New Major Release of the **Generic Interface for Cameras**

Dr. Friedrich Dierks, Basler AG
Chief Engineer, Head of SW Development, Basler Components
Chair of the GenICam Standard Group
Overview

- GenICam for the Industry
- GenICam Modules
  - GenApi
  - SFNC
  - GenTL
  - CLProtocol
- New in Version 2.0
- How to Participate
What is GenICam?

GenICam is a Tool...

...to connect cameras
- from different vendors
- with different interfaces
- via different SW libraries

⇒ to your software!
Who is GenICam Member? ➞ 72 Companies in Total
Who is Driving GenICam Actively?

History
- 7 years of intense work
- 19 international meetings
- 10..14 out of 72 members form the core team
- Common code base maintained by the group
- Homework between meetings
- Voting rights are tied to contribution (=homework) yielding in very fast progress

<table>
<thead>
<tr>
<th>Company</th>
<th>Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basler</td>
<td>18</td>
</tr>
<tr>
<td>Stommer</td>
<td>17</td>
</tr>
<tr>
<td>DALSA/Coreco</td>
<td>17</td>
</tr>
<tr>
<td>JAI/Pulnix</td>
<td>16</td>
</tr>
<tr>
<td>Leturon</td>
<td>14</td>
</tr>
<tr>
<td>e2v (ex ATMEL)</td>
<td>13</td>
</tr>
<tr>
<td>MvTec</td>
<td>13</td>
</tr>
<tr>
<td>Pleora</td>
<td>13</td>
</tr>
<tr>
<td>Matrox</td>
<td>10</td>
</tr>
<tr>
<td>NI</td>
<td>9</td>
</tr>
<tr>
<td>Euresys</td>
<td>7</td>
</tr>
<tr>
<td>Toshiba/Teli</td>
<td>4</td>
</tr>
<tr>
<td>IDS</td>
<td>4</td>
</tr>
<tr>
<td>AVT</td>
<td>4</td>
</tr>
<tr>
<td>PhotonFocus</td>
<td>4</td>
</tr>
<tr>
<td>SVS-Vistec</td>
<td>3</td>
</tr>
<tr>
<td>Matrix Vision</td>
<td>3</td>
</tr>
<tr>
<td>Sick</td>
<td>2</td>
</tr>
<tr>
<td>Kappa</td>
<td>2</td>
</tr>
<tr>
<td>Cognex</td>
<td>1</td>
</tr>
<tr>
<td>Mikrotron</td>
<td>1</td>
</tr>
<tr>
<td>PointGrey</td>
<td>1</td>
</tr>
<tr>
<td>Micro Encoder</td>
<td>1</td>
</tr>
<tr>
<td>Lumenera</td>
<td>1</td>
</tr>
<tr>
<td>Impuls Imaging</td>
<td>1</td>
</tr>
<tr>
<td>Sensor To Image</td>
<td>1</td>
</tr>
<tr>
<td>VRMagic</td>
<td>1</td>
</tr>
</tbody>
</table>

Where is GenICam Used Today?

- **GiGE Vision**
  - Part of standard

- **Camera Link**
  - 1st Beta Available
  - Included in standard proposal

- **Adimec CoaxPress**
  - Assumed to be included in standard proposal

- **I394™ IIoC 2.0**
  - Tbd

- **I394™ IIoC 1.3**
  - Some companies

- **USB 3.0**
  - Some companies

- **USB 2.0**
  - Tbd

- Misc
  - ?

