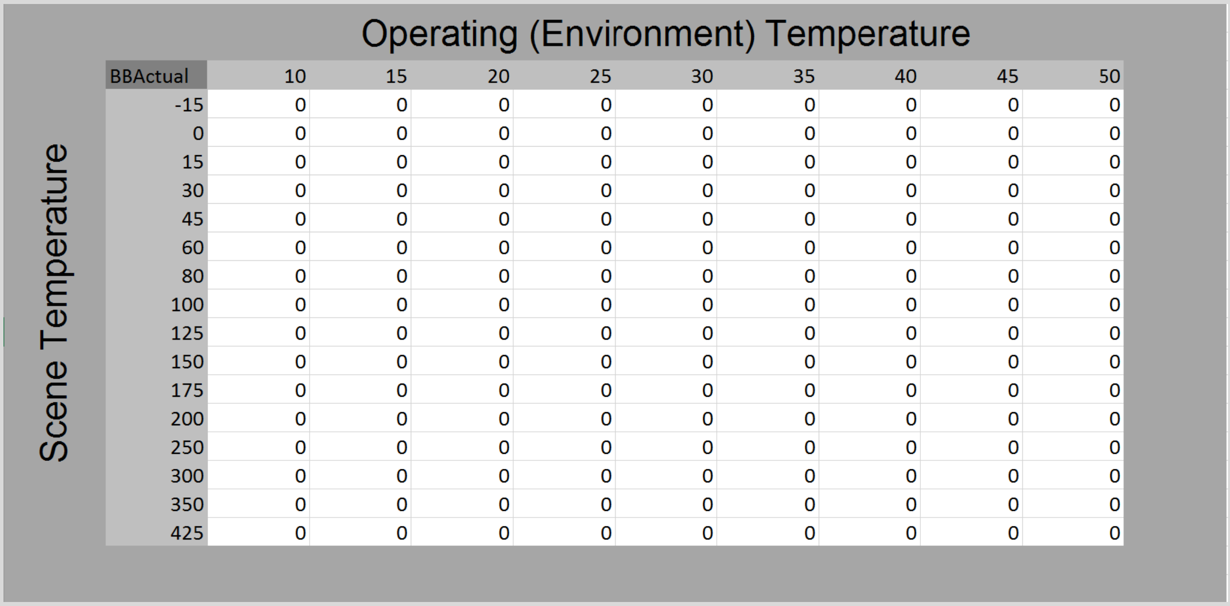
This is a read-only setting that reports the offset of the FPA thermography diode.

SETTING_THERMOGRAPHY_VERSION

This is a read-only setting that reports the version of the thermography code used by the calibration test station when the connected camera was calibrated.

Seekware_SettingEx Setttings:

Note: The following four settings are used to make thermography adjustments based on a device's scene and operating temperature. Before applying any of these settings, none of the device's thermography values are adjusted, as shown in the thermography adjustment table below:



Operating (Environment) Temperature

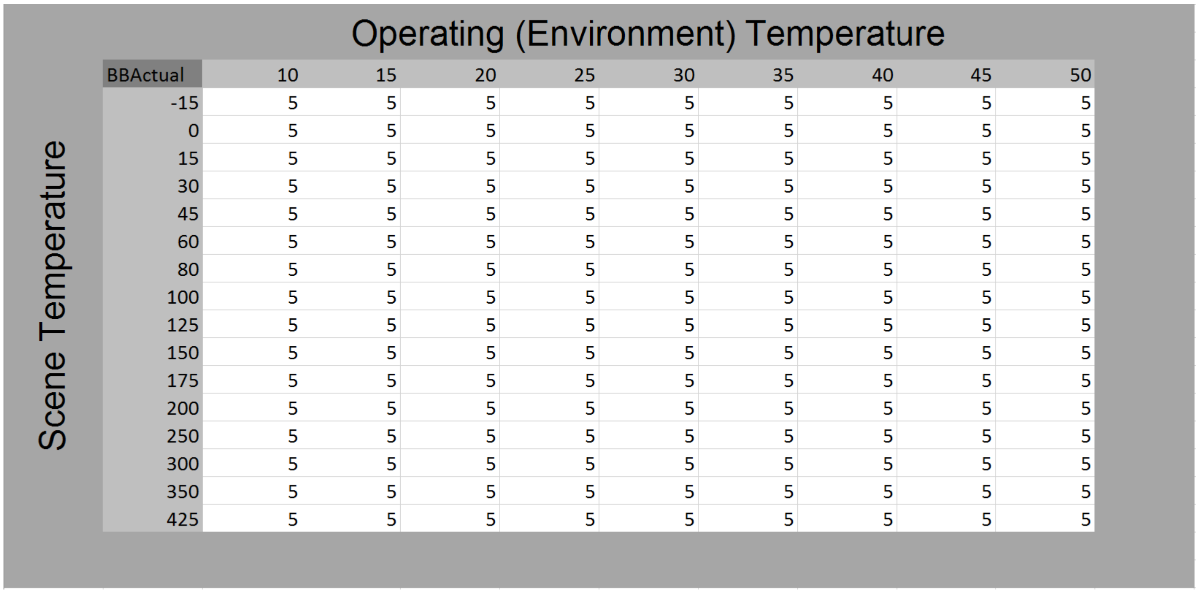
Scene Temperature

BBActual	10	15	20	25	30	35	40	45	50
-15	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0
125	0	0	0	0	0	0	0	0	0
150	0	0	0	0	0	0	0	0	0
175	0	0	0	0	0	0	0	0	0
200	0	0	0	0	0	0	0	0	0
250	0	0	0	0	0	0	0	0	0
300	0	0	0	0	0	0	0	0	0
350	0	0	0	0	0	0	0	0	0
425	0	0	0	0	0	0	0	0	0

Table 8 – Unadjusted Thermography Table

SETTING_GLOBAL_THERM_ADJUST

This is a write-only setting that allows the user to set a global temperature offset. When applying a SETTING_GLOBAL_THERM_ADJUST of 5 degrees, all thermography values are adjusted by 5 degrees.



