

# \*Youtube: Kobra Display mit Klipper

Erstellt von seho85 (Anycubic Vyper)

[seho85/klipper-dgus: A python project to connect a DGUS display to Klipper \(github.com\)](#)

Adaptiert für den Anycubic Kobra von fragmon

## Hardware

Folgende Kobra Modelle wurden getestet:

- Kobra Max mit Trigorilla Pro A 1.0.4
- Kobra Max mit Trigorilla Pro B 1.0.2

Theoretische Kompatibilität (Modelle werden nach Tests aufgenommen):

- Kobra
- Kobra Plus
- Kobra 2

Nicht kompatibel:

- Kobra Neo (kein DWIN Display)

Zur Nutzung wird das Display nicht mehr am Drucker, sondern am Raspberry Pi über dei GPIO direkt angeschlossen.

BOM:

- Dupontstecker 4 Pin (Anschluss GPIO Raspberry Pi)
- JST XH Stecker 6 Pin (Anschluss Kobra Display)
- Kabel

## Installation

Die Anleitung ist einzig für einen Raspberry Pi ausgelegt. Solltet ihr eine andere MCU betreiben, so müssen die Dateien ggf. angepasst werden.

## Displayverbindung

Das Display wird über die GPIO Leiste mit dem Raspberry Pi verbunden.

### 1. Aktivieren Sie die Serienschnittstelle auf Raspberry Pi

```
sudo raspi-config
```

\* \*3) Schnittstellenoptionen\*

\* \*6) Serielle Schnittstelle\*

\* Möchten Sie, dass eine Anmeldeschale über die serielle Schnittstelle zugänglich ist?

\* Nein

\* Möchten Sie, dass die Serienschnittstellenhardware aktiviert wird?

\* Ja

### 2. Trennen Sie Raspberry Pi von der Stromversorgung\*\*

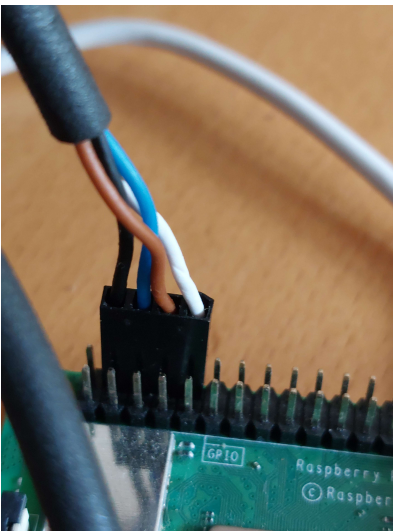
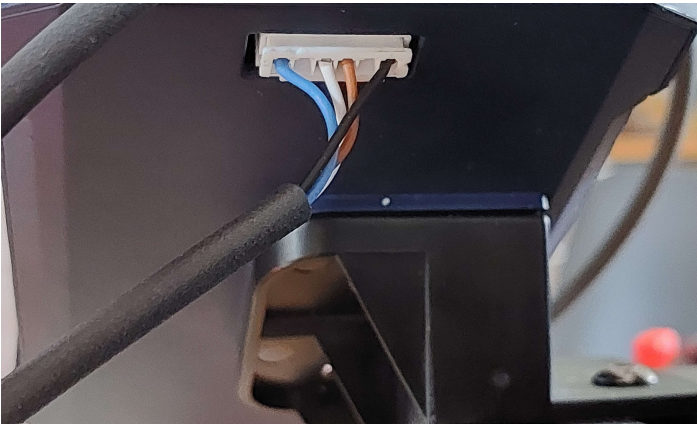
### 3. Verbinden Sie das Display mit dem Raspberry Pi GPIO Pin Header

#### ACHTUNG

Es wird dringend empfohlen, das Display anzuschließen, wenn der Raspberry Pi ausgeschaltet ist!

Display-Steckverbinder

=====



## Display flashen

1. Lade die DWIN\_SET.zip (erstellt mit dem DGUS-Tool), entpacke und kopiere diese auf eine korrekt formatierte MicroSD-Karte.

[https://github.com/cryd-s/kobra\\_extended/blob/main/dgus\\_project/DWIN\\_SET.zip](https://github.com/cryd-s/kobra_extended/blob/main/dgus_project/DWIN_SET.zip)

2. Stecke die MicroSD-Karte in das ausgeschaltete Display

3. Schalte das Display ein

4. Eine blaue Displaymaske zeigt den Fortschritt des Flashens an.

5. Während des Flash-Vorgangs zeigt die zweite Zeile an:

SD Card Process...

