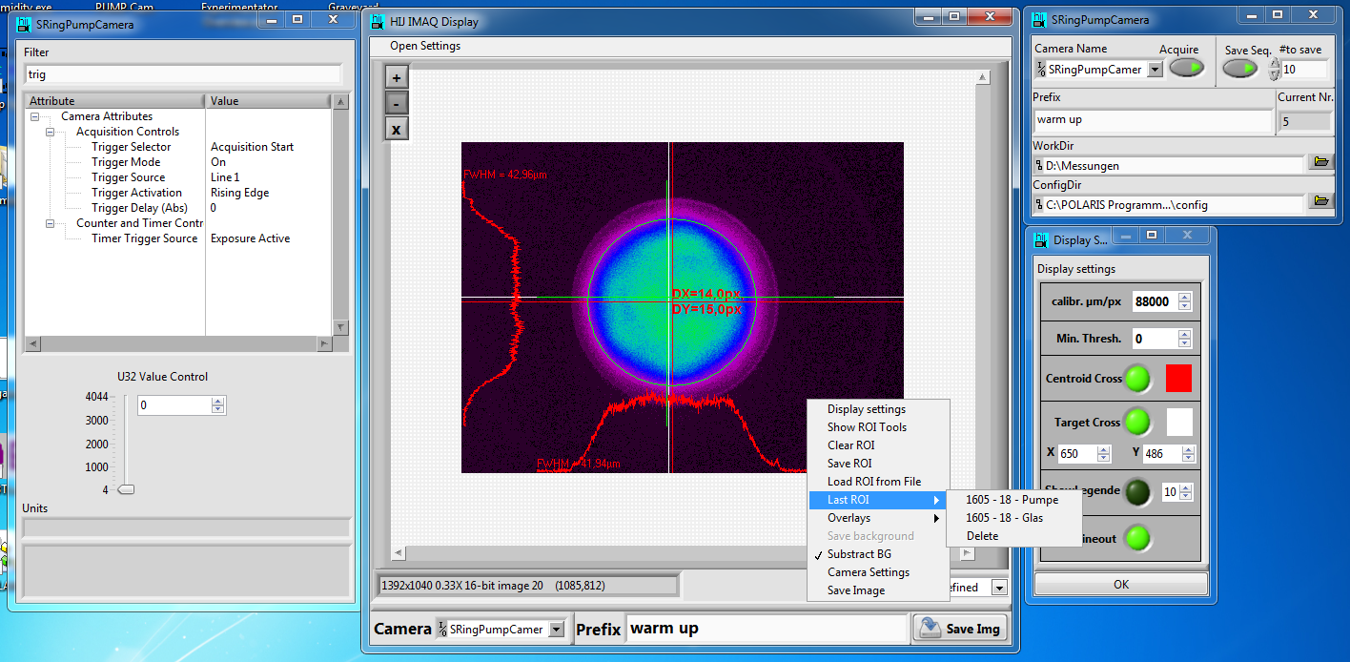
HIJ Vision

HIJ Vision ist aus den Erfahrungen mit NI Vision im Laserlaborumfeld entstanden. Weil die Bildaufnahme und Auswertung in LabVIEW sehr einfach ist, kann im Prinzip jede mit LV oberflächlich vertraute Person eine kamerabasierte Anwendung erstellen. Nun hat bereits die einfachste Anwendung i.d.R. drei Aufgaben: Bildaufnahme, Auswertung und Anzeige. Viele Anforderungen sind für die meisten Applikationen identisch, Abziehen eines Hintergrundbildes, Anpassen der Kameraeinstellungen, Einblenden der Ziel- und Istkoordinaten, Abspeichern eines Bildes und einer Bildsequenz, nachträgliche Auswertung. Wenn man diese Aufgaben im klassischen LV Still implementiert, bekommt man ein riesiges VI (vielleicht mit SubVIs). Oft ist es auf konkretes Kameratyp zugeschnitten und kann nicht wiederverwendet werden. Zwei von unterschiedlichen Personen erstellte Programme wurden zwar jeweils neu programmiert, unterscheiden sich vielleicht in 10%.

Diese Bibliothek beinhaltet wiederverwendbare und erweiterbare Bausteine, die nach dem LEGO Prinzip zu einer konkreten Anwendung zusammengebaut werden können. Die Bibliothek bietet folgende Funktionalität:



# Camera Klassen

Sendet New Image Msg an Image Receiver.

# Display Klassen

# Application Klasse

**Stellt Verbindung zwischen HIJ Vision Display und Analyze Actor.**

Übergibt eine New Results Msg der Applicationsklasse an HIJ Vision Display.

Zeigt Display in einem Subpanel.

**Weitere Konfiguration wird in After Launch Init ausgeführt. Hier wird z.B. zwischen Kamera und Display hergestellt.**

### HIJ Vision Camera App.lvclass

Die Klasse bildet die Besonderheiten einer Kamera Anwendung ab. Dazu gehören:

* Setzen der Arbeits- und Display Konfigurationsverzeichnisse
* Setzen eines Präfixes für das Abspeichern der Bilder.
* Möglichkeit eine Bildsequenz zu speichern.

Diese Punkte können in Settings Dialog (SD) bearbeitet werden.

Wenn eine Referenz auf Subpanel übergegen wurde, wird SD dort eingeblendet. Falls nicht, wird in dem Container VI ein Menü Punkt für das Öffnen des SDs der jeweiligen Kameraanwendung gesetzt.

Im Actor Core wird der Kamera Actor gestartet. Wenn Acquire == True ist, wird sofort die Aufnahme gestartet.

**Die Verbindung zwischen Kamera und Display wird in After Launch Init hergestellt.** Dafür wird ein Kind der New Image Msg und der Display Enqueuer an Kamera Actor per einer Prioritätsnachricht geschickt.

# Anwenderprogramme (AP)

Ein Anwendergramm kann mehrere Applicationsobjekte (AO) beinhalten. Für jedes AO braucht man ein Subpanel. Die AOs können das Menübar des APs erweitern.

Beispiel mit einer Kamera

Beispiel mit zwei Kameras

Beispiel Fileviewer

Beispiel: Überschreiben eines Displayactors um Bilder mit speziellem Namen abzuspeichern.

## Erweitern der Bildauswertungsbibliothek

1. Lege eine neue Kindklasse von der Analyze Base.lvclass an.
2. Falls die Auswertung spezielle Parameter erfordert, lege eine Kindklasse von der Analyze Parameter Base.lvclass und definiere die Parameter in ihrem Cluster. Definiere die Zugriffmethoden via RK→VI for Member Access
3. Um die Parameter der neuen Auswerteklasse empfangen zu können, überschreibe Set Parameter.vi der Analyze Base.lvclass. Dort kannst Du per Typecast die Parameter der neuen Auswertung filtern.
4. Lege eine Kindklasse von der Analyze Results Base.lvclass an. Lege im Klassencluster die Variablen für Ergebnisse an und entsprechende Zugriffsfunktionen an.
5. Definiere dein Auswertealgorithmus in dem Du die Analyze Base:Analyze.vi überschreibts. Dort legst Du ein Objekt deiner Results Klasse an und übergibst ihm die Ergebnisse.
6. Wenn die Resultsklass ein Overlay im Bild erzeugen soll, überschreibe das Overlay.vi.