



Softwareschnittstellen für die Bildverarbeitung – GenICam, WDM, ActiveX u. a. im Vergleich

Whitepaper

© 2009 IDS Imaging Development Systems GmbH. Alle Rechte vorbehalten.



IDS Imaging Development Systems GmbH
Dimbacher Str. 6-8
74182 Obersulm
Tel. +49-(0)7134-96196-0
Fax +49-(0)7134-96196-99

sales@ids-imaging.de
www.ids-imaging.de

Inhalt

| | | |
|-----|---|---|
| 1 | Einleitung | 2 |
| 2 | Softwareschnittstellen für Kameras | 2 |
| 2.1 | TWAIN | 2 |
| 2.2 | DirectShow | 3 |
| 2.3 | ActiveX-Komponenten | 3 |
| 2.4 | Schnittstellen für Bildverarbeitungs-Bibliotheken | 4 |
| 2.5 | GenICam – Generische Kamera-API | 4 |
| 2.6 | Herstellerspezifische Kamera-APIs | 7 |
| | Fachbegriffe auf einen Blick | 8 |
| 3 | Quellen und weiterführende Links | 9 |

1 Einleitung

Für die Einbindung von digitalen Industriekameras in verschiedene Anwendungen bietet sich unter den gängigen Software-Schnittstellen für jede Anforderung ein passender Ansatz. Doch eine Vielzahl an Funktionen und Fachbegriffen machen es schwierig, die richtige Schnittstelle auszuwählen. Dieses Whitepaper stellt etablierte Technologien dem neuen GenICam-Standard gegenüber, erklärt Aufbau und Einsatzmöglichkeiten.

2 Softwareschnittstellen für Kameras

2.1 TWAIN

TWAIN ist ein offenes, kostenloses Software-Protokoll, welches von verschiedenen PC-Herstellern zum Bildeinzug von Scannern und Kameras entwickelt wurde. Die Schnittstelle wird von Windows- und Apple-Systemen direkt unterstützt – lediglich Kameratreiber müssen noch installiert werden – und ist daher sehr weit verbreitet. Der Funktionsumfang ist auf reinen Einzelbildeinzug ausgelegt, wichtige Kameraparameter können in der Regel über einen Dialog angepasst werden. Durch die nahtlose Integration in fast alle Grafik- und Office-Anwendungen eignet sich TWAIN zum gelegentlichen Bildeinzug zu Dokumentationszwecken. Zugriff auf tiefer gehende Funktionen ist nicht möglich, in punkto Geschwindigkeit gehört TWAIN zu den langsamen Schnittstellen.

