Streaming FLIR Machine Vision Cameras on Embedded Systems

Technical Application Note TAN2014009

Revised 8/21/2017

Applicable Products ................................................................. 2
Application Note Description .................................................. 2
System Configuration ............................................................... 2
System Setup ........................................................................... 5
  ODROID-XU Getting Started ..................................................... 5
  Samsung Exynos 5250 Arndale Getting Started ......................... 6
  NVIDIA Jetson TK1 Getting Started ......................................... 7
  NVIDIA Jetson TX1 Getting Started ......................................... 8
  NVIDIA Jetson TX2 Getting Started ......................................... 9
  NVIDIA DRIVE PX Getting Started ....................................... 9
Benchmark Results ................................................................. 11
  ODROID-XU ........................................................................ 11
  Samsung Exynos 5250 Arndale ............................................... 12
  NVIDIA Jetson TK1 ............................................................... 14
  NVIDIA Jetson TX1 ............................................................... 15
  NVIDIA Jetson TX2 ............................................................... 15
  NVIDIA DRIVE PX ............................................................... 16
Troubleshooting ...................................................................... 17
  Image transfer fails to start when image size is bigger than 2 MB .......... 17
  Camera is detected but cannot stream ..................................... 17
  GigE camera frame rates are extremely low on Jetson TK1, TX1, or TX2 board ............................................... 17
Downloads and support .......................................................... 18
  Finding information ........................................................... 18
  Contacting technical support ............................................... 18
Applicable Products

- FLIR USB3 Vision cameras (excluding Blackfly®)
- FLIR GigE Vision cameras (excluding Blackfly®)

Application Note Description

This technical application note provides a summary and instructions for streaming FLIR machine vision cameras using FlyCapture®2 on ARM-based embedded boards. It includes examples of some of the more commonly used embedded boards: ODROID-XU; Samsung Exynos 5250 Amdale; and NVIDIA Jetson TK1, TX1, TX2, and DRIVE PX. The benchmark results show that embedded boards are able to support high speed machine vision applications. See TAN2012001 Getting Started with FlyCapture2 and ARM for more information on using FlyCapture2 on ARM devices in a Linux environment.

System Configuration

Note: These boards do not support Power over Ethernet (PoE), therefore all GigE models were powered externally via the GPIO. All GS3-U3 models were powered externally via the GPIO. The CM3 model was powered via the USB3 interface.

<table>
<thead>
<tr>
<th>ODROID-XU Board Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
</tr>
<tr>
<td><strong>GPU</strong></td>
</tr>
<tr>
<td><strong>RAM</strong></td>
</tr>
<tr>
<td><strong>Host Adapter Driver</strong></td>
</tr>
<tr>
<td><strong>Operating System</strong></td>
</tr>
<tr>
<td><strong>Software</strong></td>
</tr>
<tr>
<td><strong>Camera and Firmware</strong></td>
</tr>
<tr>
<td><strong>Samsung Exynos 5250 Arndale Specification</strong></td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td><strong>CPU</strong></td>
</tr>
<tr>
<td><strong>GPU</strong></td>
</tr>
<tr>
<td><strong>RAM</strong></td>
</tr>
<tr>
<td><strong>Hard Drive</strong></td>
</tr>
<tr>
<td><strong>Hard Drive Enclosure</strong></td>
</tr>
<tr>
<td><strong>Host Adapter Driver</strong></td>
</tr>
<tr>
<td><strong>Operating System</strong></td>
</tr>
<tr>
<td><strong>Software</strong></td>
</tr>
<tr>
<td><strong>Camera and Firmware</strong></td>
</tr>
</tbody>
</table>

| **NVIDIA Jetson TK1 Specification** |
|-----------------|-------------------------------------------------|
| **CPU**          | Tegra K1 SOC                                    |
| **GPU**          | Kepler GPU with 192 CUDA cores                 |
| **RAM**          | 2 GB                                            |
| **Host Adaptor Driver** | xhci-hcd                                      |
| **Operating System** | Ubuntu 14.04                                   |
| **Software**     | FlyCap 2.6.3.3 Linux ARM                        |
| **Camera and Firmware** | GS3-U3-23S6M (firmware 2.3.3.0)                |
|                  | GS3-PGE-23S6C (firmware 1.6.3.0)               |
|                  | FL3-GE-03S1M (firmware 1.26.3.0)               |

