

TECHNOLOGIEGRUPPE  
ML / HMI

**GINZINGER**  
electronic systems

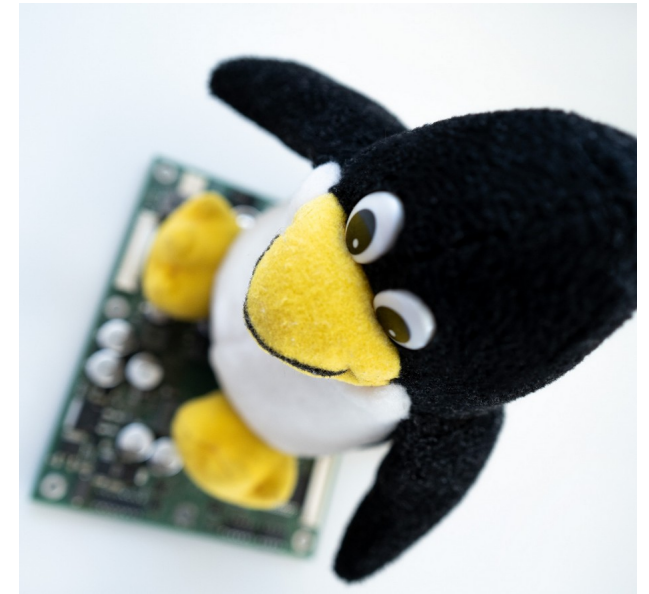
# EINLEITUNG - 1

- Warum Technologieentwicklung?
  - Fordern uns die bestehenden Kunden zu wenig?
  - Innovationen: Market Pull / Technology Push
- Technology Push
  - Schnell agieren bei neuen Kundenanforderungen
  - ... statt langsam reagieren bis der Kunde woanders hin geht



# EINLEITUNG - 2

- Beispiel Vision Systeme
  - Gibt es aktuell nicht im TLK, läuft unter ML(AI) / HMI
  - Optimale Ergänzung für das i.MX8MP Portfolio, da angenommen wird: HMI wird nicht so stark „wachsen“ wie für Geschäftserfolg nötig
  - Idee: Umkehrung: statt Bilder hinaus → Bilder hinein
  - Viel bestehende Technologie kann weiterverwendet werden:
    - GELin & Linux Erfahrung
    - i.MX8MP Plattform
  - Fehlendes Know-How wir in TLK aufgebaut