Operating (Environment) Temperature

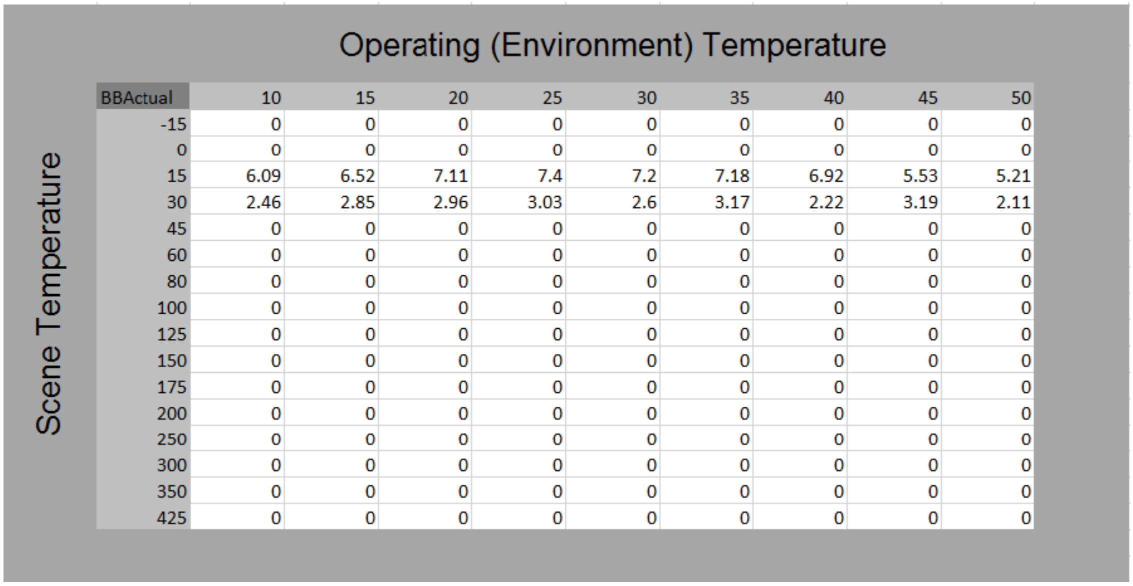
Scene Temperature

BBActual	10	15	20	25	30	35	40	45	50
-15	5	5	5	5	5	5	5	5	5
0	5	5	5	5	5	5	5	5	5
15	5	5	5	5	5	5	5	5	5
30	5	5	5	5	5	5	5	5	5
45	5	5	5	5	5	5	5	5	5
60	5	5	5	5	5	5	5	5	5
80	5	5	5	5	5	5	5	5	5
100	5	5	5	5	5	5	5	5	5
125	5	5	5	5	5	5	5	5	5
150	5	5	5	5	5	5	5	5	5
175	5	5	5	5	5	5	5	5	5
200	5	5	5	5	5	5	5	5	5
250	5	5	5	5	5	5	5	5	5
300	5	5	5	5	5	5	5	5	5
350	5	5	5	5	5	5	5	5	5
425	5	5	5	5	5	5	5	5	5

Table 9 – Global Offset Thermography Table

SETTING_SCENE_THERM_ADJUST

This is a write-only setting that allows the user to set a temperature offset for a specific scene. When applying a SETTING_SCENE_THERM_ADJUST, the two calibrated scene temperatures bounded by the adjustment are offset at all camera operating temperatures.



Operating (Environment) Temperature

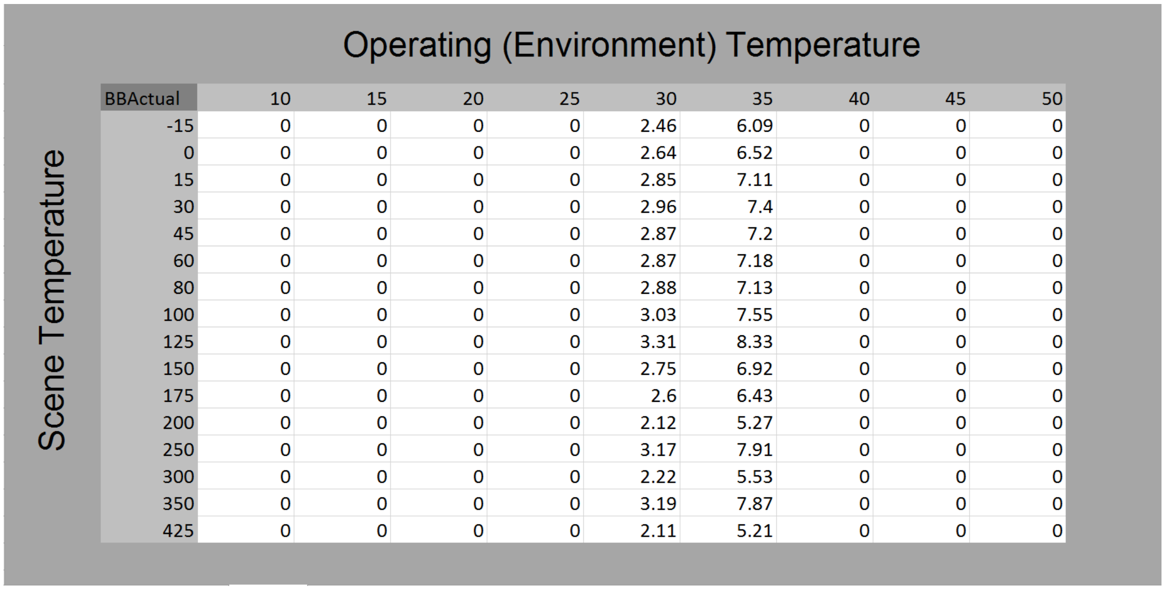
Scene Temperature

BBActual	10	15	20	25	30	35	40	45	50
-15	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
15	6.09	6.52	7.11	7.4	7.2	7.18	6.92	5.53	5.21
30	2.46	2.85	2.96	3.03	2.6	3.17	2.22	3.19	2.11
45	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0
125	0	0	0	0	0	0	0	0	0
150	0	0	0	0	0	0	0	0	0
175	0	0	0	0	0	0	0	0	0
200	0	0	0	0	0	0	0	0	0
250	0	0	0	0	0	0	0	0	0
300	0	0	0	0	0	0	0	0	0
350	0	0	0	0	0	0	0	0	0
425	0	0	0	0	0	0	0	0	0

Table 10 – Scene Offset Thermography Table

SETTING_ENVIRONMENT_THERM_ADJUST

This is a write-only setting that allows the user to set a temperature offset for a specific environment. When applying a SETTING_ENVIRONMENT_THERM_ADJUST, all calibrated scene temperatures are offset at the camera operating temperatures bounded by the adjustment.