| **NVIDIA Jetson TX1 Specification** |
|-----------------|-------------------------------------------------|
| **CPU**          | Cortex A57                                       |
| **GPU**          | Maxwell GPU with 256 CUDA cores                 |
| **RAM**          | 4 GB                                             |
| **Host Adaptor Driver** | xhci-hcd                                      |
| **Operating System** | Ubuntu 14.04                                   |
| **Software**     | FlyCap 2.9.3.43 Linux ARM64                     |
| **Camera and Firmware** | GS3-U3-23S6M (firmware 2.22.3.0)              |
|                  | GS3-PGE-23S6M (firmware 1.10.3.0)               |
### NVIDIA Jetson TX2 Specification

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Cortex A57</td>
</tr>
<tr>
<td>GPU</td>
<td>Pascal GPU with 256 CUDA cores</td>
</tr>
<tr>
<td>RAM</td>
<td>8 GB</td>
</tr>
<tr>
<td>Host Adaptor Driver</td>
<td>xhci-hcd</td>
</tr>
<tr>
<td>Operating System</td>
<td>Ubuntu 16.04</td>
</tr>
<tr>
<td>Software</td>
<td>FlyCap 2.11.3.121 Linux ARM64</td>
</tr>
<tr>
<td>Camera and Firmware</td>
<td>GS3-U3-23S6M (firmware 2.25.3.0)</td>
</tr>
<tr>
<td></td>
<td>GS3-PGE-23S6M (firmware 1.13.3.0)</td>
</tr>
<tr>
<td></td>
<td>BFLY-PGE-23S6M (firmware 1.61.3.0)</td>
</tr>
</tbody>
</table>

### NVIDIA DRIVE PX Specification

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Tegra X1 64-bit processor</td>
</tr>
<tr>
<td>GPU</td>
<td>Maxwell GPU with 256 CUDA cores</td>
</tr>
<tr>
<td>RAM</td>
<td>4 GB</td>
</tr>
<tr>
<td>Host Adaptor Driver</td>
<td>xhci-hcd</td>
</tr>
<tr>
<td>Operating System</td>
<td>Ubuntu 15.04 with 3.18.21-tegra-g06aec38 Linux NVIDIA kernel</td>
</tr>
<tr>
<td>Software</td>
<td>FlyCapture 2.9.3.43 Linux ARM64</td>
</tr>
<tr>
<td>Camera and Firmware</td>
<td>BFLY-PGE-50S5C-C (firmware 1.53.3.0)</td>
</tr>
<tr>
<td></td>
<td>BFLY-PGE-23S6M-C (firmware 1.40.3.0)</td>
</tr>
<tr>
<td></td>
<td>CM3-U3-50S5M-CS (firmware 1.10.3.0)</td>
</tr>
</tbody>
</table>

**Note:** To achieve maximum bandwidth:

GigE cameras require a Gigabit adapter connected directly to an on-board USB port.

USB3 cameras require a custom kernel as described in [NVIDIA DRIVE PX Getting Started](#).
System Setup

ODROID-XU Getting Started

1. To boot from the microSD card, ensure boot media SW1-1 is set to OFF and SW2-2 is set to ON.
2. Download Ubuntu 13.10.
3. Unpack the files. For an .xz extension, use xz -d image.img.xz on cmd line.
4. Insert the microSD card in the Ubuntu PC.
   Note: Windows users can use Windows tool Win32DiskImager.exe to write an image to the microSD card.
5. Clear the destination media by entering:
   
   ```
   dd if=/dev/zero of=/dev/sdX bs=4M
   ```
   Note: /dev/sdX where X denotes the microSD card’s file storage.
6. Flash your image:
   
   ```
   sudo dd if=image.img of=/dev/sdX bs=4M
   ```
7. Enter sync before removing the media to make sure that all data is written.
8. Download FlyCapture2 ARM hard float package (flycapture.2.6.3.2_armhf.tar.gz or newer).
9. Retrieve dependency using command:
   
   ```
   sudo apt-get install libglademmm-2.4-dev
   ```
10. Extract FlyCapture2 ARM hard float package and run the following as root
    
    ```
    flycap2-conf
    ```
11. Reboot ODROID-XU.
12. In the terminal, change to FlyCapture2 ARM hard float directory’s bin folder.
13. Run the command:
    
    ```
    export LD_LIBRARY_PATH=../../lib
    ```
14. Start FlyCapture2 using the command:

```
./FlyCap2
```

### Samsung Exynos 5250 Arndale Getting Started

1. Download the Linaro image for Arndale. This configuration used [Version 13.11](#).
2. Insert the microSD card in the Ubuntu PC.
3. Write the image to a microSD card.
   