TEIL 1

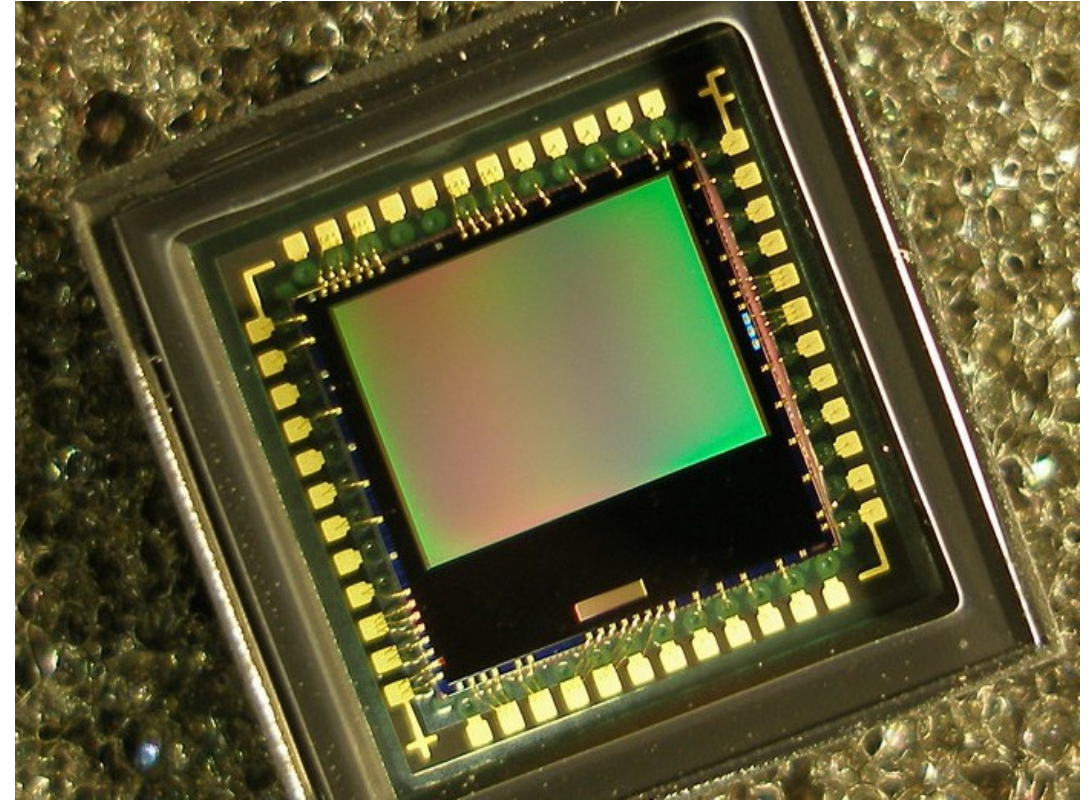
**SMARTE**

**(VIDEO) KAMERASYSTEME**

**ein Systemüberblick**

# KAMERAS – 1: BILDGEWINNUNG

- Bildsensor(en)
  - Fotozelle (1d - 2d)
  - Farbe (RGB)
  - Schwarzweiß (IR, VIS)
  - Wärembilder
  - Stereokameras (Leap Motion, Kinect...)



<https://de.wikipedia.org/wiki/Datei:Matrixw.jpg>

# KAMERAS – 2: OPTIK

- Optischer Pfad
  - Linse / Linsensysteme
    - Makro / Zoom / Weitwinkel
  - Blenden / Shutter
  - Filter
- Beleuchtung (aktiv (IR...), passiv)
- Projektionen
  - Laser / Linienbilder / Rahmen
  - Indirekt über Display

