

Embedded boards driver for IDS industrial cameras

Hardware acceleration

IDS provides the uEye Linux Embedded camera driver for different ARM architectures. For the ARMv7 packet, you can choose between a **Hardfloat (hf)** and a **Softfloat (sf)** version.

Floating-Point-Unit

Many current ARM Embedded Boards and devices have an integrated hardware support for floating-point operations (FPU) with the consequence of a higher performance in applications with various operations like image calculation, transformations, exponential or trigonometric calculations with floating point arithmetic. In the ARM environment **VFP** means *Vector Floating Point Architecture*. In the meantime there are three main version:

- **VFPv1** – outdated and no longer used
- **VFPv2** – is an optional extension of the ARMv5 and ARMv6 instructions-set.
- **VFPv3** – is the optional extension from the ARMv7 architecture on.

In contrast, softfloat emulates an FPU in software. Corresponding operations run much slower than with hardware support because to a higher CPU load.

Softfloat or Hardfloat?



The following information is related to the GNU/Linux system that is installed on the embedded board.

Querying the CPU info

If the used CPU features this hardware support, can be requested via the CPU information. The verification will show entries in the **features line of /proc/cpuinfo** or in the VFP support message, logged out while the kernel boots.

```
pi@raspberrypi:~ $ cat /proc/cpuinfo
processor       : 0
model name     : ARMv7 Processor rev 4 (v7l)
BogoMIPS      : 38.40
Features       : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32
lpae evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant    : 0x0
CPU part       : 0xd03
CPU revision   : 4
...
```

What libraries are used?

What libraries are used by basic applications? For this purpose, a standard application like e.g. “ls” can be examined. The command **ldd** shows what libraries are linked.

```
pi@raspberrypi:~ $ ldd /bin/ls
linux-vdso.so.1 (0x7eec3000)
/usr/lib/arm-linux-gnueabi/hf/libarmmem.so (0x76ef7000)
libselinux.so.1 => /lib/arm-linux-gnueabi/hf/libselinux.so.1 (0x76eb7000)
libacl.so.1 => /lib/arm-linux-gnueabi/hf/libacl.so.1 (0x76ea0000)
libc.so.6 => /lib/arm-linux-gnueabi/hf/libc.so.6 (0x76d5f000)
/lib/ld-linux-armhf.so.3 (0x54af6000)
libpcre.so.3 => /lib/arm-linux-gnueabi/hf/libpcre.so.3 (0x76cec000)
```

```
libdl.so.2 => /lib/arm-linux-gnueabi/hf/libdl.so.2 (0x76cd9000)
libattr.so.1 => /lib/arm-linux-gnueabi/hf/libattr.so.1 (0x76cc3000)
libpthread.so.0 => /lib/arm-linux-gnueabi/hf/libpthread.so.0 (0x76c9b000)
```

Here, you see that the **Hardfloat** libraries in the directory **/lib/arm-linux-gnueabi/hf** are used. That indicates a system with hardfloat support.

Detailed information about a library or an application

Using the **readelf** tool shows more detailed information very easy. The presence of the file attribute **'Tag_ABI_VFP_args'** for example shows the usage of the VFP registers.

For the check, choose the standard "C" library **libc.so.6**.

```
pi@raspberrypi:~ $ sudo readelf -A /lib/arm-linux-gnueabi/hf/libc.so.6
Attribute Section: aebi
File Attributes
  Tag_CPU_name: "6"
  Tag_CPU_arch: v6
  Tag_ARM_ISA_use: Yes
  Tag_THUMB_ISA_use: Thumb-1
  Tag_FP_arch: VFPv2
  Tag_ABI_PCS_wchar_t: 4
  Tag_ABI_FP_rounding: Needed
  Tag_ABI_FP_denormal: Needed
  Tag_ABI_FP_exceptions: Needed
  Tag_ABI_FP_number_model: IEEE 754
  Tag_ABI_align_needed: 8-byte
  Tag_ABI_enum_size: int
  Tag_ABI_HardFP_use: SP and DP
  Tag_ABI_VFP_args: VFP registers
Tag_CPU_unaligned_access: v6
```

When the line with **Tag_ABI_VFP_args** is not present, it is a Softfloat system.



If you do not know the directory of the **libc.so.6** file, you can search for it using

```
$> sudo find / -name libc.so.6
```

Software Requirements

Standard C Libraries

According to the uEye SDK version, the Linux system has to support a minimum required version of the GNU C Library (GLIBC, libc.so.6) and the GNU C++ Library (GLIBCXX, libstdc++.so.6). The required versions are described in the **Readme.txt**.

