

### Zusammenfassung

Dieses Readme deckt die Inbetriebnahme der am Medizinischen Interprofessionellen Trainingszentrum (MITZ) des Universitätsklinikum Dresden entwickelten EKG-Selbstlerneinheit für Studierende. Es beschreibt materielle und technische Voraussetzungen, ein empfohlenes räumliches Setup, Konfiguration und die Inbetriebnahme. Für substantielle Teile dieses Guides wird Unterstützung der Haus-IT benötigt, das Guide ist geschrieben für Ubuntu Linux.

## 1 Materielle, Technische Voraussetzungen

Zur idealen Verwendung des Projektes wird benötigt

- Eine medizinische Übungspuppe in Lebensgröße
- Ein EKG-Gerät mit 10 Elektroden + Klebepads
- Zwei USB-Kameras mit hoher Auflösung
- Ein PC zum Ausführen der Projektsoftware
- Ein Drucker, Klebeband und ein Klebestift

### 1.1 USB-Kameras

Für stabile Anwendung sind Kameras mit hoher Auflösung, 4K UHD oder höher, empfohlen. Das Originalsetup basiert auf zwei „HP 960 4K“. Das verwenden verschiedener Modelle ist meist möglich, solange die Kameras mit der gleichen Auflösung aufnehmen.

### 1.2 PC

Dieser Teil sollte mit der IT-Abteilung besprochen werden. 🏢 ⚠️

Der PC benötigt mind. zwei freie USB-Ports (USB 2.0+). Das Projekt wurde mit Zielplattform Ubuntu Linux (Version `abc-xyz`) entwickelt.

Auf dem PC muss die Umgebungsverwaltung Conda installiert sein. Aus Rechts- und Lizenzgründen sollte die über das Community-Projekt [Conda-Forge](#) bereitgestellte Variante („miniconda“) genutzt werden. Das Projekt wurde mit Conda-Forge Version `abc-xyz` entwickelt, sollte aber auch mit anderen Versionen funktionieren.

Nachdem conda installiert wurde, kann es im Terminal mit dem Befehl `conda` verwendet werden. Conda stellt isolierte Laufzeitumgebungen zur Verfügung und muss noch über das Terminal konfiguriert werden:

1. Neue Umgebung erstellen: `conda create -n hybparc python=3.9`
2. Neue Umgebung Aktivieren: `conda activate hybparc`

Nun ist das offene Terminal auf die neue Umgebung gesetzt. Es müssen zuerst „Pip“ (Python Paketmanager) und danach via Pip noch Python Pakete installiert werden. Sollte im Terminal nachgefragt werden, ob zusätzliche Abhängigkeiten (dependencies) mitinstalliert werden sollen, sollte dies mit `y` bestätigt werden.

1. Pip installieren: `conda install pip`
2. OpenCV installieren: `pip install opencv-contrib-python`
3. PyQt6 installieren: `pip install pyqt6`

Es sind alle dependencies des Projektes installiert, und das Projekt selbst kann heruntergeladen und konfiguriert werden.

## 2 Projekinstallation, Konfiguration

Dieser Teil sollte mit der IT-Abteilung besprochen werden. 🏠 ⚠️

Es ist empfohlen, einen vorgefertigten Release von [Twillo? Github?](#) herunterzuladen. Bei Bedarf steht der aktuelle Sourcecode auf [GitHub](#) verfügbar.

Sollten die verwendeten Kameras die **exakt** gleiche Auflösung (3840x2160) wie die des Originalsetups haben, und MJPEG unterstützen, sollte das Projekt (unter Ubuntu Linux) nun ohne Probleme funktionieren.

Potentiell müssen die interface IDs angepasst werden, diese können mit den `v4l2` command line tools (müssen potentiell nachinstalliert werden) ausgelesen werden: `v4l2-ctl -list-devices`. Die Konfigurations-Reihenfolge"der Kameras (Unterkörper vs. Oberkörper) spielt keine Rolle.

### 2.1 Andere Auflösung 🙄

Bei anderen Auflösungen müssen im Code unter `/hybparc_aruco/main.py` fast zu Beginn der `__init__` Funktion die entsprechenden Parameter angepasst werden:

```

32
33     # Setup and warm up cameras
34     #! Adjustments are specific to the HP 960 4K in our physical setup
35     self.stream0 = cv.VideoCapture(index=0, apiPreference=cv.CAP_ANY)
36     self.stream0.set(cv.CAP_PROP_FOURCC, cv.VideoWriter_fourcc('M', 'J', 'P', 'G'))
37     self.stream0.set(cv.CAP_PROP_FRAME_WIDTH, 3840) ←
38     self.stream0.set(cv.CAP_PROP_FRAME_HEIGHT, 2160) ←
39     self.stream1 = cv.VideoCapture(index=4, apiPreference=cv.CAP_ANY)
40     self.stream1.set(cv.CAP_PROP_FOURCC, cv.VideoWriter_fourcc('M', 'J', 'P', 'G'))
41     self.stream1.set(cv.CAP_PROP_FRAME_WIDTH, 3840) ←
42     self.stream1.set(cv.CAP_PROP_FRAME_HEIGHT, 2160) ←
43

```

### 2.2 Kein MJPEG 😞

Die unterstützten Formate einer Kamera bzw. Ihres Interfaces mit der Nr. X können via `v4l2-ctl -d /dev/videoX -list-formats-ext` ausgelesen werden. Der entsprechende Codec muss im [FourCC Format](#) unter `/hybparc_aruco/main.py` fast zu Beginn der `__init__` Funktion angepasst werden:

```

32
33     # Setup and warm up cameras
34     #! Adjustments are specific to the HP 960 4K in our physical setup
35     self.stream0 = cv.VideoCapture(index=0, apiPreference=cv.CAP_ANY)
36     self.stream0.set(cv.CAP_PROP_FOURCC, cv.VideoWriter.fourcc('M', 'J', 'P', 'G'))
37     self.stream0.set(cv.CAP_PROP_FRAME_WIDTH, 3840)
38     self.stream0.set(cv.CAP_PROP_FRAME_HEIGHT, 2160)
39     self.stream1 = cv.VideoCapture(index=4, apiPreference=cv.CAP_ANY)
40     self.stream1.set(cv.CAP_PROP_FOURCC, cv.VideoWriter.fourcc('M', 'J', 'P', 'G'))
41     self.stream1.set(cv.CAP_PROP_FRAME_WIDTH, 3840)
42     self.stream1.set(cv.CAP_PROP_FRAME_HEIGHT, 2160)
43

```

### 3 🚨 Troubleshooting 🚨

Um das Projekt über das Terminal zu starten, oder um Änderungen am conda environment vorzunehmen, muss das environment bei jedem Terminal neustart wieder aktiviert werden (conda activate hybparc).

**OpenCV Package Type** Das Guide empfiehlt opencv-contrib-python zu installieren. Dies ist eine Ubuntu Linux spezifische Empfehlung. Sollte das Projekt unter MacOS und Windows verwendet werden empfiehlt sich das Package opencv-python.