   **Note:** Windows users can use Windows tool `Win32DiskImager.exe` to write an image to the microSD card.

4. Download the FlyCapture2 ARM hard float package (`flycapture.2.6.3.2_armhf.tar.gz` or newer).
5. Extract the FlyCapture2 ARM hard float package and run `flycap2-conf` as root.
6. Reboot.
7. In the terminal, change to FlyCapture2 ARM hard float directory’s bin folder.
8. Run the command:

```
export LD_LIBRARY_PATH=./../lib
```

9. Start FlyCapture2 using the command:

```
./FlyCap2
```
NVIDIA Jetson TK1 Getting Started

1. Follow NVIDIA’s readme file and boot up the pre-installed Ubuntu 14.04 on TK1.

2. By default, Jetson TK1 does not provide support for USB3 cameras. To enable USB3 support:
   a. Go to https://developer.nvidia.com/linux-tegra-rel-19 and download both of these tar-balls:
      - Driver Package: Jetson TK1
      - Sample file system
   b. On the Linux PC, unzip the tar files by running the command:

   ```
   export RELEASE_NAME=Tegra124_Linux_R19.3.0_armhf.tbz2
   sudo tar xpf ${RELEASE_NAME}
   cd Linux_for_Tegra/rootfs/
   sudo tar xpf ../../Tegra_Linux_Sample-Root-Filesystem_R19.3.0_armhf.tbz2
   cd ../
   ```
   Note: It’s important to use sudo to extract these files.

3. In the extracted Linux_for_Tegra folder, edit the jetson-tk1.conf file by uncomment the line (ODMDATA=0x6209C000) for USB2 and comment in the line for USB3 (ODMDATA=0x6009C000).

4. Enter:

   ```
   sudo ./apply_binaries.sh
   ```

5. Connect the TK1 to the Linux machine using the micro-USB cable that comes with the TK1.

6. On the TK1, hold down the Recovery button and press the Reset button once. Wait two seconds and release the Recovery button.

7. Verify that the TK1 is connected to the Linux PC by issuing the lsusb command on the Linux PC and look for a connected USB device called NVIDIA Corp.

8. Copy the Linux_for_Tegra unzipped folder to the Linux machine.

9. In the Linux machine, enter:
sudo ./flash.sh -S 14GiB jetson-tk1 mmcblk0p1

10. Reboot TK1 by pressing the Reset button.

The steps above allow you to stream your USB3 camera at a maximum image size of 2 MB.

To acquire images greater than 2 MB in resolution, add the following to the APPEND line:

```bash
usbcore.usbfs_memory_mb=1000
```

to this file:

`/boot/extlinux/extlinux.conf`

### NVIDIA Jetson TX1 Getting Started

1. Follow NVIDIA’s [Quick Start Guide](#) file to install the 64-bit Linux 4 Tegra R24.1 package.
   
   Note: This step overwrites your existing file system.

2. Install 64-bit FlyCapture2 ARM by downloading the latest [FlyCapture2 64-bit ARM](#) and follow the readme file for installation instructions.

3. Install the g++ compiler in order to install the FlyCapture2 SDK’s C++ samples:

```bash
sudo apt-get update
dsudo apt-get install build-essential
```

The steps above will allow you to stream your USB3 camera at a maximum image size of 2 MB. To acquire images greater than 2 MB in resolution, add the following to the APPEND line:

```bash
usbcore.usbfs_memory_mb=1000
```

to this file:

`/boot/extlinux/extlinux.conf`
NVIDIA Jetson TX2 Getting Started

1. Follow NVIDIA’s Quick Start Guide file to install the 64-bit Linux 4 Tegra R24.1 package.
   
   Note: This step overwrites your existing file system.

2. Install 64-bit FlyCapture2 ARM by downloading the latest FlyCapture2 64-bit ARM and follow the readme file for installation instructions.