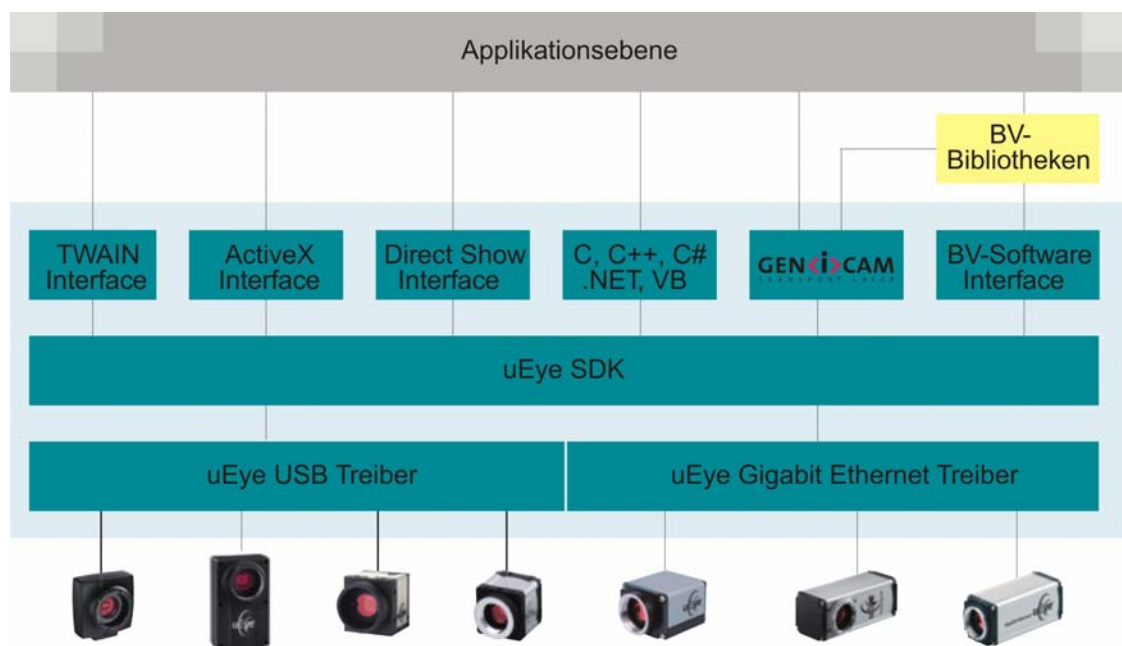


Abbildung 1: Gängige Software-Schnittstellen am Beispiel des uEye® SDK

2.2 DirectShow

Mit dem aktuell in Version 10 erhältlichen DirectX-Paket stellt Microsoft unter Windows allgemeine Multimedia-Schnittstellen für Audio, Video und Grafik zur Verfügung. Mit dem darin enthaltenen DirectShow-Interface lassen sich in vielen grafischen Anwendungen Video-Streams von angeschlossenen Kameras einziehen, ohne zuvor herstellereigene Software-Schnittstellen einzubinden.

Der Nachfolger des Standards *Video for Windows* setzt auf Microsofts COM-Technologie auf und bringt auch eine generische (d. h. hersteller- und technologieunabhängige) Programmierschnittstelle mit. Da die Funktionalität von DirectShow primär für Multimedia-Geräte ausgelegt ist, müssen Funktionen aus dem Bereich der industriellen Bildverarbeitung oft herstellerspezifisch hinzugefügt werden, womit der universale Einsatz eingeschränkt wird.

Durch die zum einen sehr komplexe Programmierung der DirectShow-Schnittstelle, zum anderen aufgrund der Beschränkung hinsichtlich des Funktionsumfangs verwendeter Kameras wird DirectShow eher im Bereich der Videoaufzeichnung eingesetzt. Für hohe Anforderungen im Bereich der industriellen Bildverarbeitung ist diese Leistung in der Regel nicht ausreichend.

2.3 ActiveX-Komponenten

ActiveX ist eine Schnittstelle für sogenannte aktive Inhalte, die ebenfalls auf der Microsoft COM-Technologie basiert. In einer ActiveX-Komponente können die Befehle der Kamera-API gekapselt und dem Anwender Prozesse wie die Speicherverwaltung abgenommen werden. Wichtige Funktionen lassen sich über eine fertige grafische Benutzeroberfläche zur Verfügung stellen.

Dies bietet den Vorteil, dass innerhalb von Minuten eine Kamera-Anwendung zum Laufen gebracht werden kann, da lediglich die ActiveX-Komponente integriert werden muss. Sofort können Bilder angezeigt und die meisten Kameraparameter eingestellt werden – sogar im Webbrowser.

Durch die Kapselung ist eine individuelle Erweiterung des ActiveX-Controls allerdings nicht vorgesehen, auch tiefer gehende Funktionen wie Speicherzugriffe sind nicht möglich. Daher bietet sich die Verwendung der ActiveX-Kameraschnittstelle dann an, wenn eine schnelle Viewer-Anwendung auf einem reinen Windows-System erstellt werden soll.

