

Erzeugen einer "Gerber-Export-Datei" aus einer PNG-Bilddatei - Kochrezept für das [Tool KiCad 10.0](#)

Prozess-Zusammenfassung

Anmerkung: Ordner, Dateinamen und Steuerelemente sind farbig unterlegt und in Schriftart **Courier New** dargestellt.

01: Das Layout-Bild für KiCad vorbereiten

Ergebnis ist die Bilddatei **Bildname.png**

02: Projekt-Ordner vorbereiten und öffnen

Die Projekt-Bezeichnung **XXX** wird hier global für alle Ordner/Dateien verwendet

Anlegen eines Projekt-Ordners **XXX**

Im Projekt-Ordners **XXX** den Unterordner **GBR_XXX** und den

Unterordner **Footprint.kicad_mod** anlegen

Vorbereitete Projekt-Datei **XXX.kicad_pro** in den Projektordner kopieren

Vorbereitete Bilddatei **Bildname.png** in den Projektordner kopieren

03: KiCad 10.0 starten und Projekt öffnen

04: Bildumwandler: PNG-Bilddatei in Footprint-Datei wandeln

Bilddatei **Bildname.png** öffnen

Ausgabegröße (z.B. 160x100 mm) und **Ausgabeformat (F.Cu)** einstellen

Footprint **XXX.kicad_mod** in Ordner **Footprint.kicad_mod** speichern

05: Footprinteditor: Footprint-Datei bearbeiten

XXX.kicad_mod importieren

Lagen auf **Edge.Cuts** umstellen

Unterordner **Bibliothek.pretty** einrichten

Bearbeitete **XXX.kicad_mod** in **Bibliothek.pretty** speichern

06: Leiterplatteneditor: Footprint-Datei bearbeiten

Das **Printed Circuit Board** (Datei **XXX.kicad_pcb**) erstellen

Footprint-Datei XXX.kicad_mod aus **Bibliothek.pretty** laden

06.1: Layout-Begrenzungsrahmen festlegen

Lagen ändern in **Edge.Cuts**, Werkzeug **Rechteck zeichnen** einstellen

Exaktes Rechteck um das Layout (Footprint) legen

Datei **XXX.kicad_pcb** mit zusätzlichem Rahmen **Edge.Cuts** speichern

06.2: Lötstopmmaske für Vorderseite (Frontsite) festlegen

Auswahl **Edge.Cuts** aufheben und **Lagen** auf **F.Mask** umstellen
Werkzeug **Rechteck zeichnen** einstellen
Exaktes Rechteck um das Layout (Footprint) legen
Rechteck **F.Mask** füllen
Datei **XXX.kicad_pcb** mit zusätzlichem Rahmen **F.Mask** sichern

06.3: Lötstopmmaske für Rückseite (Backsite) festlegen

Auswahl **F.Mask** aufheben und **Lagen** auf **B.Mask** umstellen
Werkzeug **Rechteck zeichnen** einstellen
Exaktes Rechteck um das Layout (Footprint) legen
Rechteck **B.Mask** füllen
Datei **XXX.kicad_pcb** mit zusätzlichem Rahmen **B.Mask** sichern

06.4: Layout für fertige Leiterplatte "plotten"

Menü **Plotten** aufrufen **Datei => Plotten ...**
Lagenauswahl bestimmen auf: **F.Cu**, **F.Mask**, **B.Mask** und **Edge.Cuts**
Ausgabeverzeichnis **GBR_XXX** auswählen
Datei **XXX.kicad_pcb** plotten
Sicherung des **Printed Circuit Board XXX.kicad_pcb**
Leiterplatteneditor Schließen

07: Gerber-Betrachter: Layout-Kontrolle

Datei => Gerber-Plotdatei(en) öffnen ...

Gerber-Dateien der Reihe nach kontrollieren:

XXX-B_Mask.gbr	Lötstopmmaske für Rückseite
XXX-Edge_Cuts.gbr	Layout-Begrenzungsrahmen
XXX-F_Cu.gbr	Layout
XXX-F_Mask.gbr	Lötstopmmaske für Vorderseite

08: Die ZIP-Datei für den Hersteller fertigen

Alle Dateien im Ordner **GBR_XXX** markieren und mit ...

... **Senden an => ZIP-komprimierter Ordner "zippen"**

Ergebnis:

XXX.zip
XXX-B_Mask.gbr
XXX-Edge_Cuts.gbr
XXX-F_Cu.gbr
XXX-F_Mask.gbr
XXX-job.gbr job

Prozess - Ausführlich

Der sinnvollste und einfachste Weg führt über die Nutzung des **Bildkonverters KiCad**. Da die Vorlagen bereits im Format **1:1** (z.B. **160 x 100**, **100 x 160** oder **200 x 150 mm**) vorliegen, entfällt kompliziertes Skalieren in einem CAD-Programm.

Da die Bohrungen nach der Fertigung der Platine selbst vorgenommen und keine Lötstopmmaske benötigt werden, muss als Ergebnis lediglich eine ZIP-Datei mit den Gerber-Dateien für die Produktion des **XXX-Projektes** erzeugt werden. Diese Gerber-Dateien stehen für:

XXX-F_Cu.gbr	Kupferschicht (die Leiterbahnen auf der Vordserseite)
XXX-Edge_Cuts.gbr	Außenkontur (der Platinenumriss)
XXX-A_Maske.gbr	Lötstopmmaske für die Vorderseite (ausgefüllt = kein Lack)
XXX-B_Maske.gbr	Lötstopmmaske für die Rückseite (ausgefüllt = kein Lack)
XXX-job.gbrjob	Gerber-Job-Datei , Zusammenfassung der Arbeitsschritte

Um fertige Gerber-Dateien aus einer Bilddatei zu erzeugen ist eine hochaufgelöste **Bilddatei Bildname.png** am besten geeignet. Der Wandlungsprozess findet für jedes Layout in einem gesonderten Gerber-Projekt statt.