The steps above allow you to stream your USB3 camera at a maximum image size of 2 MB. To acquire images greater than 2 MB in resolution, add the following to the APPEND line:

```
usbcod.usbfs_memory_mb=1000
```

...to this file:

```
/boot/extlinux/extlinux.conf
```

NVIDIA DRIVE PX Getting Started

To obtain maximum bandwidth with DRIVE PX:

- USB3 cameras require the latest version of a custom kernel from GitHub, as described below.
- GigE cameras require a USB 3.1-to-Ethernet adapter connected directly to an on-board USB port.

For USB3 cameras:

1. Get the latest kernel from GitHub:

   ```
   # Clone source repo for mainline kernel
   git clone https://github.com/torvalds/linux.git
   
   # Get list of tags to determine appropriate release (I settled on v4.7.0-rc7 for specific technical reasons).
   git tag -l
   
   # Checkout specific tagged revision to local branch.
   # git checkout tags/<tag_name> -b <branch_name>
   git checkout tags/v4.7-rc7 -b nv_custom_branch
   
   2. To acquire images greater than 2 MB in resolution, modify the os_args parameter of the KERNEL_PRIMARY partition in:

   ```
   vibrante-<platform|ver>-foundation/utils/scripts/bootburn/<cfg_file>
   ```

   Add the following to os_args:
3. After the kernel is built, flash the custom kernel onto the board. Because the process depends on the board and kernel version, consult the NVIDIA Vibrante Linux Development guide at /home/nvidia/PDK_Docs/ for specific instructions.

Note: You only want to re-flash the kernel, not the complete file system. To avoid flashing the complete file system, add the "-s" option (skip filesystem) to the bootburn.sh command:

```
"./utils/scripts/bootburn/bootburn.sh -b e2379b00a -s"
```

The sample in the documentation does not use the -s.

For GigE and USB3 cameras:
(Note: Connect the GigE camera to the PC with a USB-to-Ethernet adapter.)

1. Install the g++ compiler in order to install the FlyCapture2 SDK’s C++ samples:

```
sudo apt-get update
sudo apt-get install build-essential
```

2. Install 64-bit FlyCapture2 ARM by downloading the latest [FlyCapture2 64-bit ARM](#) and follow the README file for installation instructions.
Benchmark Results

The following tables benchmark the performance of running selected USB3 or GigE cameras on the following boards: ODROID-XU; Samsung Exynos 5250 Arndale; and NVIDIA Jetson TK1, TX1, TX2, and DRIVE PX.

ODROID-XU

These results are from a console application that continuously captures images using GS3-U3-23S6M. The benchmark using the console application is shown below.

<table>
<thead>
<tr>
<th>GS3-U3-23S6M benchmark results</th>
<th>Requested Frame Rate</th>
<th>Processed Frame Rate</th>
<th>CPU Usage</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1280 x 960 Raw8</td>
<td>200 FPS</td>
<td>200 FPS</td>
<td>22%</td>
<td>234 MB/s</td>
</tr>
<tr>
<td>1792 x 960 Raw8</td>
<td>200 FPS</td>
<td>200 FPS</td>
<td>28%</td>
<td>328 MB/s</td>
</tr>
</tbody>
</table>

Note: At the time of writing, ODROID-XU officially supports up to Linux kernel 3.4.67. Expect the CPU usage on the ODROID-XU board to reduce by 50% when running Linux kernel 3.6 or higher.
Samsung Exynos 5250 Arndale

Using FlyCapture2 to enumerate camera

These are the results from a console application that continuously captures images using GS3-U3-23S6M.

<table>
<thead>
<tr>
<th>Requested Frame Rate</th>
<th>Processed Frame Rate</th>
<th>CPU Usage</th>
<th>Bandwidth</th>
<th>Image Save Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1280 x 960 Raw8</td>
<td>200 FPS</td>
<td>200 FPS</td>
<td>24%</td>
<td>234 MB/s</td>
</tr>
<tr>
<td>1792 x 960 Raw8</td>
<td>200 FPS</td>
<td>200 FPS</td>
<td>28%</td>
<td>328 MB/s</td>
</tr>
<tr>
<td>1920 x 1200 Raw8</td>
<td>162 FPS</td>
<td>162 FPS</td>
<td>29%</td>
<td>356 MB/s</td>
</tr>
</tbody>
</table>

Console application for Arndale

This is a typical solution for an in-field system solution where the embedded board is portable and capable of performing basic processing as well as image/data saving. The benchmark results using the console application are shown below.
GS3-U3-23S6M running at 1280 x 960 resolution, 24% CPU usage

GS3-U3-23S6M running at 1280 x 960 resolution, Raw8 Pixel Format, approx. 200 fps

Note: The speed of image saving is dependent on the hard drive. Typical hard drive write speed is advertised with sequential write speed. When benchmarking hard drives, use a file size similar to the image size.
GS3-U3-23S6M running at 1280 x720 resolution, Raw8 Pixel Format, 24 fps

NVIDIA Jetson TK1

These are the results from a console application that continuously captures images using GS3-U3-23S6M, GS3-PGE-23S6C, and FL3-GE-03S1M. The resulting benchmark using the console application is shown below.

<table>
<thead>
<tr>
<th>GS3-U3-23S6M benchmark results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requested Frame Rate</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>1280 x 960 Raw8</td>
</tr>
<tr>
<td>1920 x 1200 Raw8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GS3-PGE-23S6C benchmark results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requested Frame Rate</td>
</tr>
<tr>
<td>-----------------------</td>
</tr>
<tr>
<td>1920 x 1200 Raw8</td>
</tr>
</tbody>
</table>
FL3-GE-03S1M benchmark results

<table>
<thead>
<tr>
<th>Requested Frame Rate</th>
<th>Processed Frame Rate</th>
<th>CPU Usage</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>640 x 480 Raw8</td>
<td>120 FPS</td>
<td>24%</td>
<td>36 MB/s</td>
</tr>
</tbody>
</table>

**NVIDIA Jetson TX1**

These are the results from a console application that continuously captures images using GS3-U3-23S6M and GS3-PGE-23S6M. The resulting benchmark using the console application is shown below. In our console application, we enabled maximum packet size supported on the Gigabit host controller, which is 9000 bytes.

![Console application for NVIDIA Jetson TX1](image)

**GS3-U3-23S6M benchmark results**

<table>
<thead>
<tr>
<th>Requested Frame Rate</th>
<th>Processed Frame Rate</th>
<th>CPU Usage</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1280 x 960 Raw8</td>
<td>201 FPS</td>
<td>27%</td>
<td>234 MB/s</td>
</tr>
<tr>
<td>1920 x 1200 Raw8</td>
<td>163 FPS</td>
<td>27%</td>
<td>356 MB/s</td>
</tr>
</tbody>
</table>

**GS3-PGE-23S6M benchmark results**

<table>
<thead>
<tr>
<th>Requested Frame Rate</th>
<th>Processed Frame Rate</th>
<th>CPU Usage</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1920 x 1200 Raw8</td>
<td>45 FPS</td>
<td>30%</td>
<td>104 MB/s</td>
</tr>
</tbody>
</table>

**NVIDIA Jetson TX2**

These are the results from a console application that continuously captures images using GS3-U3-23S6M, GS3-PGE-23S6M and BFLY-PGE-23S6M. The resulting benchmark using the console application is shown below. In our console application, for GigE cameras we enabled maximum packet size supported on the Gigabit host controller, which is 9000 bytes.

![Console application for NVIDIA Jetson TX2](image)

**GS3-U3-23S6M benchmark results**

<table>
<thead>
<tr>
<th>Requested Frame Rate</th>
<th>Processed Frame Rate</th>
<th>CPU Usage</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1920 x 1200 Raw8</td>
<td>163 FPS</td>
<td>25%</td>
<td>375 MB/s</td>
</tr>
</tbody>
</table>
### GS3-PGE-23S6M benchmark results

<table>
<thead>
<tr>
<th>Requested Frame Rate</th>
<th>Processed Frame Rate</th>
<th>CPU Usage</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1920 x 1200 Raw8</td>
<td>45 FPS</td>
<td>46 FPS</td>
<td>25%</td>
</tr>
</tbody>
</table>

### BFLY-PGE-23S6M benchmark results

<table>
<thead>
<tr>
<th>Requested Frame Rate</th>
<th>Processed Frame Rate</th>
<th>CPU Usage</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1920 x 1200 Raw8</td>
<td>41 FPS</td>
<td>41 FPS</td>
<td>25%</td>
</tr>
</tbody>
</table>

### NVIDIA DRIVE PX

These are the results from a console application that continuously captures images using BFLY-PGE-23S6M, BFLY-PGE-50SM-C and CM3-U3-50SM. The resulting benchmark using the console application is shown below.

**Console application for NVIDIA DRIVE PX**

### BFLY-PGE-23S6M-C benchmark results

<table>
<thead>
<tr>
<th>Requested Frame Rate</th>
<th>Processed Frame Rate</th>
<th>CPU Usage</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>2448 x 2048 Mono16</td>
<td>18 FPS</td>
<td>18 FPS</td>
<td>80%</td>
</tr>
</tbody>
</table>

### BFLY-PGE-50S5M-C benchmark results

<table>
<thead>
<tr>
<th>Requested Frame Rate</th>
<th>Processed Frame Rate</th>
<th>CPU Usage</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>2448 x 2048 Mono16</td>
<td>9 FPS</td>
<td>9 FPS</td>
<td>80%</td>
</tr>
</tbody>
</table>

### CM3-U3-50S5M-CS benchmark results

<table>
<thead>
<tr>
<th>Requested Frame Rate</th>
<th>Processed Frame Rate</th>
<th>CPU Usage</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1920 x 1200 Mono16</td>
<td>19 FPS</td>
<td>19 FPS</td>
<td>40%</td>
</tr>
</tbody>
</table>
Troubleshooting

Image transfer fails to start when image size is bigger than 2 MB

**Cause:** The USBFS buffer size is too small (16 MB by default). Use the following command to check `usbfs_memory_mb` size:

```
cat /sys/module/usbcore/parameters/usbfs_memory_mb
```

**Solution:** Increase the memory by adding the following to the APPEND line:

```
usbcore.usbfs_memory_mb=1000
```

to this file:

```
/boot/extlinux/extlinux.conf
```

If this method fails to set the memory limit, run the following command:

```
sudo sh -c 'echo 1000 > /sys/module/usbcore/parameters/usbfs_memory_mb'
```

Camera is detected but cannot stream

**Cause:** The USB3 bus power provided by the ARM board is not sufficient to power the camera.

**Solution:** Power the USB3 camera externally using a GPIO cable.

GigE camera frame rates are extremely low on Jetson TK1, TX1, or TX2 board

**Cause:** The socket buffer size by default is too low.

**Solution:** Increase socket buffer size by entering the following in Terminal:

```
sudo sysctl -w net.core.rmem_max=33554432
sudo sysctl -w net.core.wmem_max=33554432
sudo sysctl -w net.core.rmem_default=33554432
sudo sysctl -w net.core.wmem_default=33554432
```
Downloads and support

FLIR endeavors to provide the highest level of technical support possible to our customers. Most support resources can be accessed through the Support section of our website.

The first step in accessing our technical support resources is to obtain a customer login account. This requires a valid name and email address. To apply for a customer login account go to our downloads page.

Customers with a customer login account can access the latest software and firmware for their cameras from our website. We encourage our customers to keep their software and firmware up-to-date by downloading and installing the latest versions.

Finding information

FlyCapture® SDK—The FlyCapture SDK provides API examples and the FlyCap camera evaluation application. Available from our Downloads page.

API Documentation—The installation of the FlyCapture SDK comes with API references for C++, C#, and C code. Available from Start Menu→All Programs→Point Grey FlyCapture2 SDK→Documentation

Product Documentation—The camera’s Getting Started Manual provides information on installing components and software needed to run the camera. The Technical Reference provides information on the camera’s specifications, features and operations, as well as imaging and acquisition controls. They are available from the downloads page.

Knowledge Base—A database of articles and application notes with answers to common questions as well as articles and tutorials about hardware and software systems. Available from our knowledge base.

Learning Center—Our Learning Center contains links to many resources including videos, case studies, popular topics, other application notes, and information on sensor technology.

Contacting technical support

Before contacting technical support, have you:

1. Read the product documentation?
2. Searched the knowledge base?
3. Downloaded and installed the latest version of software and/or firmware?

If you have done all the above and still can’t find an answer to your question, contact our technical support team.