GNU C Library

You can query the supported version of the GLIBC with following command via the Linux command line:

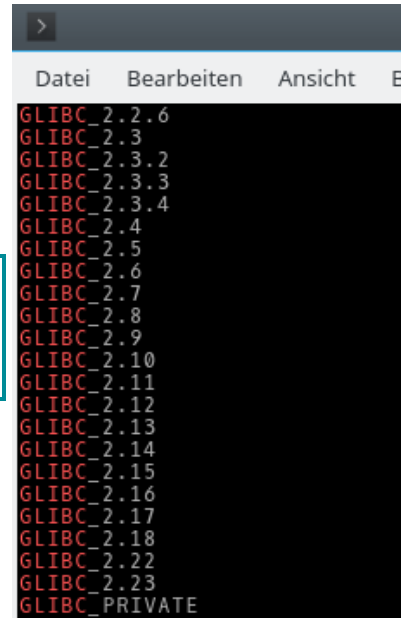
```
$> strings /lib/arm-linux-gnue.../libc.so.6 |
grep GLIBC
```



If you do not know the directory of the **libc.so.6** file, you can search for it using

```
$> sudo find / -name libc.so.6
```

Check if the required version is present in the console output.



```
GLIBC_2.2.6
GLIBC_2.3
GLIBC_2.3.2
GLIBC_2.3.3
GLIBC_2.3.4
GLIBC_2.4
GLIBC_2.5
GLIBC_2.6
GLIBC_2.7
GLIBC_2.8
GLIBC_2.9
GLIBC_2.10
GLIBC_2.11
GLIBC_2.12
GLIBC_2.13
GLIBC_2.14
GLIBC_2.15
GLIBC_2.16
GLIBC_2.17
GLIBC_2.18
GLIBC_2.22
GLIBC_2.23
GLIBC_PRIVATE
```

GNU C++ Library

You can query the supported version of the GLIBCXX with following command via the Linux command line:

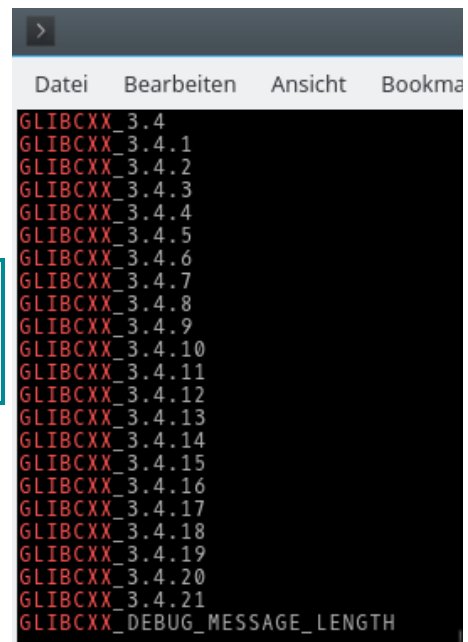
```
$> strings /usr/lib/arm-linux-
.../libstdc++.so.6 | grep GLIBCXX
```



If you do not know the directory of the **libstdc++.so.6** file, you can search for it using

```
$> sudo find / -name libstdc++.so.6
```

Check if the required version is present in the console output.



```
GLIBCXX_3.4
GLIBCXX_3.4.1
GLIBCXX_3.4.2
GLIBCXX_3.4.3
GLIBCXX_3.4.4
GLIBCXX_3.4.5
GLIBCXX_3.4.6
GLIBCXX_3.4.7
GLIBCXX_3.4.8
GLIBCXX_3.4.9
GLIBCXX_3.4.10
GLIBCXX_3.4.11
GLIBCXX_3.4.12
GLIBCXX_3.4.13
GLIBCXX_3.4.14
GLIBCXX_3.4.15
GLIBCXX_3.4.16
GLIBCXX_3.4.17
GLIBCXX_3.4.18
GLIBCXX_3.4.19
GLIBCXX_3.4.20
GLIBCXX_3.4.21
GLIBCXX_DEBUG_MESSAGE_LENGTH
```

Additional Libraries

The following Qt libraries must be installed if you want to use the GUI programs **uEye Demo** or **IDS Camera Manager**:

- libQt3Support.so.4
- libQtGui.so.4
- libQtXml.so.4
- libQtCore.so.4
- libQtNetwork.so.4
- libQtCore.so.4

The following library must be installed if you want to save images as JPEGs or use the special **JPEG** mode of the XS camera:

- libjpeg.so.62 (*Alternatively, you can use the version libjpeg.so.8.*)

The following library must be installed if you want to save images as **PNG**:

- libpng12.so.0

Installing libraries

On Debian- or Ubuntu-based systems, you can install the following packages for example by using **apt-get**:



Important: To run the commands, you must be “root”.

```
$> apt-get install libqt4-qt3support libqtgui4 libqt4-network libqtcore4  
libqt4-xml libqt4-sql libjpeg62 libpng12-0
```

Installation

Installing the SDK



Important: To run the commands, you must be “root”.

1. Copy the TAR archive into your home directory on the target system.
2. Unpack the archive with the **tar** program on the target system:

```
tar xvf uEyeSDK-[version number]-ARM_LINUX_IDS_[setup type].tar -C /
```

With the command parameter “**-C /**” the package will be extracted right into the root folder. So the setup components are copied to the required places in the Linux system.
3. Execute the setup script

```
/usr/local/share/ueye/bin/ueyesdk-setup.sh
```
4. After the installation, start the uEye daemon:
 - a. Via the IDS Camera Manager (root), if you use a graphical user interface.
 - b. Via the command line
USB daemon:

```
/etc/init.d/ueyeusbdrc start
```


ETH daemon:

```
/etc/init.d/ueyeethdrc start
```

Uninstalling the SDK



Important: To run the commands, you must be “root”.

Execute the uninstall script

```
$> /usr/local/share/ueye/bin/ueyesdk-uninstall.sh
```