2.4 Schnittstellen für Bildverarbeitungs-Bibliotheken

Wird eine Kamera zusammen mit Bildverarbeitungs-Applikationen oder Bibliotheken von Drittanbietern eingesetzt, erfolgt die Einbindung zwingend über eine Schnittstelle des Softwareanbieters. Diese hat die Aufgabe, Befehlsaufrufe der Bibliothek in Kommandos der kameraeigenen API zu „übersetzen“. Einige Funktionen wie die Speicherverwaltung sind darin bereits gekapselt und müssen nicht vom Anwender programmiert werden. Eine solche Schnittstelle bringen beispielsweise Bildverarbeitungs-Bibliotheken wie *HALCON*, *NeuroCheck* oder *Common Vision Blox* mit.

2.5 GenICam – Generische Kamera-API

Der Ruf nach einem einheitlichen Standard für alle Arten von Bildverarbeitungskameras konnte in der Vergangenheit noch nicht zufriedenstellend beantwortet werden. DirectShow zum Beispiel unterstützt zwar Geräte vieler Hersteller, ist aber nicht auf industrielle Anforderungen ausgelegt (s. o.). Industrielle Kamerastandards wie DCAM/IIDC für FireWire, CameraLink oder GigE Vision wiederum unterstützen jeweils nur eine Hardware-Technologie.

Mit einem neuen Standard soll dieser Bedarf nun ein für allemal gedeckt werden: *GenICam* bedeutet *Generic Interface for Cameras* und bietet ein einheitliches, technologieübergreifendes und herstellerunabhängiges Verfahren für die Steuerung und den Bildeinzug von Kameras – und prinzipiell sogar analogen Framegrabbern. Die *European Machine Vision Association (EMVA)* leitet seit 2005 ein Konsortium mehrerer Kamera- und Software-Hersteller, das sich mit der Definition dieses Standards befasst; die Verabschiedung der ersten offiziellen Version erfolgte 2008.

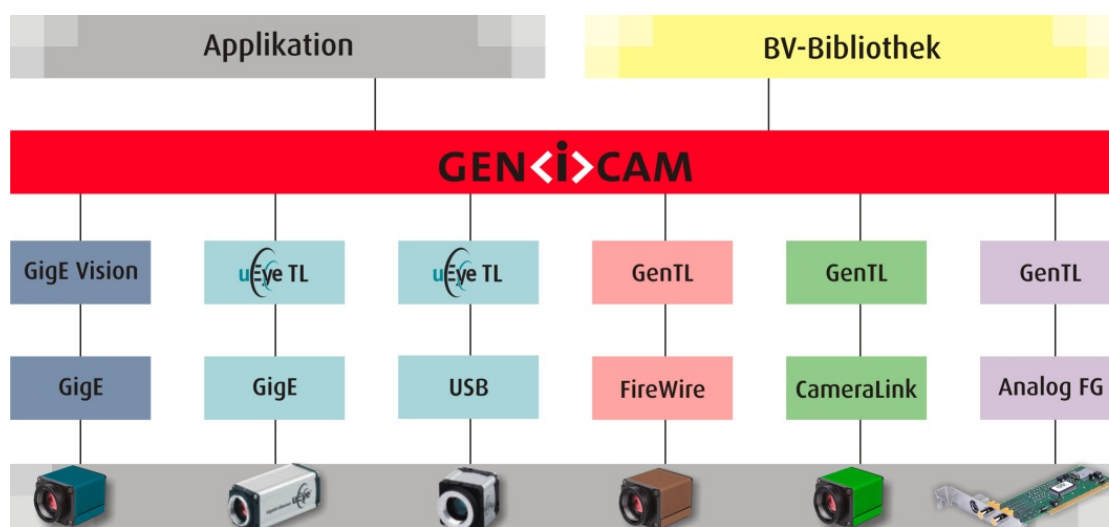


Abbildung 2: Verwendung von GenICam als Kameraschnittstelle (schematisch)