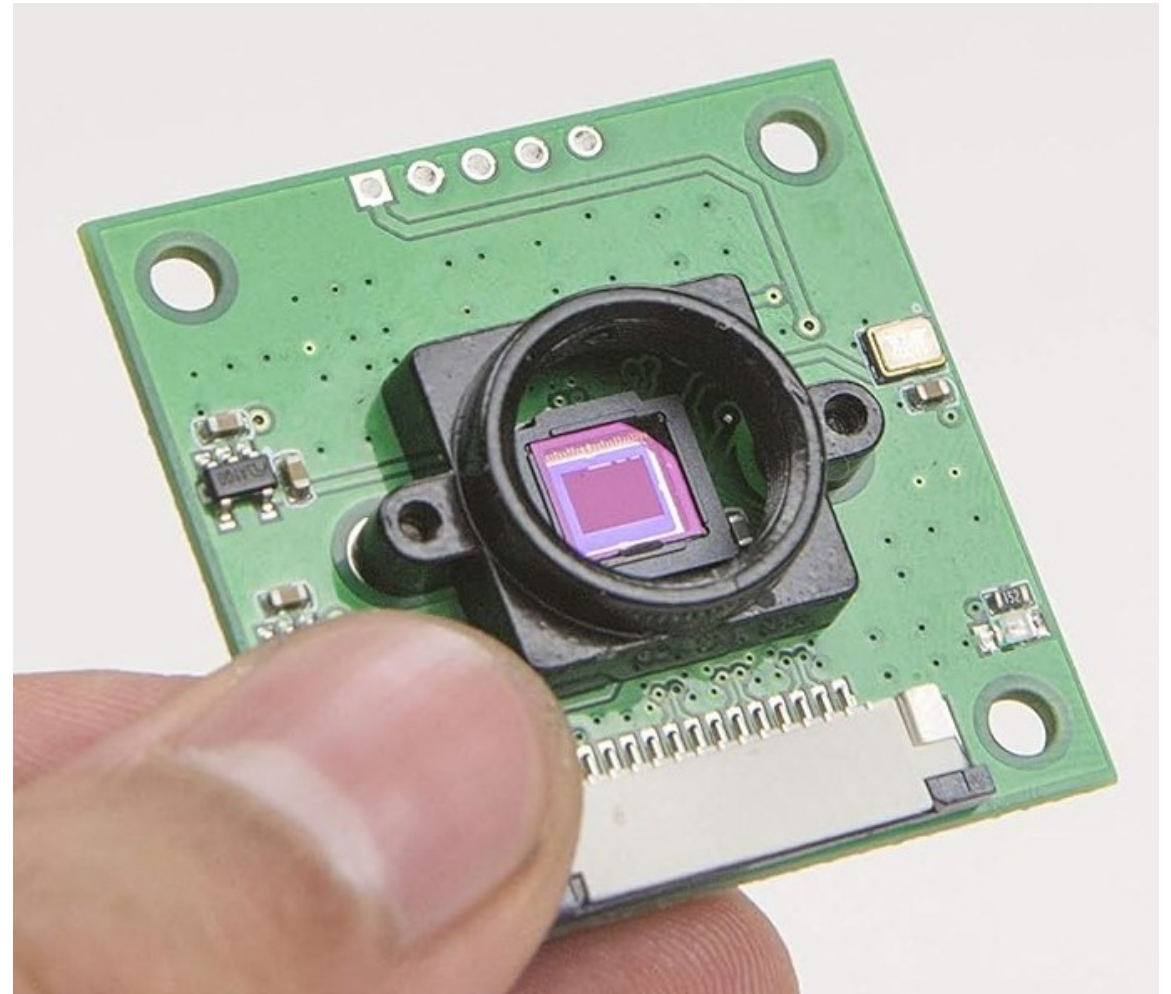


<https://www.arducam.com/product/m12-mount-camera-lens-kit-arduino-raspberry-pi/>



# KAMERAS – 3: MECHANIK

- Linsenhalter / Linse / Leiterplatte
- Justierung der Optik!
  - Fokusanpassung (Schärfe)
  - Blenden / Shutter / Filter
  - statisch/dynamisch
- Gehäuse (Kompaktheit, Dichtheit, Wärmeabfuhr, Montagemöglichkeit)
- Steckverbindungen (intern, nach außen)



<https://www.amazon.com/Arducam-Mount-Lens-Holder-Styles/dp/B07QMRDZYS>

# KAMERAS – 4: ELEKTRONIK 1

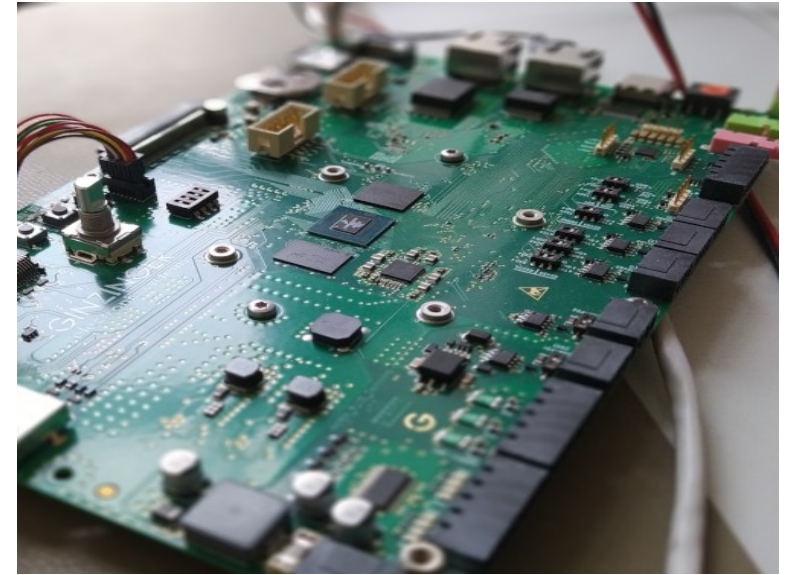
- PSU (AC/DC/POE/Batterie/Akku Versorgung)
- Zusätzliche Sensoren: (Helligkeit, Temperatur, Beschleunigung)
- Mikrofon
- RTC
- LED(s) (Aufnahme, Beleuchtung (IR...))
- Bussystem für Daten des Bildsensors (intern / extern)
  - Intern: CSI (~20cm)
  - Extern: Analoges Videosignal (xx m)
  - Extern: FPD-Link (~15m)
  - Extern: USB (~5 m)