Wenn das Flashen abgeschlossen ist, zeigt die zweite Zeile an:

SD Card Process... END!

Schalte das Display aus

Entferne die MicroSD-Karte

**ACHTUNG:**

Es ist von entscheidender Bedeutung, dass die MicroSD-Karte korrekt formatiert ist:

- Dateisystem: FAT32
- Zuordnungseinheitsgröße: 4096

## Installationsskript

Für eine einfache Installation auf MainsailOS ist ein Einrichtungsskript namens `install_for_mainsailos.sh` im Verzeichnis des Projektordners zu finden.

1. Kopieren der benötigten Daten

```
git clone https://github.com/cryd-s/kobra_extended.git
mv kobra_extended klipper-dgus
chmod -R 700 klipper-dgus
cd klipper-dgus
```

2. Führen das Installations-Skript aus:

```
./install_for_mainsailos.sh
```

**ACHTUNG** Während der Installation werden Sie aufgefordert, Ihr sudo-Passwort einzugeben.

## Konfigurationsdateien

**dgus\_display\_macros.cfg**

Diese Konfigurationsdatei muss in den config Ordner kopiert und in der *printer.cfg* eingebunden werden.

```
[include dgus_display_macros.cfg]
```

Die Datei definiert die Aktionen, die von einigen Steuerelementen des Displays ausgeführt werden.

## ACHTUNG

Je nachdem, welchen Drucker Sie verwenden, ist eine Anpassung dieser Datei erforderlich (siehe Kommentare in der Datei *dgus\_display\_macros.cfg*)

## serial\_config.json

Die Datei *serial\_config.json* enthält die Einstellungen für die serielle Schnittstelle, die zur Kommunikation mit dem DGUS-Display verwendet wird.

Die Angabe ist abhängig vom verbauten Raspberry PI sowie der Rasbian Version. Wählt zunächst eine Schnittstelle aus und sollte das Display nicht starten, testet hilfsweise die zweite Bezeichnung. Sollte dies auch nicht funktionieren schaue bitte und Troubleshooting.

Standardmäßig wird folgender Schnittstelle genutzt:

```
"serial_port": "/dev/ttyS0"
```

Sollte das Display nicht funktionieren, kann mit diesem Befehl, die Schnittstelle geändert werden:

ttyS0 zu ttyAMA0

```
sed -i '3s|/dev/ttyS0|/dev/ttyAMA0|' /home/pi/klipper-dgus/config/serial_config.json
```

wenn es wieder zurückgesetzt werden soll (ttyAMA0 zu ttyS0)

```
sed -i '3s|/dev/ttyAMA0|/dev/ttyS0|' /home/pi/klipper-dgus/config/serial_config.json
```

## websocket.json

Die Datei *websocket.json* enthält die Einstellungen für die Websocket-Verbindung, die zur Kommunikation mit Klipper über den Moonraker-Socket verwendet wird.

In der Regel müssen nur die ersten beiden Parameter *\*ip\** und *\*port\** geändert werden.

Neben den Einstellungen *\*ip\** und *\*port\** gibt es einige andere Objekte, die hier beschrieben werden.

```
{
  "websocket": {
    "ip": "127.0.0.1",
    "port": 7125,

    "printer_objects": {
      "some_more_entries" : "...",
    },

    "data_model": {
      "some_more_entries" : "...",
    }
  }
}
```

#### IP

IP-Adresse der Maschine, auf der Moonraker läuft. Wenn MainsailOS verwendet wird, ist es dieselbe IP-Adresse, wie die zur Verbindung mit der Weboberflächeport

#### Port

Der Port, auf dem Moonraker überträgt. Standardmäßig läuft es auf 7125 und es sind keine Änderungen erforderlich.

#### printer\_objects

Hier sind die *\*Druckerobjekte\** definiert, die vom Display verwendet werden. Siehe [https://moonraker.readthedocs.io/en/latest/printer\\_objects/](https://moonraker.readthedocs.io/en/latest/printer_objects/)

Änderungen hier können erforderlich sein, wenn eine neue Information im Display verwendet werden soll.

#### data\_model

In *data\_model* befinden sich die zuletzt mit Moonraker ausgetauschten Daten. Dieser Abschnitt benötigt überhaupt keine manuellen Anpassungen. Er wird von der Display-Controller-Anwendung generiert. Manuelle Änderungen werden beim Starten/Herunterfahren der Controller-Anwendung überschrieben.