Wie bringt nun GenICam die verschiedenen Technologien unter einer einheitlichen Schicht zusammen? Das Prinzip ist recht einfach: Zu jeder GenICam-kompatiblen Kamerafamilie liefert der Hersteller neben einem kompatiblen Treiber eine Beschreibungsdatei im XML-Format mit. Diese Datei, die man sich bildlich als maschinenlesbares Kamera-Handbuch vorstellen kann, benennt die von jedem Modell unterstützten Funktionen sowie die dafür zu setzenden Kameraregister (*Abbildung 3* zeigt einen Auszug aus einer solchen Beschreibungsdatei). Die definierten Funktionen sind klassifiziert in den Stufen *verpflichtend*, *empfohlen* und *freiwillig*. Eine GenICam-Konvention zur Benennung der Funktionen bietet dabei die Grundlage für problemlose Austauschbarkeit von Hardware verschiedener Hersteller, wobei die Einhaltung der Konvention nicht verpflichtend ist.

```
<Integer Name="Width" NameSpace="Standard">
  <ToolTip>Width of the image (px)</ToolTip>
  <Visibility>Beginner</Visibility>
  <pValue>WidthReg</pValue>
  <pMin>WidthMinReg</pMin>
  <pMax>WidthMaxReg</pMax>
  <pInc>WidthIncReg</pInc>
  <Representation>Linear</Representation>
</Integer>
<IntReg Name="WidthReg">
  <Visibility>Guru</Visibility>
  <pIsLocked>DevTLLocked</pIsLocked>
  <Address>0x118</Address>
  <Length>4</Length>
  <AccessMode>RW</AccessMode>
  <pPort>Device</pPort>
  <Sign>Unsigned</Sign>
</IntReg>
```



Abbildung 3: Auszug aus einer XML-Datei mit Kamerabeschreibungen (hier beispielhaft Funktions- und Registerbeschreibung der Funktion Bildbreite).

Das Kernstück des Standards ist die *GenAPI* genannte API, über die eine Endanwendung auf die Kamera zugreifen kann. Diese API sucht nun nach verfügbaren Kameras mit kompatiblen Treibern und liest die zugehörigen Beschreibungsdateien ein. Mittels einer daraus erstellten Featurebeschreibung kann nun für jede angeschlossene Kamera zuerst geprüft werden, ob bestimmte Funktionen überhaupt unterstützt werden. Nur wenn dies der Fall ist,

wird die Funktion gesetzt – Software, die nach diesem Prinzip arbeitet, ist demnach komplett unabhängig von Kameramodell und -technologie.

Zuletzt muss – vom Hersteller der Hardware oder der Software – ein sogenanntes Transport Layer bereitgestellt werden. Dieses stellt der GenAPI die Beschreibungsdatei zur Verfügung und übersetzt Befehle der GenAPI für den Kameratreiber. Erst das Transport Layer macht letztlich die GenAPI-Software unabhängig von proprietären Kameratreibern und unterschiedlichen Schnittstellen wie USB, GigE, FireWire oder CameraLink. Das Transport Layer beinhaltet fünf Module (*System, Interface, Device, Stream* und *Buffer*), die beim Einbinden eines Transport Layers der Reihe nach initialisiert werden müssen. Diese Vorgehensweise ist aufgrund des hohen Abstraktionsgrads der GenCam-Schnittstelle erforderlich.¹