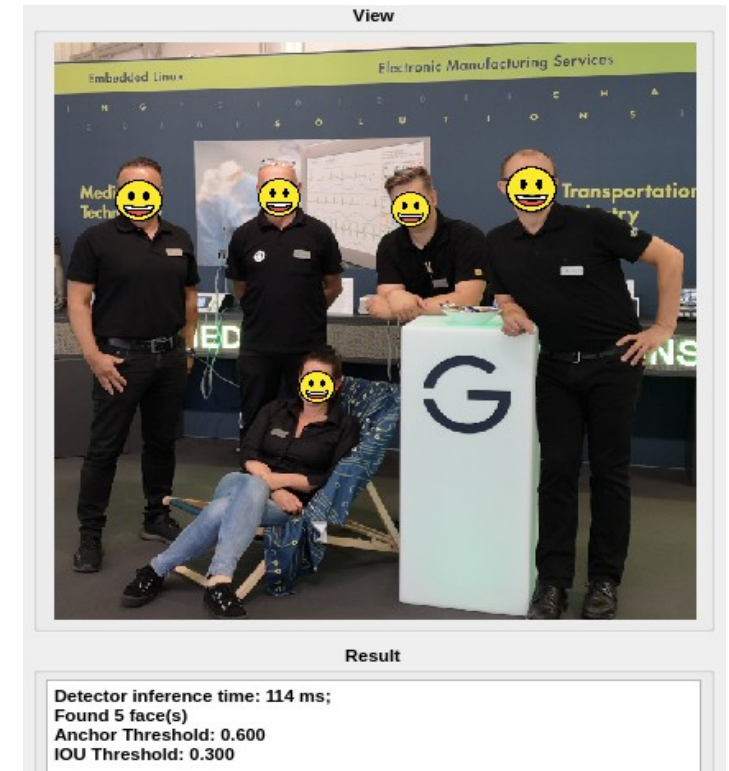
# KAMERAS – 5: ELEKTRONIK 2

- i.MX8MP Modul (VPU, NPU...)
- Speicher zur Zwischenspeicherung von Videodaten (SD...)
- GPIOs
  - Aktivierung externer Beleuchtung
  - Taster, Schalter: (Snapshot / Start-Stopp des Videos)...
- Mikrocontrollersysteme / FPGA
  - Schnelle Datenvorverarbeitung
- Bussysteme zur Weiterleitung der Videos / Daten (LAN, CAN, RSXXX...)
- Display für Vorschau, HMI...
- USB für HW-Lizenzschlüssel



# KAMERAS – 6: SOFTWARE

- Steuerung der HW (Blende, Shutter, externe Beleuchtung...)
- Datentransformation / Filter / Kompression
- Berechnungen (Tiefenbilder, Privacy...)
- Lizenzen (Keys für Proprietäre Softwarekomponenten)
- **Smart Kamera**
  - Datenverarbeitung per ML (NPU...)
  - Webinterface (Konfiguration, Status, Live Überwachung, Updates...)
  - Steuerung von IO's / simplen Bussystemen
  - Das Thema wird kommen, mit oder ohne uns...



## TEIL 2



# GStreamer

ein Überblick

# GSTREAMER – 1: EINLEITUNG

- Was ist GStreamer?
- Multimedia Framework
- Schweizer Taschenmesser zum
  - Erstellen
  - Editieren
  - Streamen
  - Konsumieren  
von Multimedia (**Video** und/oder Audio)