Ein Gerber-Projekt besteht aus den hier reduzierten folgenden Primär-Elementen:

Ordner und Projektdateien

- **.kicad_pro (Project)**: Das ist die **Haupt-Projekt-Datei**. Allgemein hält sie das gesamte Projekt zusammen, verwaltet die Verknüpfungen zwischen Schaltplan und Platine und speichert allgemeine Projekteinstellungen. Zur Vereinfachung der hier verwendeten Projekte wird sie bereits anfangs als eine vereinfachte vorgefertigte **Vorlage-Projekt-Datei** im **Projekt-Ordner**

XXX

vorinstalliert als:

XXX.kicad_pro [Sie ist hier downloadbar](#)

(XXX steht für die beabsichtigte individuelle Projekt-Bezeichnung)

- **.kicad_mod (Footprint / Moduls)**: Das ist die Datei für ein einzelnes **Bauteil (z.B. Footprint)**. Sie beschreibt die physische Form des Bauteils. In dem hier verwendeten Fall ist es das Platinen-Layout, d.h. die Summe aller Leiterbahnen aus Kupfer und aller Zwischenräume ohne Kupfer. Das ist die Footprint-Ausgangsdatei

XXX.kicad_mod

die im Ordner

Footprint.kicad_mod

abgelegt wird.

- **.kicad_pcb (Printed Circuit Board)** ist die Datei-Endung mit allen Daten für den Fertigungsprozess der Leiterplatte. Die Dateiendung **.kicad_pcb** verrät dem Computer also, dass es sich bei dieser Datei um das eigentliche **Platinen-Layout** von **KiCad** handelt – also das Dokument, in dem allgemein

die Leiterbahnen, Lötstopps, Bauteile, der Platinenumriss usw. platziert und miteinander verbunden sind.

- Das Kürzel **.pretty** bezeichnet in **KiCad** einen **Footprint-Bibliotheksordner**. Jedes Bauteil stellt eine eigene Datei dar. Da hier nur ein Bauteil (nämlich die Oberfläche der Platine mit der strukturierten Kupfer-Schicht) bearbeitet wird, steht zum Ende nur die fertige bearbeitete Footprint-Datei

XXX.kicad_mod

im Bibliotheksordner

Bibliothek.pretty

- **.kicad_prl** (**Project Local Settings**): Diese Datei stellt ein **Bildschirmprotokoll** dar und ist für die Platinenfunktion unwichtig.

Beispiel für eine Umwandlung mit konkreten Daten

Das neue Projekt erhält den Namen: **Proj_203_AVR_ALE_Testboard**.

Die hier als Beispiel verwendete Platine ist das [Experimentierboard für AVR-Mikrocontroller](#), das im [AVR-Tutorial](#) für Versuchsschaltungen verwendet wird.

01: Das Bild für KiCad vorbereiten

Dieser Schritt wird mit **CorelDRAW** und dem Bildprogramm **IrfanView** vollzogen.

1. Die mit **CorelDRAW** (oder einem anderen Zeichenprogramm) erzeugte **Bildname.png**-Datei (300 DPI) wird im Bildbearbeitungsprogramm **IrfanView** geöffnet.
2. Das Bild muss als **reines Schwarz-Weiß-Bild** (keine Graustufen) **ohne Ränder** (d.h. als exaktes Layout-Format) vorliegen.
3. Die Leiterbahnen müssen **schwarz** und der Hintergrund (wo das Kupfer später weggeätzt wird) **weiß** sein. Sollte es umgekehrt sein, müssen die Farben im Bildprogramm konvertiert werden.
4. Mit **IrfanView** wird die Bildgröße von **300 DPI** auf **1200 DPI** hochskaliert.

02: Projekt-Ordner vorbereiten und öffnen

1. Es wird ein neuer Projekt-Ordner **XXX** angelegt. Zum Beispiel:
Proj_203_AVR_ALE_Testboard
2. Von der Website <https://www.alenck.de/> wird die vorbereitete Vorlage-Projekt-Datei **XXX.kicad_pro** geladen und mit dem geänderten Projektnamen als neue Projekt-Datei angelegt:
Proj_203_AVR_ALE_Testboard.kicad_pro
3. Die vorbereitete **Bildname.png**-Bilddatei wird in den neuen Projekt-Ordner kopiert, zum Beispiel ([Bilddatei](#) als Vorlage):
203_Bild_3_1_03_AVR_ALE_Testboard_Platine_160x100.png

4. Nun werden noch der Footprint-Ordner und der Gerber-Datei-Ordner als leere Ordner in den Projekt-Ordner eingefügt:

Footprint.kidat_mod

und

GBR_XXX

zum Beispiel:

GBR_Proj_203_AVR_ALE_Testboard

Damit ist der Projekt-Ordner für den Umwandlungsprozess vorbereitet; zum Beispiel:

Proj_203_AVR_ALE_Testboard

Footprint.kidat_mod

GBR_Proj_203_AVR_ALE_Testboard

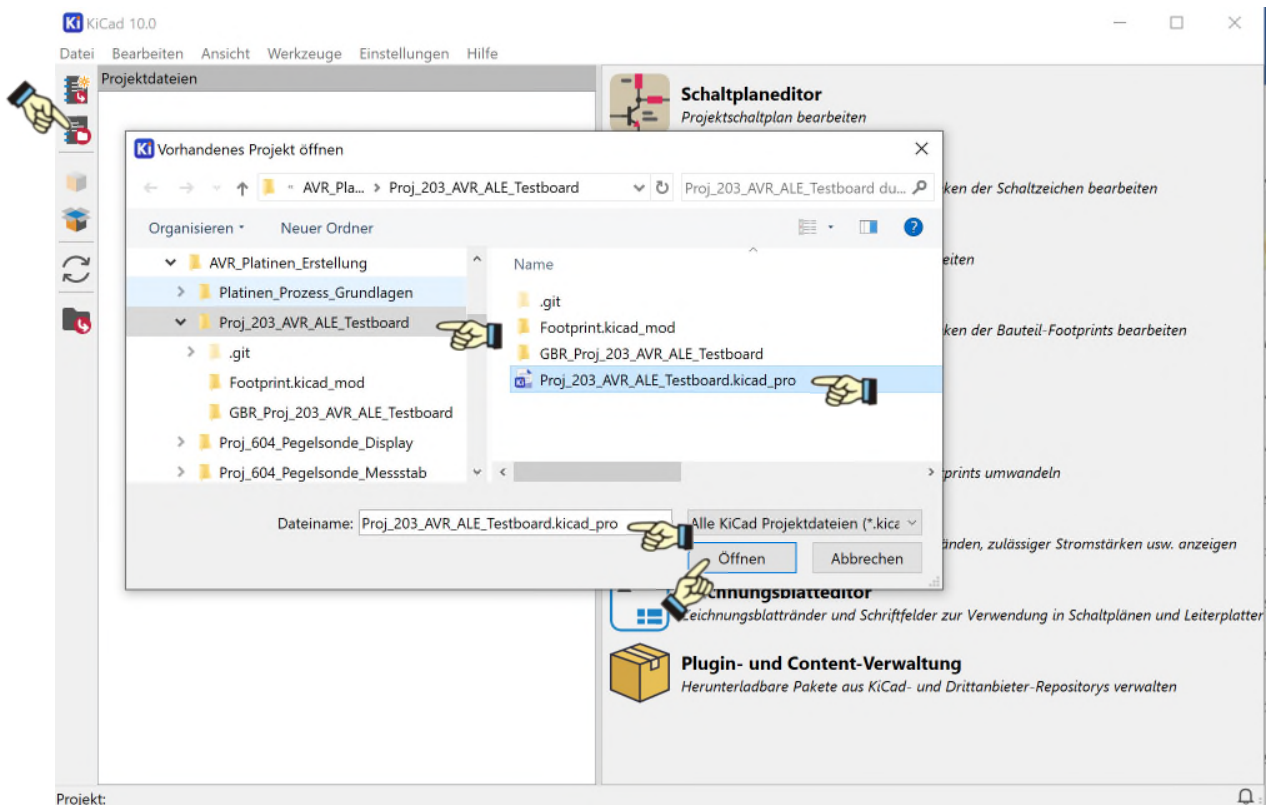
203_Bild_3_1_03_AVR_ALE_Testboard_Platine_160x100.png

Proj_203_AVR_ALE_Testboard.kidat_pro

03: KiCad 10.0 starten und Projekt öffnen

Als Beispiel wird das Projekt **Proj_203_AVR_ALE_Testboard.kidat_pro** in allen Phasen beschrieben.

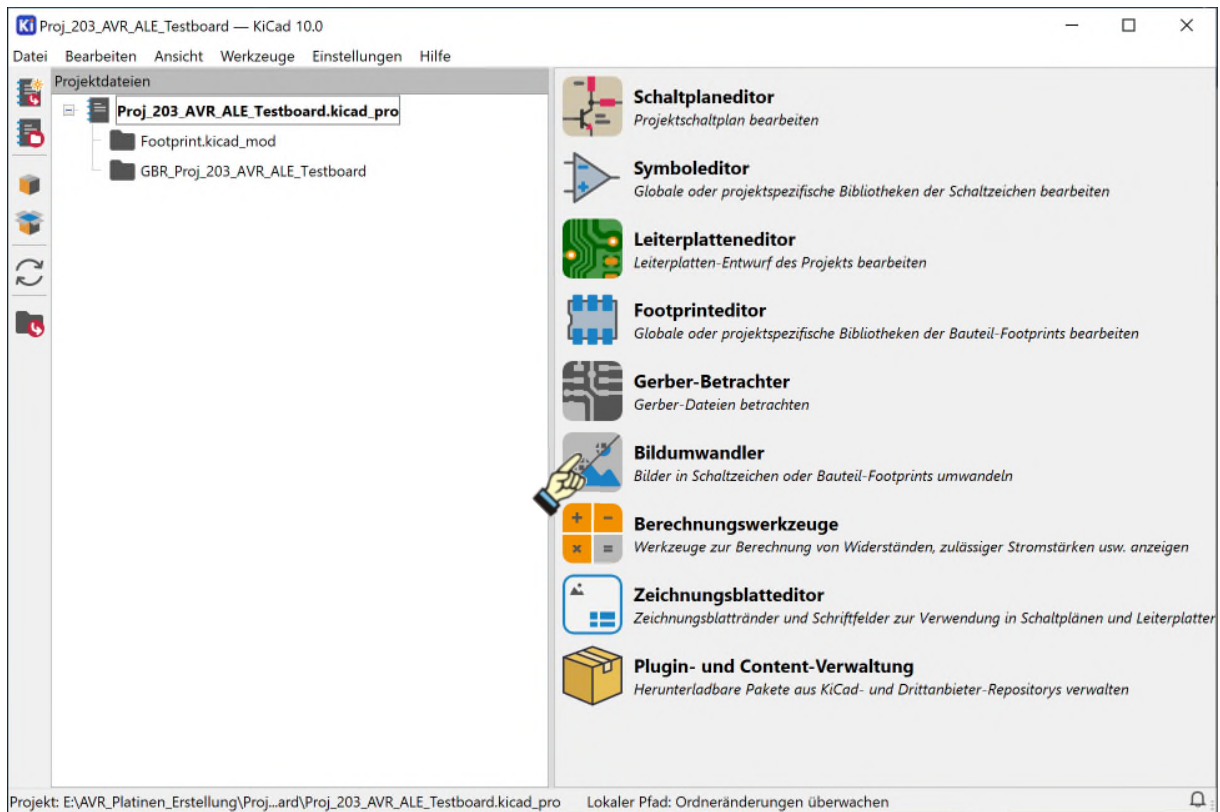
1. **KiCad 10.0** starten, es wird das **KiCad-Hauptfenster** geöffnet (es bleibt auch später für die Prozess-Steuerung im Hintergrund bis zum Schluss geöffnet). Dann wird das folgende vorbereitete Projekt mit **Projekt öffnen ...** eingeleitet:



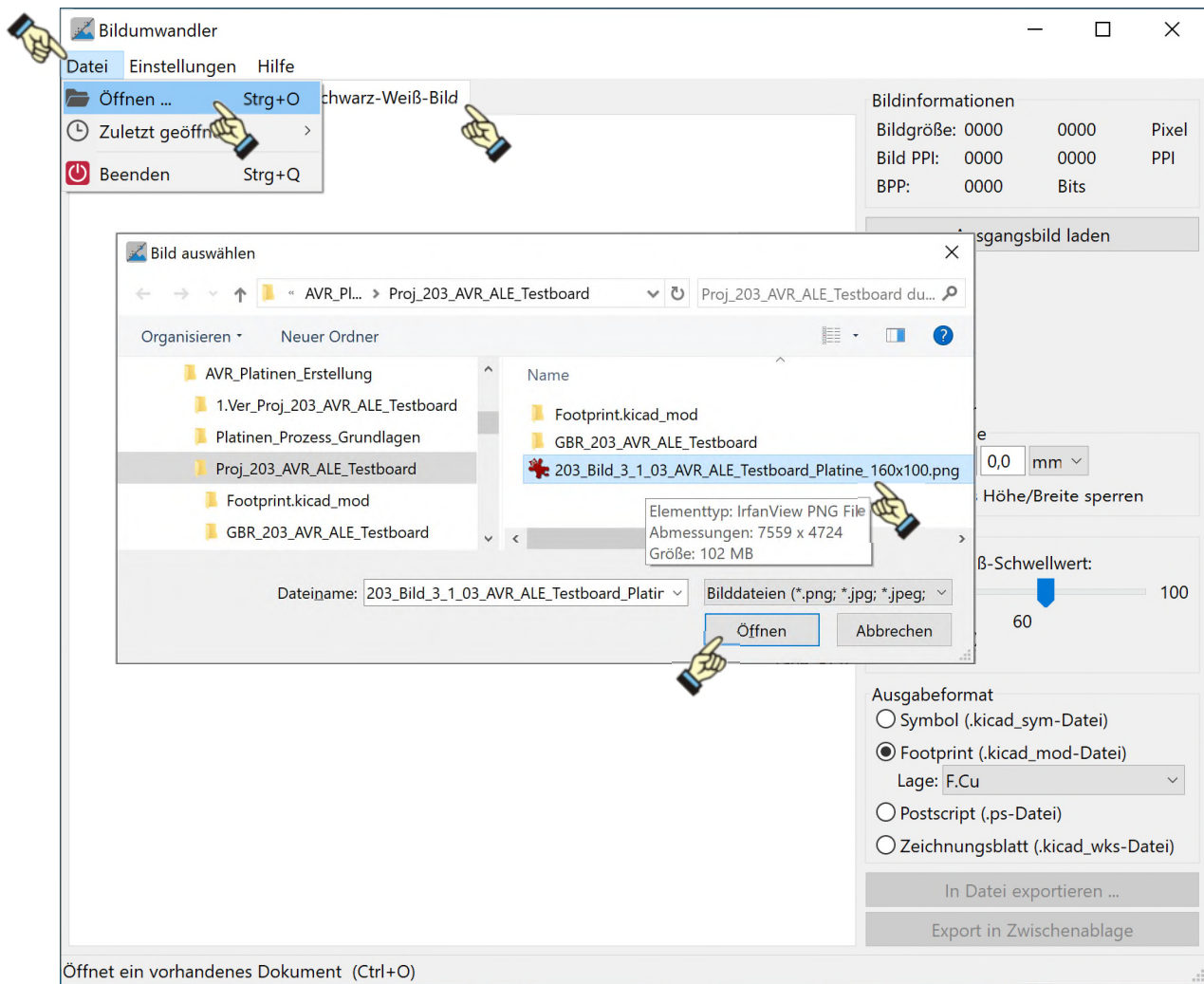
= > Öffnen

04: Bildumwandler: PNG-Bilddatei in Footprint-Datei wandeln

1. Im rechten Hauptmenü den **Bildumwandler**  wählen.



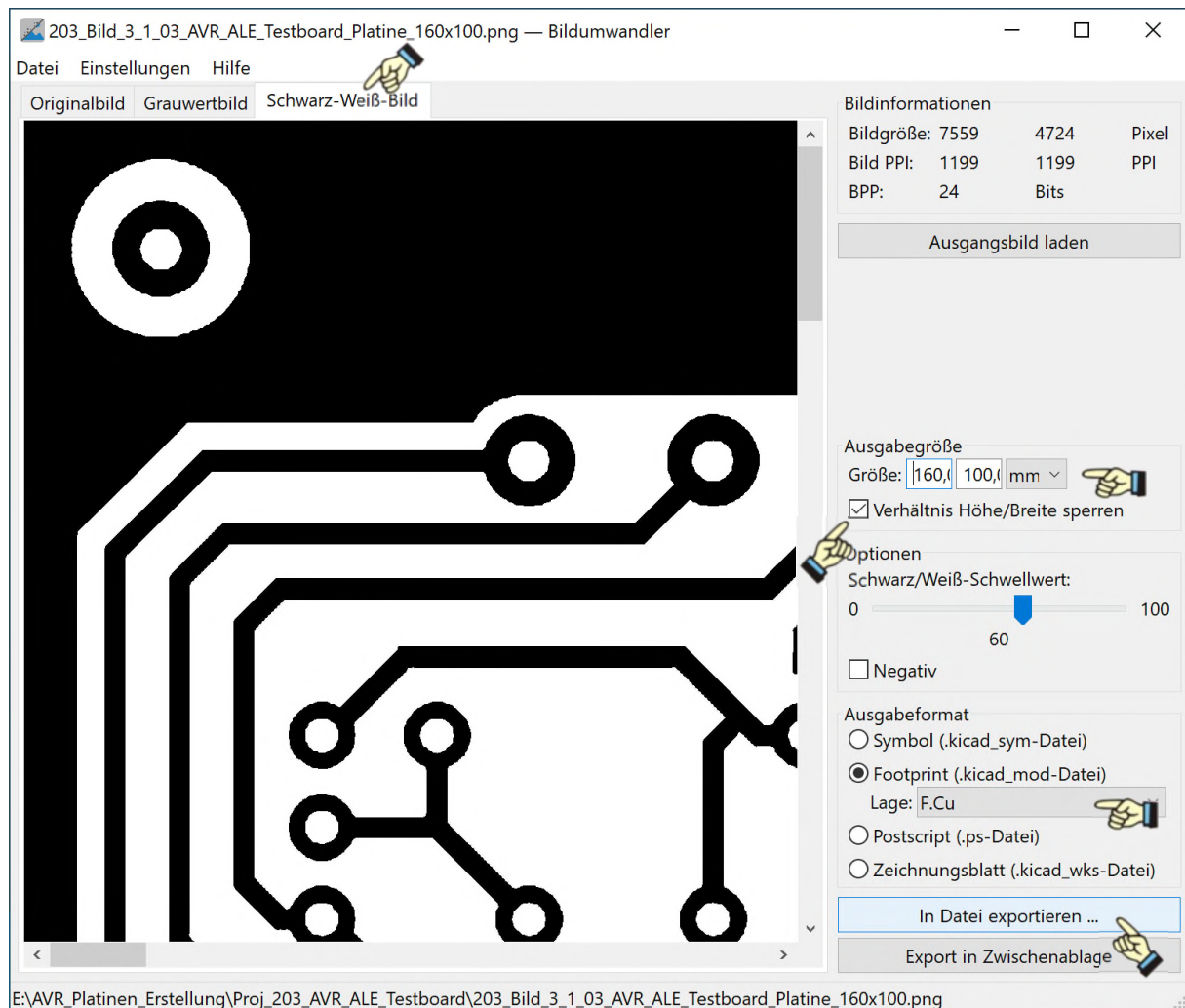
2. Das Fenster **Bildumwandler** wird mit markiertem Reiter **Schwarz-Weiß-Bild** geöffnet.



3. **Datei => Öffnen ...** im Fenster **Bild auswählen** die **Bildname.png-Datei** suchen und auswählen:

203_Bild_3_1_03_AVR_ALE_Testboard_Platine_160x100.png

=> Öffnen



4. Das Layout wird stark vergrößert dargestellt, da es auf 1200 DPI hochskaliert wurde.