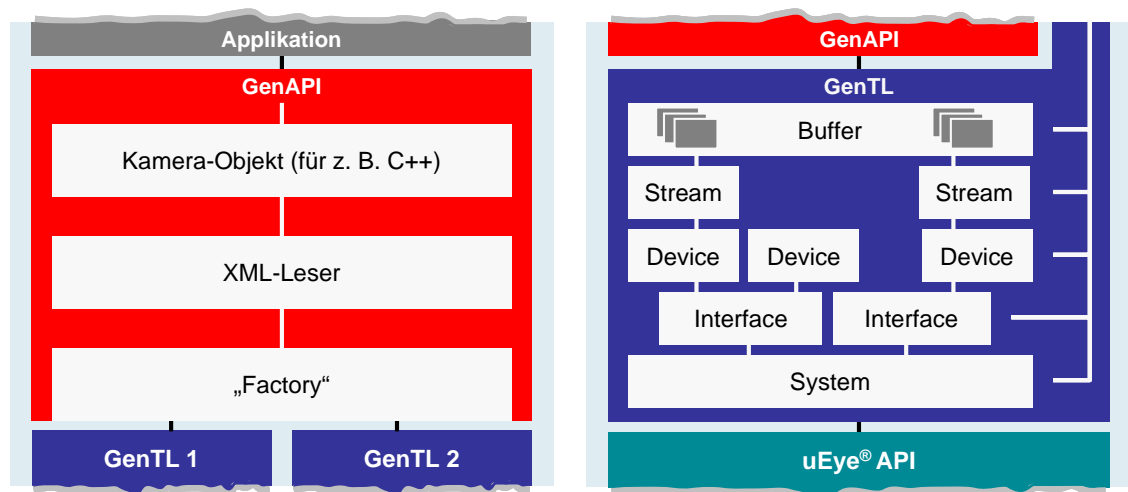


Abbildung 4: Prinzipieller Aufbau der GenCam-API (links) und des Genicam-TL (rechts)

Da das GenICam Transport Layer in C geschrieben ist, lässt es sich in verschiedenen anderen Sprachen und Betriebssystemen einbinden. In der Programmierung mit C++ etwa sähe die Ansteuerung einer GenICam-kompatiblen Kamera folgendermaßen aus, hier am Beispiel der Funktion „Hardware-Gain setzen“:

```
if( IsAvailable(Camera.Gain) )
    Camera.Gain = 42;
```

Es wird also zunächst abgefragt, ob die geöffnete Kamera die Funktion Gain unterstützt. Wenn ja, kann der Wert (hier 42) zugewiesen werden – dieser Code ist also generisch, sprich herstellerunabhängig. Das GenICam Transport Layer übersetzt diesen Befehl nun in ein Kommando, das der Kameratreiber versteht.

¹ Für weitere Informationen zur Programmierung mit GenICam siehe auch: IDS GmbH, „uEye GenICam-Programmierhandbuch“, 2009

Dieser Ansatz macht es vor allem Anbietern von Bibliotheken leicht, Kameras unterschiedlicher Hersteller und mit verschiedenen Hardwareschnittstellen zu integrieren. Es ist daher zu erwarten, dass in Kürze alle großen Anbieter von BV-Bibliotheken auf den GenICam-Standard einschwenken werden.

Doch auch für Systemintegratoren und andere Firmen, die umfangreiche Bildverarbeitungs-Software erstellen, bietet GenICam einen deutlichen Nutzen. Neben der Unabhängigkeit vom Interface und Hersteller der Kamera lassen sich mit GenICam aufgrund des sehr strukturierten Aufbaus auch komplexe Applikationen sehr effizient erstellen. Neue Kamerafunktionen können nach einem Update der Treiber und der XML-Datei sehr einfach integriert werden, da sie direkt aus der XML-Datei ausgelesen werden. Auch wenn Kameras von nur einem Hersteller, aber mit unterschiedlichem Funktionsumfang verwendet werden, erleichtert GenICam die Abstimmung der Anwendung auf die Verfügbarkeit bestimmter Kamerafunktionen. Durch die breite Unterstützung von Softwarelieferanten und Kameraherstellern ist gewährleistet, dass der Standard zukunftsfähig bleibt und weitergepflegt wird.

Noch zwei weitere Fachbegriffe sind an dieser Stelle zu nennen: Als *GenICam Producer* wird ein Hersteller bezeichnet, der zu seiner Kamera ein *GenICam Transport Layer* liefert; als Consumer wird eine Anwendung (z. B. eine Bibliothek) bezeichnet, die GenICam-kompatible Geräte ansteuern kann.