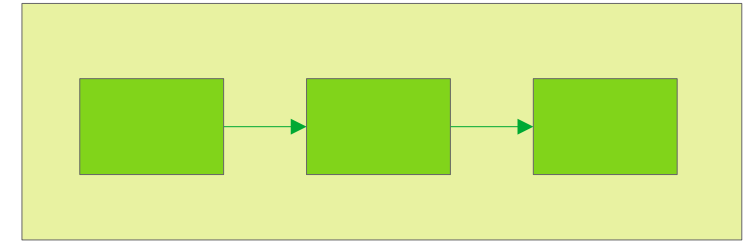
# GSTREAMER – 2: MODULARES KONZEPT 1

- Graph des Datenstroms: **Pipeline**
- **Plugin**: Element der Pipeline
  - Genau EINE Funktion: z.B: Videoquelle, Videosenke
  - Bestitzt **Quellen** und/oder **Senken** für Daten
  - **PADs** sind die Anschlüsse der Plugins (Sink Pads (In), Source Pads (Out))
  - Fähigkeiten der Plugins: **Capabilities** (Formate, Framerrates...)
  - Beliebig kombinierbar solange Capabilities zusammenpassen
  - Jeder Plugin kann einen vom mehreren Zuständen haben (gestoppt, warten, abspielen...)





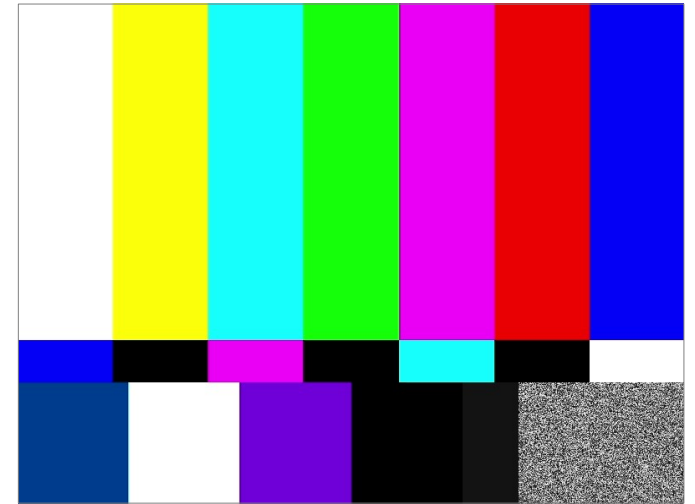
# GSTREAMER – 3: MODULARES KONZEPT 2



- GStreamer Framework
  - Kümmt sich um die Orchestrierung der Pipeline mit ihren Plugins
  - Synchronisiert der Plugins zueinander
  - Bietet gute Debugmöglichkeiten mit vielen Debuglevels (bis zur Überflutung...)
  - Kann Basis für grafische Darstellung der Pipeline ausgeben
  - Steuerung der Pipeline über eigenes Programm möglich
  - Aufbau und Start der Pipeline über Kommandozeile möglich

# GSTREAMER – 4: PLUGINS-2: QUELLEN

- Videosource
  - Kamera
  - Datei
  - Netzwerk
  - Fakesource



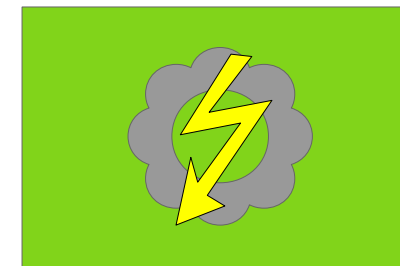
# GSTREAMER – 5: PLUGINS-3: SENKEN

- Videosink
  - Monitor (autovideosink, kmsink(embedded)...)
  - Anwendung (Qt...)
  - Datei
  - Netzwerk
  - Fakesink
  - FPS Sink
    - für Zeitmessungen als Zwischenschritt zum Debugging
    - Als Option wird tatsächliche Sink benötigt



# GSTREAMER – 6: PLUGINS-4: VERARBEITUNG

- Filter
- Queue (Warteschlange/Puffer)
- Videoconvert (Format z.B.: YUV→ RGB...)
- Videocodec (Encode / Decode): SW oder HW(VPU)
- Streaming ((De)-Payloader)
- Multiplexer / Demultiplexer (z.B.: Audio/Video zusammenfügen/trennen)
- Mixer (z.B.: Greenbox/Alphakanal mit Hintergrund mischen)