Operating (Environment) Temperature

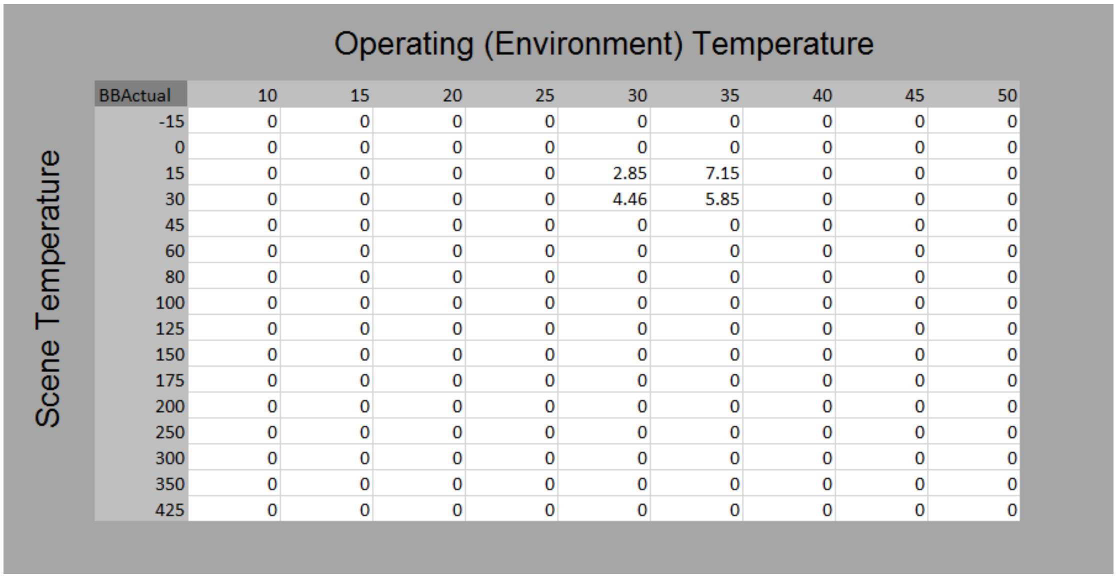
Scene Temperature

BBActual	10	15	20	25	30	35	40	45	50
-15	0	0	0	0	2.46	6.09	0	0	0
0	0	0	0	0	2.64	6.52	0	0	0
15	0	0	0	0	2.85	7.11	0	0	0
30	0	0	0	0	2.96	7.4	0	0	0
45	0	0	0	0	2.87	7.2	0	0	0
60	0	0	0	0	2.87	7.18	0	0	0
80	0	0	0	0	2.88	7.13	0	0	0
100	0	0	0	0	3.03	7.55	0	0	0
125	0	0	0	0	3.31	8.33	0	0	0
150	0	0	0	0	2.75	6.92	0	0	0
175	0	0	0	0	2.6	6.43	0	0	0
200	0	0	0	0	2.12	5.27	0	0	0
250	0	0	0	0	3.17	7.91	0	0	0
300	0	0	0	0	2.22	5.53	0	0	0
350	0	0	0	0	3.19	7.87	0	0	0
425	0	0	0	0	2.11	5.21	0	0	0

Table 11 – Environment Offset Thermography Table

SETTING_SPECIFIC_THERM_ADJUST

This is a write-only setting that allows the user to set a temperature offset for a specific scene and environment. When applying a SETTING_SPECIFIC_THERM_ADJUST, only calibrated temperature bounded by the correct scene and environment temperatures are offset.



