



Usergroup

ibaVision



Daniel Gauss
Produktmanager



Fragen / Diskussionsbeiträge

Bitte Mikrofon freigeben und direkt im Plenum Ihre Frage stellen.



Oder nutzen Sie die Chat Funktion. Ihre Frage wird dann im Anschluss an den Vortrag beantwortet.



Kamera

Wir freuen uns über jedes Gesicht, welches wir sehen.



Aufzeichnung

Die Session wird nicht aufgezeichnet.

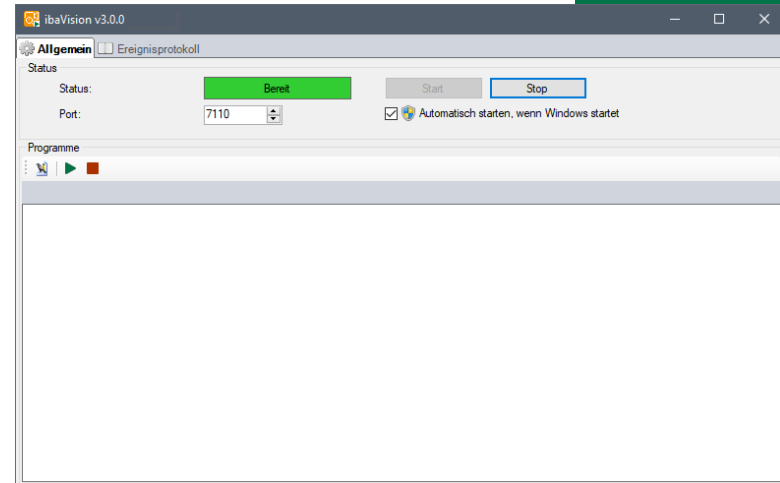


Präsentation

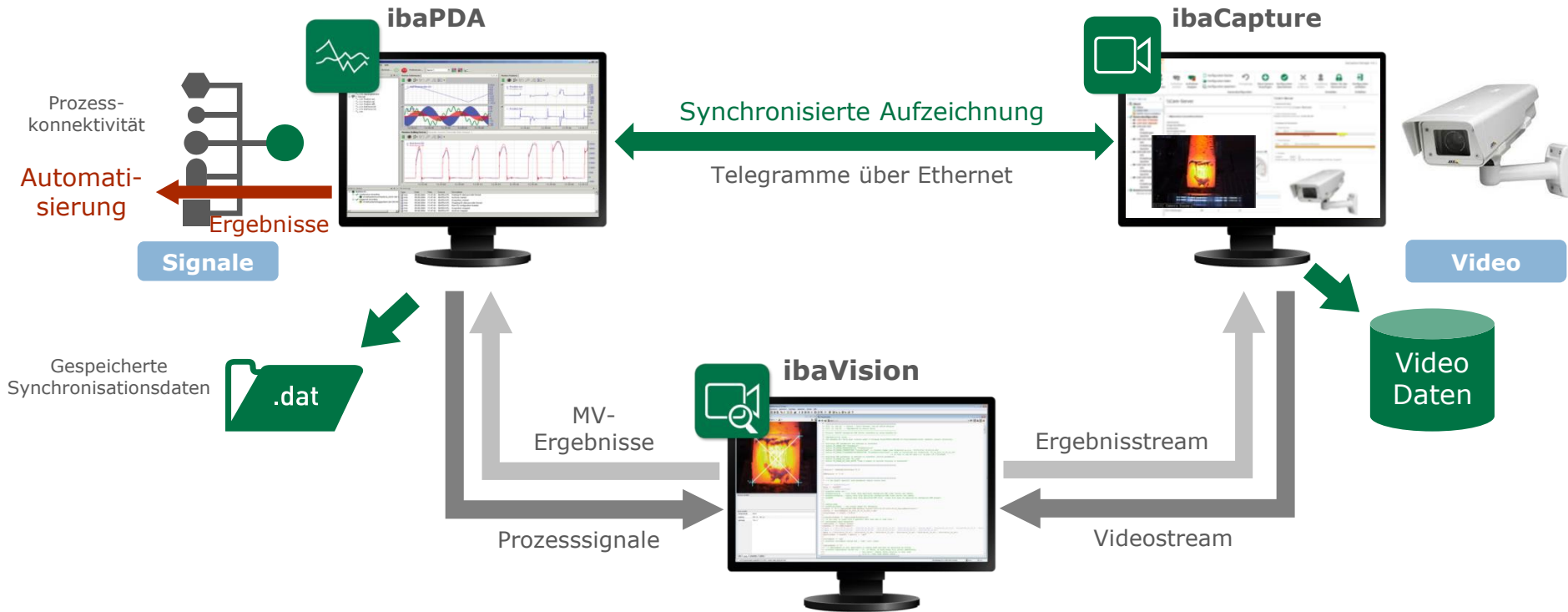
Im Anschluss an den digitalen iba-Tag stehen die Präsentationen der Usergroups und der Vortrag „Produktneuheiten“ auf <https://www.iba-ag.com/de/iba-tag> zur Verfügung.