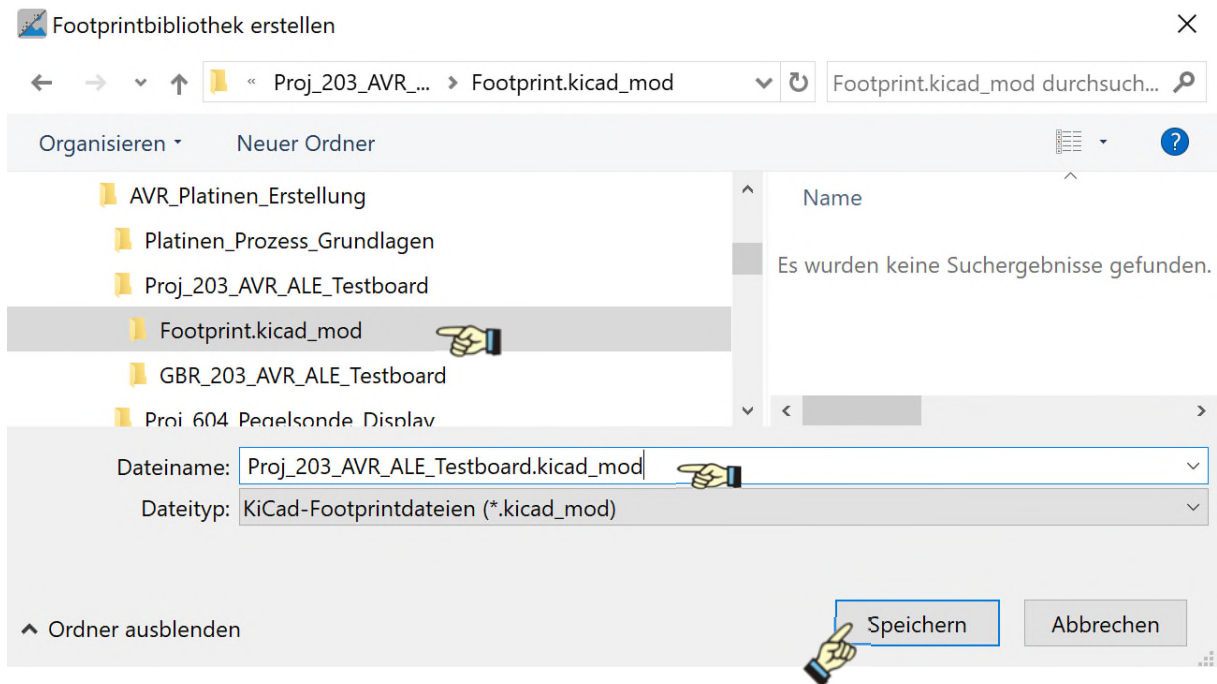
- Im rechten Feld **Ausgabegröße** einstellen: **Größe: 160,0 und 100,0 mm** einstellen (auf Nachkommagröße "Null" achten) und ein Häkchen bei **Verhältnis Höhe/Breite sperren** setzen.
- Optionen** unverändert belassen.
- Rechts unten im **Ausgabeformat** die Option **Footprint (.kicad_mod-Datei)** einstellen und als **Lage:** die Einstellung **F.Cu** wählen.
- Rechts unten **In Datei exportieren ...** wählen und die **Footprint-Datei** als

