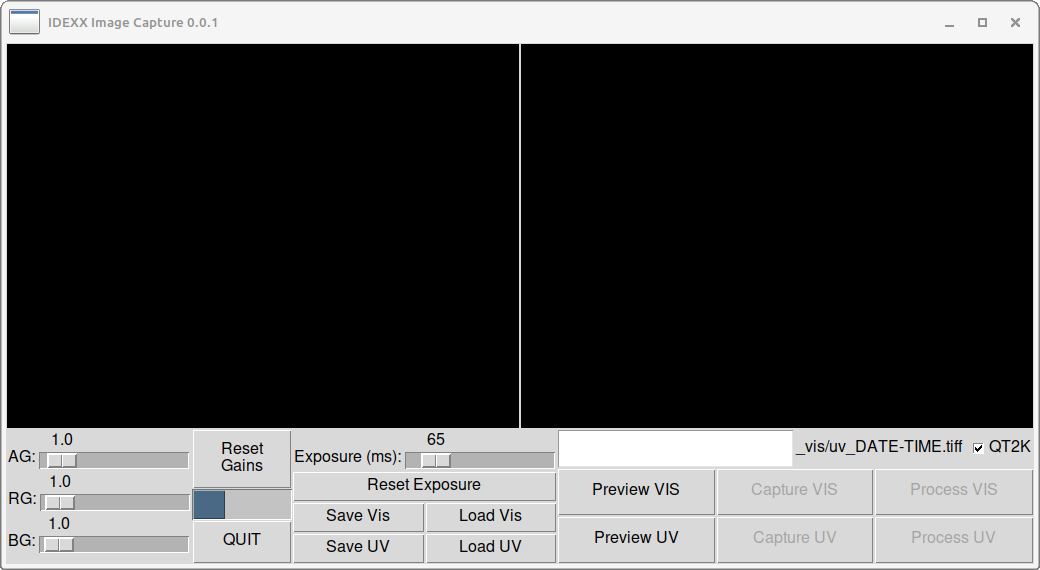
**Image Capture App**

Tom Radcliffe – 2024-02-06/2024-02-08

Figure 1: Image Capture App UI

**Overview**

The image capture app is a Python3 application designed to test out camera and GPIO functionality, and to be useful in taking test images for further processing and algorithm development. It has also been useful in testing the viability of various UI and other libraries on the RPi.

The UI is shown in Figure 1.

Run the application from the RPi command line with:

*python3 image\_capture\_app.py*

The app should start full-screen. If it does not, F11 can be used to toggle fullscreen mode. ESC can be used to exit it.

**NOTE**: The RPi must be ***connected to the Internet*** the first time the application is run on a given machine, unless that machine has previously had the python3-pil.imagetk package installed (sudo apt-get install python3-pil.imagetk). If it is not installed, the application will attempt to install it. Once it is installed the unit can be run offline.

**Camera Tuning File**

The camera tuning file is called imx477.2.json and is stored in the install directory. Any file in the user's documents directory is ignored, and should probably be removed to reduce the risk of confusion.

**Settings**

Four settings are available to the user:

1) Analog Gain (AG)

2) Red Gain (RG)

3) Blue Gain (RG)

4) Exposure Time (ms)

The gains are represented by floating point numbers that are changeable using the sliders with an increment of 0.2. This resolution can be refined if required.

The gains can be reset to 1.0 by hitting the "Reset Gains" button, and the exposure time to 65 ms by hitting the "Reset Exposure" button.

Two sets of gain/exposure values can be saved to the parameters.txt file (which is created if it does not exist). Set the gain/exposure values to ones you want based on experimentation with the preview functionality (see below) or whatever, and then hit "Save Vis" or "Save UV" to record these value.

They are written to the parameters.txt file on exit, not before.

Once created, these values can be loaded using "Load Vis" or "Load UV", and on normal exit will persist across restarts of the app or reboots of the computer.

**Preview Images**

Preview images can be taken at any time using the "Preview" buttons. They use the currently displayed gain and exposure time settings (exposure time is converted to micro-seconds before being passed to the camera). Preview images are lower resolution (640x480) and are not saved.

Lights on GPIO outputs 16, 20, 21, 13, 12, 6 are activated when a Vis image is captured.

UV LEDs must be controlled manually.

**Capture Images**

Capture images can only be taken when a filename tag has been entered. The tag will be lower-cased and all spaces will be replaced with underscores. The rest of the filename will be in the pattern shown: either "\_vis\_" or "\_uv\_" followed by an ISO 8601 date-time (because it's 2021) and a ".tiff" extension.

Captured images are saved to the "images" directory in the install directory. They are full res.

A metadata file for each image is stored in the "images" directory as well. It has the same name as the .tiff file but the ".tiff" extension is replaced with "\_metadata.txt".

There is currently no strong indication of when image capture has completed. Watch the button, and when it is no longer depressed the process is done. An additional indicator may be added in future.

**Process Images**

Once an image has been capture it can be processed. This will generate a tagged image in the images/processed/tagged directory under the install directory, which is displayed after processing.

Currently processing can fail for many reasons. These are not yet reported in the UI. If the warped raw image is not replaced by an unwarped tagged image, processing has failed. There may be output on the command line that will be useful in this case.

**Quit**

Hit the "Quit" button to quit the application. Reboot if the app won't restart (sometimes the camera hangs.)