Operating (Environment) Temperature

Scene Temperature

BBActual	10	15	20	25	30	35	40	45	50
-15	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	2.85	7.15	0	0	0
30	0	0	0	0	4.46	5.85	0	0	0
45	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0
125	0	0	0	0	0	0	0	0	0
150	0	0	0	0	0	0	0	0	0
175	0	0	0	0	0	0	0	0	0
200	0	0	0	0	0	0	0	0	0
250	0	0	0	0	0	0	0	0	0
300	0	0	0	0	0	0	0	0	0
350	0	0	0	0	0	0	0	0	0
425	0	0	0	0	0	0	0	0	0

Table 12 – Specific Offset Thermography Table

SETTING_TRANSIENT_CORRECTION_ENABLE

This setting enables a global transient correction of temperature readings

SETTING_TRANSIENT_CORRECTION_PARAMS

This setting determines the amplitude and decay used for transient correction. The offset is calculated based on the following equation where t is run time in seconds, and amplitude A and decay d are set by the user using the sw_transient_corr_params_t in Seekware.h. The parameters are not stored in camera memory and much be set after each device is opened.

$$Offset = A\, e^{-\frac{t}{d}}$$

Note: On cameras with a QVGA sensor, time (t) resets to zero only when the camera is powered off. On cameras with a 206 sensor, time resets to 0 each time Seekware_Open is called.

SETTING_SMOOTHING

This setting smooths the display image. Passing Seekware_SetSettingEx a value > 0 for this setting enables smoothing.

SETTING_AUTOSHUTTER

This setting enables/disables the auto shutter. Passing Seekware_SetSettingEx a value > 0 for this setting enables autoshutter. Passing 0 disables the shutter.

FEATURE_MINMAX

This feature returns the min/max temperature values of a device's scene and the x,y coordinates of these values. Use this feature as a parameter of Seekware_GetSettingEx.

FEATURE_OEM

When calling either Seekware_GetSettingEx or Seekware_SetSettingEx, structure the index/setting parameter as (FEATURE_OEM + SeekProvidedSetting), where SeekProvidedSetting will be provided after contacting Seek.

Seekware_GetSpot

```
sw_retcode Seekware_GetSpot(psw id, float *temp, float *min, float
*max)
```

Description

Retrieve the spot thermography for the center 6x6 pixels of the most recently processed image frame. If this function is called without specifying a temperature buffer in Seekware_GetImage or Seekware_GetImageEx or before calling Seekware_GetThermographyImage, Seekware_GetSpot will return SW_RETCODE_NOFRAME.

Parameter(s)

`id`	A pointer to a Seekware device structure.
`temp`	A float pointer to the place to put the spot (average) temperature.
`min`	A float pointer to the place to put the minimum temperature.
`max`	A float pointer to the place to put the maximum temperature.

Seekware_SetUserLut

```
sw_retcode Seekware_SetUserLUT(psw id, uint32_t lut_index, uint32_t
*lut_data, uint32_t length)
```

Description

Loads user LUT data to a USER LUT. Only supports ARGB32 look-up tables.

Parameter(s)

`id`	A pointer to a Seekware device structure.
`lut_index`	Index of the user LUT to set. SW_LUT_USER0 <= lut_index <= SW_LUT_USER4.
`lut_data`	A uint32_t array containing the LUT data to write. Should be NR_LUTCOLORS elements.
`length`	Number of LUT elements to write. Should be NR_LUTCOLORS.

Seekware_UploadFirmware

```
sw_retcode Seekware_UploadFirmware(psw id, const char* filename)
```

Description

Loads firmware into Seekware device. For use with firmware files provided separately by Seek.

Parameter(s)

`id`	A pointer to a Seekware device structure.
`filename`	The firmware file to be uploaded to Seekware device.

Seekware_GetThermographyImage

```
sw_retcode Seekware_GetThermographyImage(psw id, uint16_t* thermography, uint32_t num_elements)
```

Description

Returns a frame of fixed point uint16_t thermography values. To get temperature, apply the following formula: Temperature = (count / 64) – 40. Temperature units are controlled by SETTING_TEMP_UNITS.

Parameter(s)

`id`	A pointer to a Seekware device structure.
`thermography`	A pointer to a uint16_t frame buffer.
`num_elements`	Number of elements in the display buffer.