Proj_203_AVR_ALE_Testboard.kicad_mod

in den für die Footprint-Datei ausgewählten Ordner

Footprint.kicad_mod

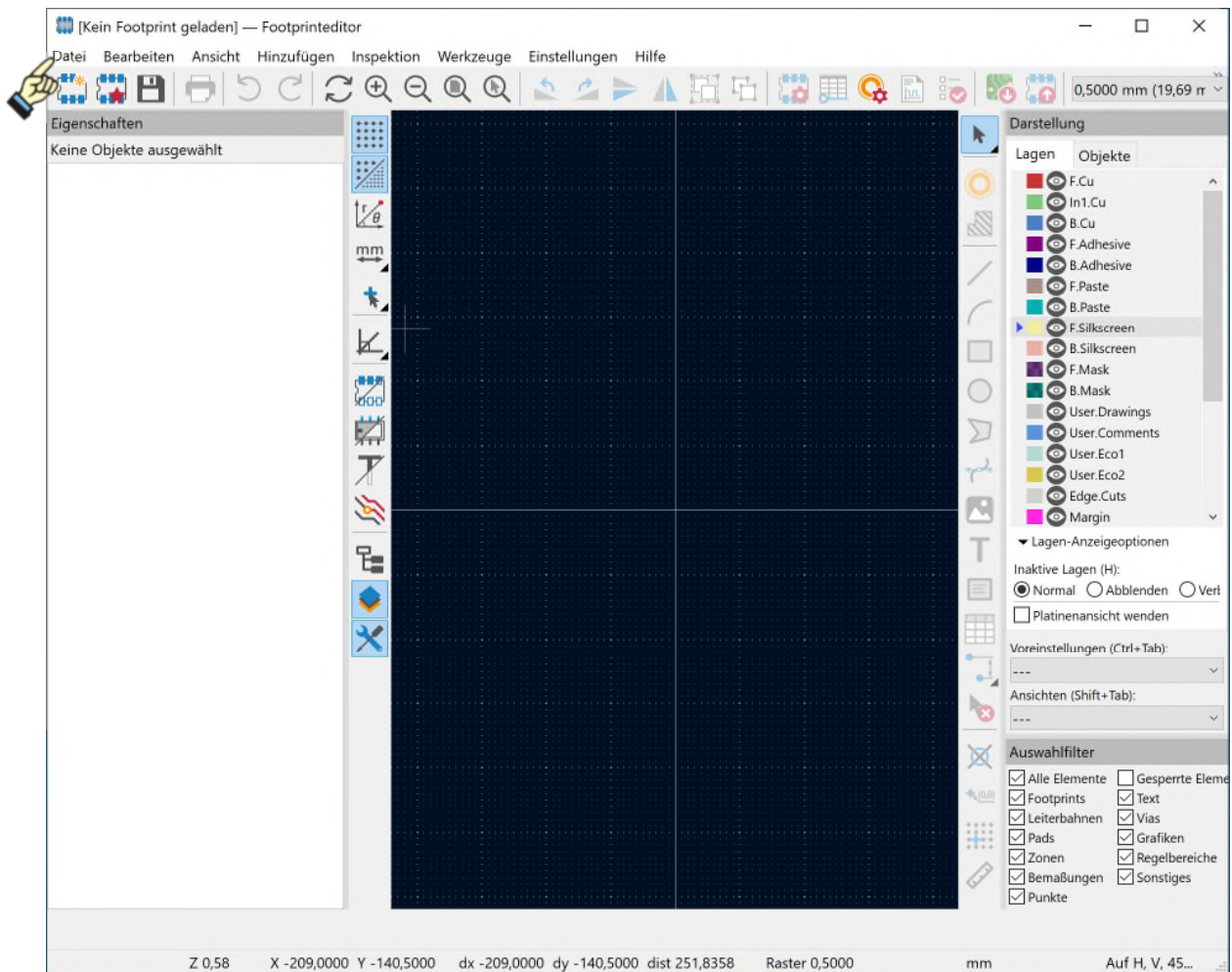
ablegen:



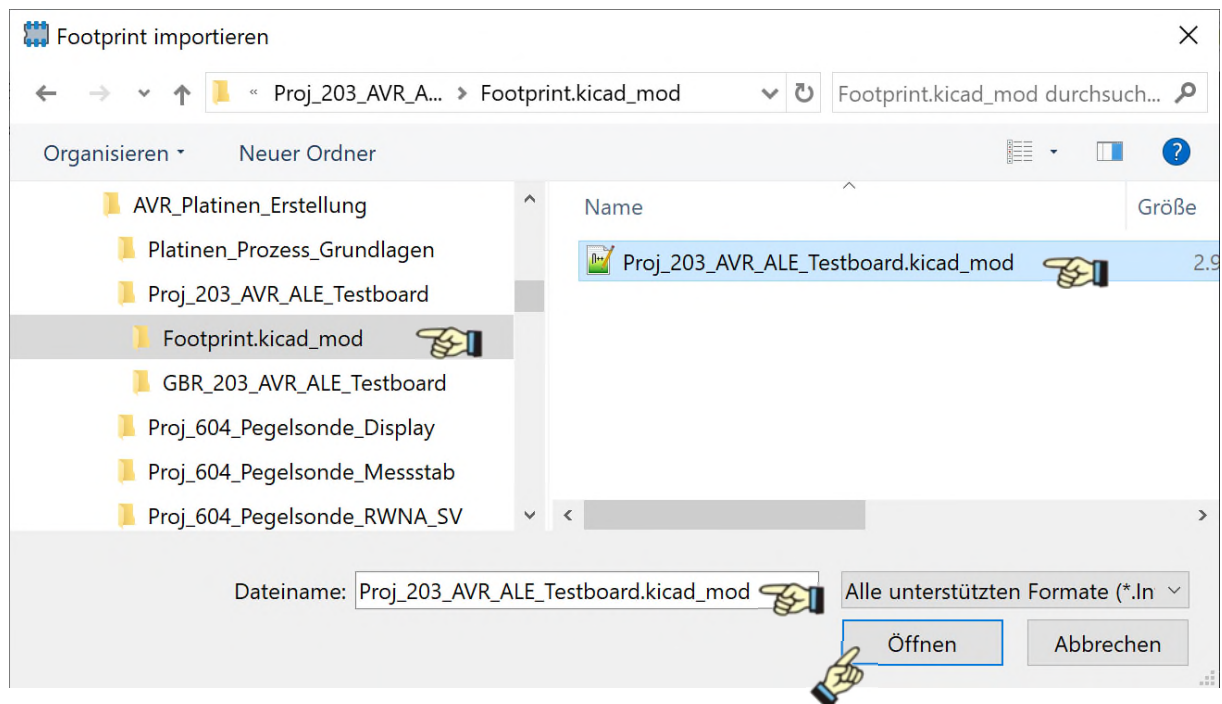
5. Der **Bildumwandler** kann geschlossen werden.

05: Footprinteditor: Footprint-Datei bearbeiten

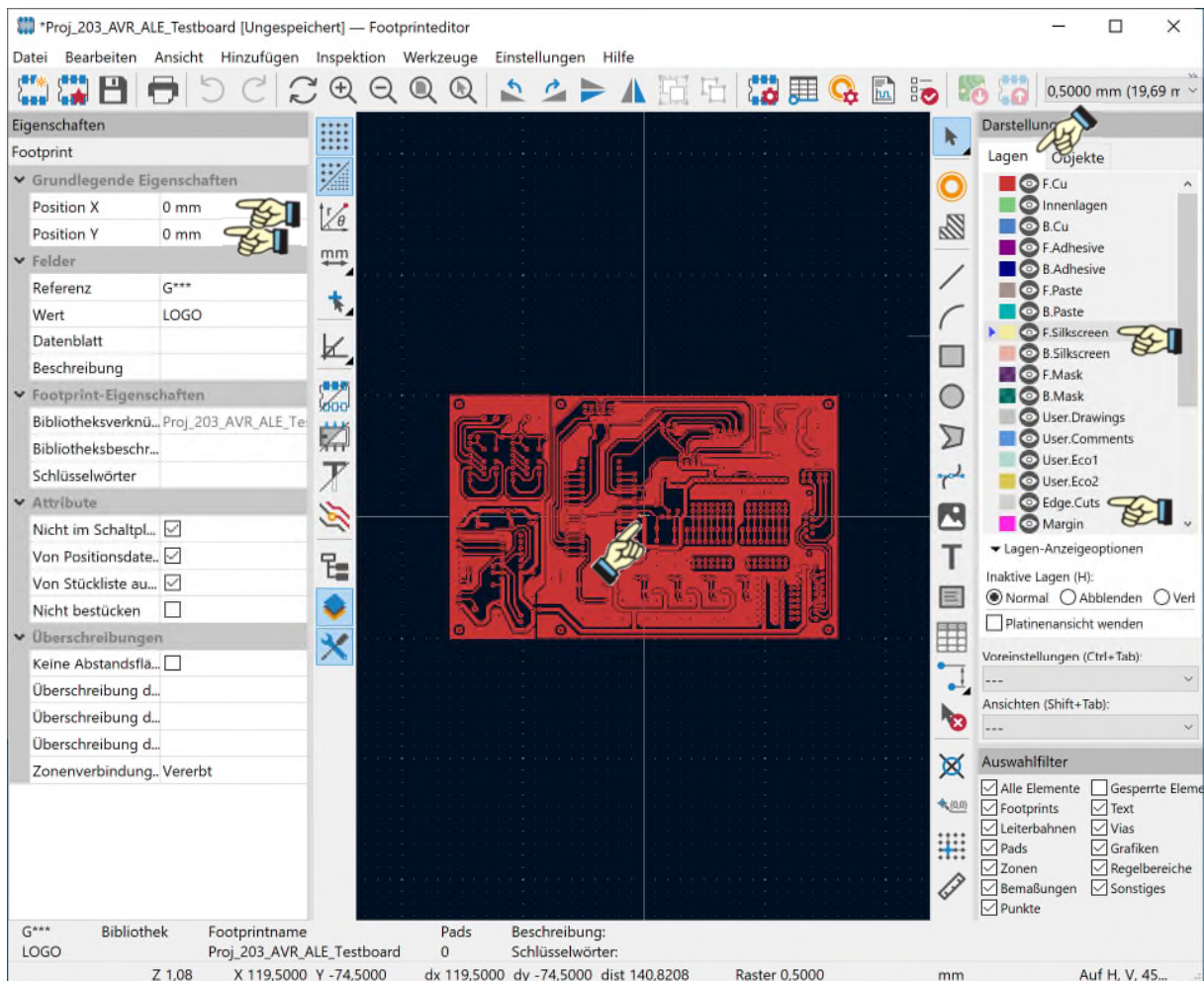
1. Jetzt wird im **KiCad-Hauptfenster** der **Footprinteditor**  aufgerufen. Es öffnet sich das große Fenster des Footprinteditors mit leerem Bildfeld.
2. Footprint-Datei **Proj_203_AVR_ALE_Testboard.kicad_mod** importieren