Eine beispielhafte Referenz-Implementierung der GenAPI wird u. a. auf der Webseite des GenICam-Komitees zur Verfügung gestellt; für Mitglieder ist auch der Quellcode verfügbar.

| GEN <i>i</i> CAM | |
|------------------|------------------|
| VERSION | XML |
| 1.0 | Producer |
| GenTL | CAMERA INTERFACE |
| Producer | USB GigE |

Abbildung 5: „Nutrition Facts“-Tabelle, die für jedes GenICam-kompatible Produkt die unterstützten Technologien angibt

2.6 Herstellerspezifische Kamera-APIs

Die meisten Kamerahersteller bieten auch eine eigene Funktionsbibliothek zur Entwicklung von Software-Anwendungen an, welche Funktionen des Treibers als Programmier-Befehle zur Verfügung stellt. Dieser Weg erfordert in der Regel zwar Grundkenntnisse der Programmierung, bietet aber die höchste Flexibilität, Leistung und Funktionalität, da man direkt auf die Hardwarefunktionen zugreifen kann. Durch diese ideale Anpassung für die Kamerahardware ist es nicht möglich, Modelle anderer Hersteller mit der gleichen Software anzusprechen.

Aufbau und Umfang der API variieren je nach Hersteller, ebenso die Zahl der unterstützten Sprachen. Das SDK ist nicht bei allen Herstellern im Lieferumfang der Kamera enthalten sondern muss teilweise zusätzlich gekauft werden.

Die IDS Imaging Development Systems GmbH bietet für alle Kameras der uEye®-Familie kostenlos ein umfangreiches SDK an, das alle der aufgeführten Schnittstellen unterstützt. Die IDS GmbH ist aktives Mitglied des GenlCam-Konsortiums, so dass diese Schnittstelle auch für die uEye® erhältlich ist. Mit der uEye®-API bietet IDS eine C-Programmiersbibliothek mit über 140 Funktionen für die Steuerung der Kamera und des Bildeinzugs. Die API kann unter Windows und Linux in verschiedenen Programmiersprachen eingebunden werden und vereinigt unter einer Schnittstelle alle uEye®-Modelle, angefangen beim günstigen Modell mit VGA-Auflösung und USB-Schnittstelle bis hin zur leistungsfähigen Gigabit Ethernet Kamera mit 10 MPixel-Farbsensor. Über 20 Demoprogramme mit Quellcode in C, C++, C# und VB erleichtern die schnelle Erstellung eigener Anwendungen. Mit dem uEye®-Demo liegt ein umfangreicher Viewer zum sofortigen Arbeiten mit der Kamera vor, der alle wichtigen Funktionen bereitstellt. Weitere Informationen zu IDS und den uEye®-Kameras finden Sie im Internet unter www.ids-imaging.de.

Fachbegriffe auf einen Blick

ActiveX: Software-Komponente für aktive Inhalte auf Basis der Microsoft COM-Technologie, wird von Kameraherstellern zur Kapselung von API-Funktionen verwendet.

API (Application Programming Interface): Funktionsbibliothek (z. B. als DLL-Datei) zur Entwicklung von Software-Anwendungen. Stellt bei Kameras die Funktionen des Treibers als Programmier-Befehle zur Verfügung.

DirectShow: Komponente des Microsoft DirectX-Pakets zur Einbindung von Video- und Audio-Geräten.

GenlCam (Generic Interface for Cameras): Neuer Schnittstellen-Standard, der vom europäischen Dachverband der Bildverarbeiter EMVA gepflegt wird und eine allgemeine API speziell für Bildverarbeitungsgeräte bietet.

SDK (Software Development Kit): Üblicherweise vom Kamerahersteller mitgeliefertes Paket, welches die unterstützten Interfaces und Programmiersbibliotheken zur Erstellung eigener Anwendungen enthält.

TWAIN (Transmit Windows Advanced Interface): Schnittstelle zur Einbindung von Scannern und Kameras.

WDM (Windows Driver Model): Von Windows entwickeltes Treiber-Modell, auf dem u. a. das DirectShow-Interface basiert. Oft fälschlicherweise als Synonym für DirectShow verwendet.

3 Quellen und weiterführende Links

- Spezifikation des GenICam-Standards: <http://www.genicam.org/>
- uEye[®]-Softwarepaket (enthält u. a. Dokumentation des GenICam-Interface für uEye[®]-Kameras): <http://www.ids-imaging.de/ueyetreiber>