Seekware_GetDisplayImage

```
sw_retcode Seekware_GetDisplayImage(psw id, uint32_t* display , uint32_t num_elements)
```

Description

Returns a frame of ARGB display values with auto gain control enabled. Does not perform any thermography calculations.

Parameter(s)

`id`	A pointer to a Seekware device structure.
`display`	A pointer to a uint32_t frame buffer.
`num_elements`	Number of elements in the display buffer.

Sample Applications

Linux

Sample applications are normally stored in /usr/src. If you change the default installation directory when you install the SDK, the samples will be stored in the /src directory under your specified installation directory.

seekware-simple

Running the seekware-simple app:

./seekware-simple

seekware-test

Running the seekware-test app:

./seekware-test

The valid options for this app are:

-h | --help   Print usage information and exit.

--device <dev>   The name of the framebuffer device. Its default is "/dev/fb0".

-d | --double   Doubles the size of the displayed rectangle(s) both in horizontal and vertical directions.

-lut | --lut <l>   Sets the given LUT for RGB image.

Valid LUTs are reported with –h.

Also supports imagej LUT filenames.

seekware-upgrade

Running the seekware-upgrade app:

./seekware-upgrade <upgrade-file>

The valid options for this app are:

-h | --help   Print usage information and exit.

Adding Temperature Adjustment

Applying a temperature correction is a 2 step process:

1) Use one of the following typedefs in seekware.h to define the struct that holds the desired adjustment parameters:

    - sw_global_therm_adjust_t
    - sw_scene_therm_adjust_t
    - sw_environment_therm_adjust_t
    - sw_specific_therm_adjust_t

2) Call Seekware_SetSettingEx with one of the following settings defined in sw_settings and pass a pointer to the struct defined in (1):

    - SETTING_GLOBAL_THERM_ADJUST
    - SETTING_SCENE_THERM_ADJUST
    - SETTING_ENVIRONMENT_THERM_ADJUST
    - SETTING_SPECIFIC_THERM_ADJUST

Important Note:
At least 1 successful call to Seekware_GetImage or Seekware_GetImageEx must precede Seekware_SetSettingEx when applying a thermography adjustment, otherwise Seekware_SetSettingEx will fail and return SW_RETCODE_ERROR. To avoid this, it is recommended to wrap the calls to Seekware_SetSettingEx for thermography adjustments like the example code below:

Example Code: Apply a +5 degree offset to a 25 degree scene

```
if(Seekware_GetImageEx(dev, NULL, NULL, NULL) == SW_RETCODE_NONE){
        /*Apply temperature adjustments as needed*/

        sw_scene_therm_adjust_t scene_adjust;
        scene_adjust.scene_temp = 25.0f;
        scene_adjust.offset = 5.0f;
        Seekware_SetSettingEx(dev, SETTING_SCENE_THERM_ADJUST, &scene_adjust,sizeof(scene_adjust));
}
```

Revision History

v2.9

1. Added new API functions (Seekware_GetThermographyImage and Seekware_GetDisplayImage) that each return a frame of either fixed point thermography values or ARGB display values.
2. Added new features (FEATURE_MINMAX and FEATURE_OEM) that are used as parameters of Seekware_GetSettingEx and Seekware_SetSettingEx.
3. Deprecated API functions (Seekware_GetSetting and Seekware_SetSetting). For v3.0+, Seekware_SetSettingEx and Seekware_GetSettingEx will be renamed to Seekware_SetFeature and Seekware_GetFeature.

Attributions

1. **Linux Standard Base** is a trademark of The Linux Foundation.
2. **Linux** is a registered trademark of Linus Torvalds.
3. **Ubuntu** is a trusted open source product. Canonical owns and manages certain intellectual property rights in Ubuntu and other associated intellectual property (Canonical IP) and licenses the use of these rights to enterprises, individuals and members of the Ubuntu community in accordance with their IP rights Policy. Canonical's IP rights Policy. This policy is published by Canonical Limited (Canonical, we, us and our) under the Creative Commons CC-BY-SA version 3.0 UK license.
4. **ARM** is a registered trademark of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. **ARM6** (ARMv6) and **ARM7** (ARMv7) are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.
5. **Compact, CompactXR** and **Compact Pro** are trademarks of Seek Thermal, Inc.