- Some companies tbd

Note: The image contains logos and symbols representing various companies and standards.
Overview

- GenICam for the Industry
- GenICam Modules
  - GenApi
  - SFNC
  - GenTL
  - CLProtocol
- New in Version 2.0
- How to Participate
GenICam Configurations

The majority of systems is still Config Only

Camera Vendor

Application

Software Library

GenICam

NIC

Camera

Driver

XML

Grab

Config &

Application

Software Library

GenICam

NIC

Camera

Driver

XML

Grab

Config only
GigE Vision History

First attempt: fixed register layout
- Kick-off meeting June 2003
- Every company tried to get their proprietary register layout standardized
- After one year no conclusion was reached ➔ committee was stuck 😞

Escape Route
- Let every camera have their own register layout
- Define standard features abstractly
- Have a camera description file in XML format with describes how to map the abstract features to the registers

Camera

Camera Description File

Standard interface
- ReadRegister(…)
- WriteRegister(…)
GenICam Modules GenApi and SFNC

**GenApi Module**
- Defines the XML language of the camera description file
- Ideas
  - Supported types: Integer, Float, Enumeration, Bool, String, Command
  - Each type corresponds to an interface with methods like GetValue, SetValue, GetMin, GetMax, GetAccessMode etc.
  - Camera possesses a set of features
  - Each feature has a name, a type and a meaning ➔ abstract
- Example: IInteger::Gain // amplification

**SFNC\(^\d\) Module**
- Defines a set of abstract features forming the ideal camera
- No details, just the name, type and meaning ➔ committee was un-stuck 😊
- List has grow to 220 features in 14 categories like ImageFormatControl, AcquisitionControl, AnalogControl, DigitalIO etc.

**Example**
- SFNC_Camera.chm
  - all standard interfaces
  - all standard features (“ideal camera”)

\(^\d\) SFNC = Standard Feature Naming Convention
History Again & Business Dynamics

GigE Vision ➔ GenICam
- GenICam can be used not only for GigE Vision but also for all register based transport layer like, e.g. 1394 IIDC, USB
- GenICam was made a separate standard
- GigE Vision refers to GenICam

What makes a Good Standard?
- Balance between interoperability and room for competition
- Custom Features in GenICam
  - XML language describes custom and standard features alike
  - Only if vendors use SFNC plug&play is achieved
- Business Dynamics
  - Camera vendors like XML language
  - Software vendors insist on SFNC usage

CameraLink ➔ not enough interoperability
- No plug&play due to extremely poor definition of configuration interface (“serial port”)
- Result
  - Every camera comes with a stand-alone configuration tool
  - Rare and restricted API only

1394 IIDC ➔ not enough room for competition
- Fixed Register Layout yields plug&play
- Problems
  - No competition for standard features.
  - It’s quite impossible to overcome restrictions of standard features, e.g. 12 bit for exposure time only
  - Extension possible but no standard way to access custom features (void* only)
GenICam Module GenTL

GenTL Module

- Defines an object model and abstract C++ interfaces for grabbing images
- Use Cases covered
  - Enumerating transport layers (GEV, 1394, CL, USB, …)
  - Enumerating Devices
  - Configuring Devices using GenApi
  - Opening one or more video streams
  - Buffer handling
- GenTL can handle any number of devices, drivers, and interface technologies with one common API
- It just came a little late for GigE Vision…
Supporting CameraLink

CameraLink Configuration Interface
- CameraLink provides a standard mechanism to access a serial port on the camera.
- Every frame grabber comes with a port driver DLL named CLSerXXX.dll whose C interface is standardized.
- The CL standard committee provides a freeware CLAllSerial.dll which covers the following use cases:
  - Enumerates all CLSerXXX DLLs
  - Enumerates all frame grabber boards per DLL
  - Enumerates all camera port per frame grabber board
  - Allows to send and receive packages per camera
- Problem: No registers…
If the camera is **register based** the CLProtocol.dll is just running a frame protocol.