Sie werden darüber per E-Mail informiert.

- Übersicht
- Neues Plugin-Konzept
- Ziele und Möglichkeiten
- Beispiel
- Fragen

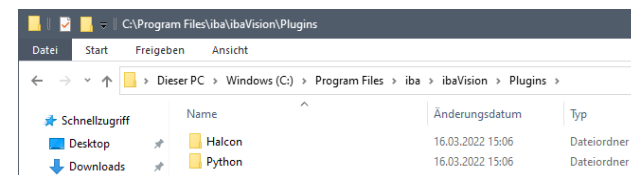


Prozesse optimieren, überwachen und steuern



Plugin-Konzept

- ibaVision bisher nur mit HALCON nutzbar
 - Hilfreicher Support, benutzerfreundliches SDK, gute Performance
 - Einstiegshürden durch Zusatzkosten, evtl. Lernkurve
- Ab ibaVision v3: HALCON als Standard-Plugin
 - ➔ Gleiche Funktionalität wie zuvor
- Neues Standard-Plugin: Python
 - ➔ Zugriff auf verfügbare Erweiterungen und Bibliotheken, z.B.
 - OpenCV
 - TensorFlow
 - Viele mehr...
- Individuell Erstellung von Plugins mit .Net
 - Beispielprojekt im Installationsverzeichnis



- Ausbau von ibaVision zu flexibler Bildverarbeitungs-Plattform
- Offene Plugin-Schnittstelle

- Werkzeug für Entwickler
- Einfache Integration von Bilddaten und Bildverarbeitung ins iba-System