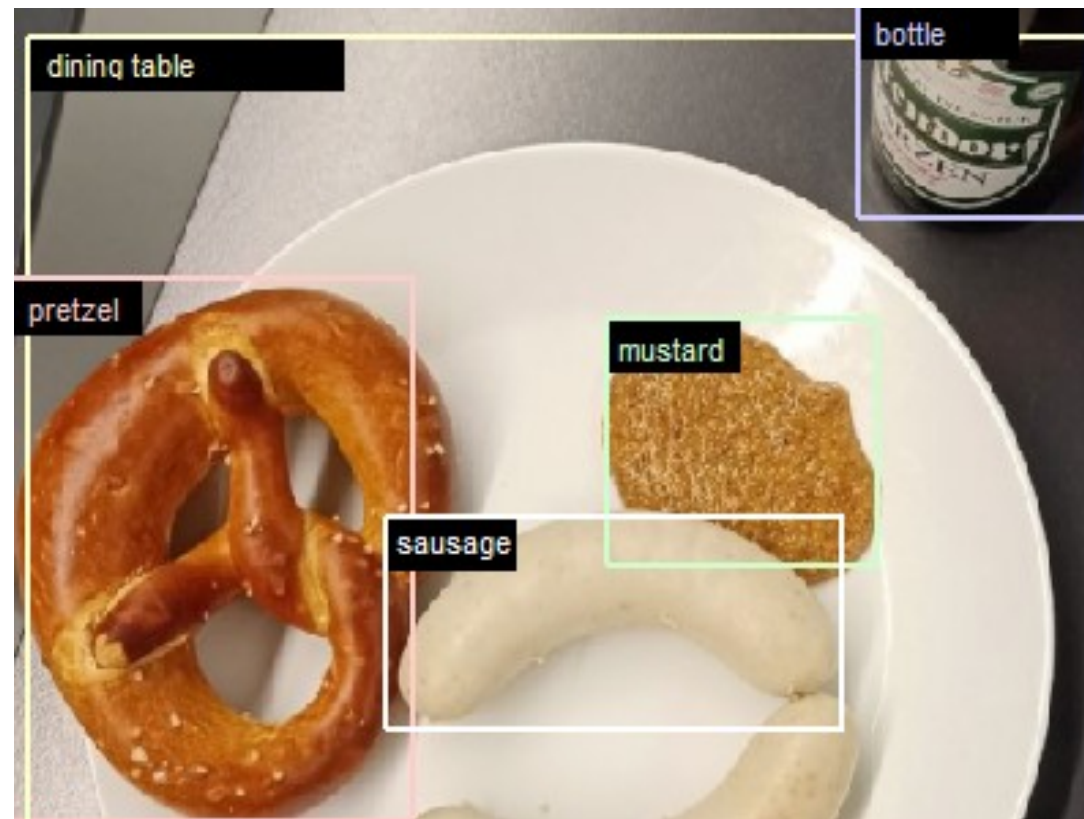
# GSTREAMER – 7: DEMO

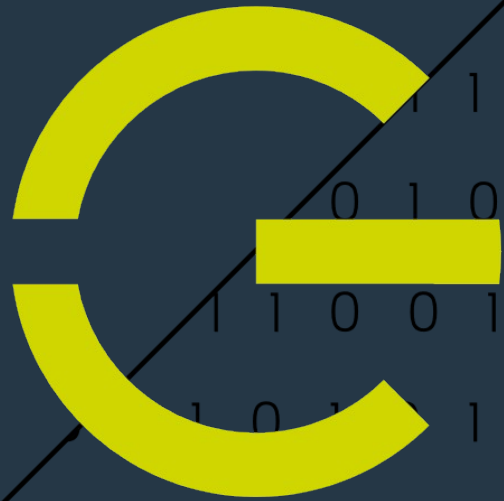
- Fakevideosink → Monitor
- USB-Kamera → Filter → Monitor



# ENDE

- Smarte (Video-) Kamerasysteme ein Systemüberblick
- GStreamer
- Fragen?





CONVERTING CHALLENGES  
INTO SOLUTIONS

**GINZINGER**  
electronic systems