If the camera is **command based** the CLProtocol.dll needs to emulate a pseudo register space.

```c
Camera.WriteRegister( address, data, len )
{
    switch( address )
    {
        case adrGain: // 0xffff0001
            int Gain = extractIntData( data, len );
            // set gain via serial port of camera
            Camera.SetGain( Gain );
            break;
        ...
    }
}
```
GenICam Module CLProtocol (2/2)

**CLProtocol Module**
- Defines a camera driver CLProtocol.DLL which implements a (virtual) register interface for a camera
- Camera manufacturer provides setup to install the driver besides the frame grabber software
- Use Cases covered by CCLPort wrapper class:
  - Enumerating serial ports
  - Enumerating possible driver DLL bindings
  - Dynamically loads CLProtocol.dll
  - Retrieves XML description from camera of file
  - Provides IPort interface required by GenApi
How Things Worked Out

Original Assumption
- Customers use the native GenICam API
- XML file contains a ~1:1 mapping of registers to features

What happened in Reality
- Library vendors used GenICam as engine under the hood
- Customers got the functionality of GenICam but through the libraries’ native API
- XML file is used to map legacy registers to SFNC features
- XML language v1.0 not powerful enough for all use cases

*) some use GenICam natively; many have a back-door
Overview

- GenICam for the Industry
- GenICam Modules
  - GenApi
  - SFNC
  - GenTL
  - CLProtocol
- New in Version 2.0
- How to Participate
GenICam Reference Implementation

GenICam Standard
- Text of the standard modules
- XML schema file (GenApi)
- C++ Header (GenTL)

GenICam Reference Implementation
- Maintained by standard committee
- Can be used free of charge
- Not part of the standard
- Technical data
  - Written in C++
  - Supports Win32 / Win64 with VisualStudio 7.1 / 8.0 / 9.9
  - Supports Linux32 / Linux64 with Suse 10.0 (gcc>=4.0, glibc>=2.3.5)
  - Strict focus on quality

⇒ IMPORTANT : Just the engine, no driver!

⇒ v2.0 adds 50% more code
## New Features in v2.0

### General Changes
- Code refactoring (CMake)
- Speed improvement (pre-processing)
- Parallel handling of old and new schema version

### Hidden Features
- Empowered XML Language
- New use cases (Replicator)
- Better maintainability (Multiplexer)
- Reduced complexity (PolyPointers)
- Better Debugging Tools (logging)
- XML Injection

### Customer Features
- Supports self-clearing values
- Better formatting (float, IP, …)
- Supports error flags in the camera
- DocuURL
- Float now has an increment
- Float Aliases
  - v1.0 had multiple feature names were the committee could not agree on one type, e.g. GainRaw (Int) and GainAbs (Float)
  - v2.0 every feature is converted to Float but there are means to access the native implementation (Gain->GetIntAlias)
Status and Roadmap

GenICam v2.0
- Released
- Fully backward compatible to v1.0 cameras
- Easy migration from v1.0 to v2.0
- Rollout of products expected H1 / 2010

GenICam v2.1
- Adds Camera Link support
  - New CLProtocol module
  - CL specific features in SFNC
- Beta available
- Release expected e/o 2009

What comes next?
- Improving documentation & adding tutorials
- Supporting more compilers / platforms
Overview

- GenICam for the Industry
- GenICam Modules
  - GenApi
  - SFNC
  - GenTL
  - CLProtocol
- New in Version 2.0
- How to Participate
For Customers

Sorry, but...

- GenICam is not intended to be directly used by end customers
- The reference implementation does not(!) contain a free GigE Vision driver
- The code downloadable from www.GenICam.org installs the GenICam engine only and is intended for vendors who do not want to become GenICam member

Instead, please...

- Buy GenICam aware cameras and software only
- Make sure Cameras follow the SNFC
- Make sure software libraries hand out the full functionality of GenICam

Look out for the logos
For Camera and Software Vendors

- GenICam is hosted by the European Machine Vision Association (EMVA)
- You can become member
  - free of charge
  - without being EMVA member
- Membership grants you
  - access to the source code
  - access to the mailing list
  - access to the wiki and the archive
- Membership allows you to contribute
  - contribution means homework
  - only contributing members can vote
- Register Today at www.genicam.org

Contributing Members
(currently 8 companies)

Active Members
(another 6 companies)

Passive Members
(the rest of 58 companies)
Thank you for your attention!

Contact me ➔ friedrich.dierks@baslerweb.com
Get information ➔ www.genicam.org

Visit the International Machine Vision Standards booth for a CLProtocol Live Demo