→ Für jedes Projekt kann das passende Werkzeug genutzt werden

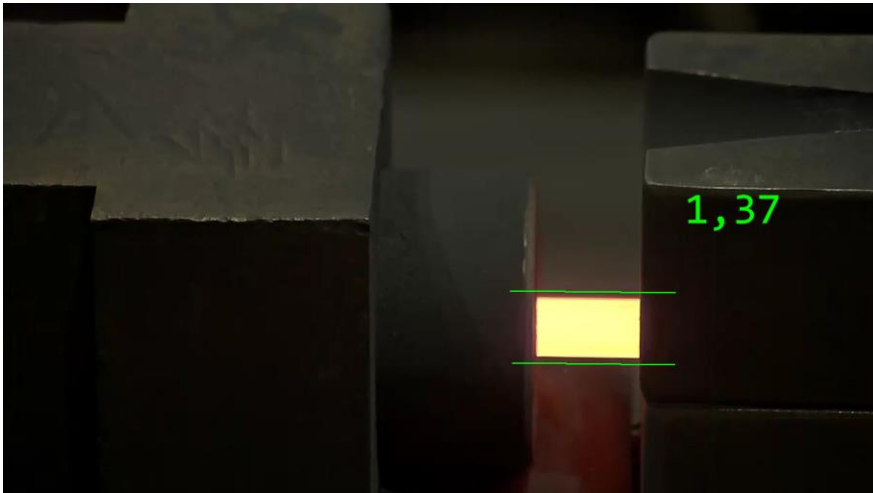


LOW-END

Projektaufwand

HIGH-END

- Einfache Informationen/Daten in Videos einblenden
 - ibaPDA als Datenquelle



- Hersteller-SDKs für Kameras nutzen
 - Anbindung bisher nicht unterstützter Kameras an ibaCapture
 - Lesen von Bilddaten aus der Vergangenheit → Erstellen des Videostreams mit angepasstem Zeitstempel in ibaCapture



Live Demo

Beispiel: Python-Skript für ibaVision



- Einfache Demo der iba-Schnittstellen
- Nutzung von OpenCV

```
Rotamagpy X
PYTHON > Rotamagpy > Rotamagpy 1
1 import numpy as np
2
3 import cv2
4
5 ibaVis(interfaceVersion = "1.0.0")
6
7 ProcedureDescriptions = [
8     ["Description": "Init", "ExecuteInit", "Initialize some values",
9      "InputsControl": {
10       (0, "Image width", "Image width"),
11       (1, "Image height", "Image height")
12      }
13    ],
14    ["Description": "Main", "ExecuteMain", "Rotates the image by a specified angle and return the rotated image and the average brightness (range 0 to 255) as outputs.",
15     "InputsControl": {
16      (0, "Input rotation angle", "The rotation angle"),
17      (1, "Input digital signal", "Digital signal for testing")
18     },
19     "InputsIcomic": { (0, "Input image", "The unrotated input image"),
20                       (0, "Image brightness int", "The int average brightness of the image"),
21                       (1, "Image brightness float", "The float average brightness of the image"),
22                       (2, "Output digital signal", "Digital signal value from input")
23     },
24     "OutputsIcomic": { (0, "Output image", "The rotated output image"),
25                       }
26   ]
27
28 # Inputs
29 InputsIcomic = []
30 InputsIcomic["Main"].append((128, 128, 3, np.ones((128,128,3)), dtype="uint8")) # Add a dummy 128x128 RGB image that allows the script to run standalone.
31
32 InputsControl = {}
33 InputsControl["Init"] = []
34 InputsControl["Init"].append(648)
35 InputsControl["Init"].append(488)
36 InputsControl["Main"] = []
37 InputsControl["Main"].append(8) # Add a dummy control value that allows the script to run standalone.
38 InputsControl["Main"].append(false) # Add a dummy control value that allows the script to run standalone.
39
40 # Outputs
41 OutputsIcomic = []
42 OutputsIcomic["Main"] = []
43 OutputsIcomic["Main"].append((128, 128, 3, np.ones((128,128,3)), dtype="uint8")) # Add a dummy 128x128 RGB image that allows the script to run standalone.
44
45 OutputsControl = {}
46 OutputsControl["Main"] = []
47 OutputsControl["Main"].append(8) # Add a dummy control value that allows the script to run standalone.
48 OutputsControl["Main"].append(0.8) # Add a dummy control value that allows the script to run standalone.
49 OutputsControl["Main"].append(true) # Add a dummy control value that allows the script to run standalone.
50
51 def ExecuteInit():
52     init_width = InputsControl["Init"][0]
53     init_height = InputsControl["Init"][1]
54
55
56
```





v5.1.6 – Juni 2022

- Unterstützung für ibaVision v3.0

v5.2 – August 2022

- Video beim Export re-encodieren (z.B. Zeitstempel einprägen, Export mit Overlay-Bild, Auflösung reduzieren)



v3.0 – Juni 2022

- Einführung der Plugin-Architektur
- Plugins können selbst implementiert werden
- Plugins im Lieferumfang: HALCON, Python
- Bidirektionale Text-Kommunikation mit ibaPDA
 - Benötigt ibaPDA v8.0.0 und ibaCapture v5.1.6 oder neuer

Zeit für Ihre Fragen