Durch klicken auf **Datei => Importieren => Footprint ...**



3. ... wird die soeben erstellte Footprint-Datei
Proj_203_AVR_ALE_Testboard.kicad_mod
 importiert:



4. In der rechten Liste **Darstellung**, Reiter **Lagen**, wird der Marker von **F.Silkscreen** (Bestückungsdruck) auf **Edge.Cuts** umgestellt.

5. **Man beachte:** Das Layout ist mittig platziert, d.h. die Mitte des Layouts liegt auf den **Positionen X = 0** und **Y = 0**

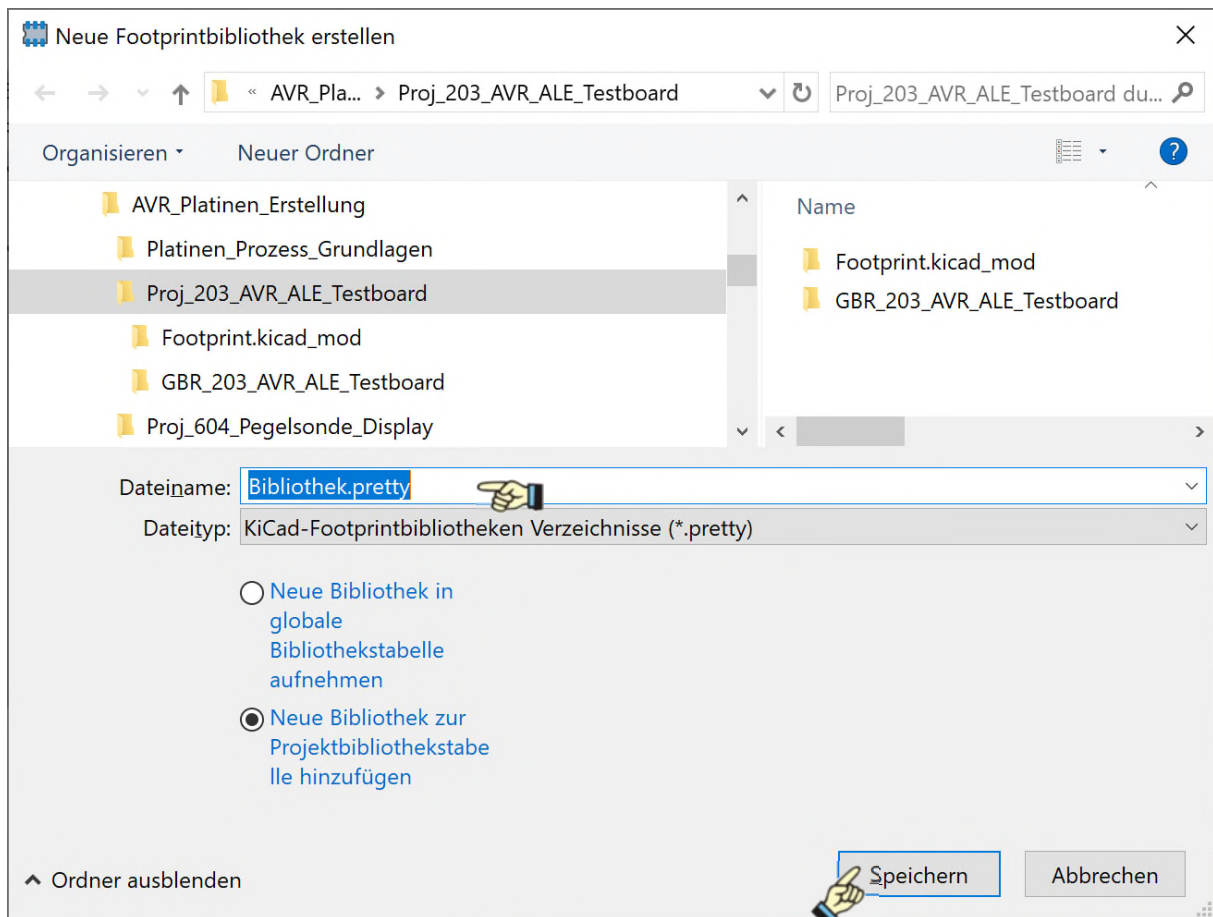
Der **Footprinteditor** kennt bisher ggf. nur die Bibliotheken von vorherigen Projekten. Aber bei diesem Projekt soll die eben erzeugte **Footprint-Datei** jeweils in einem eigenen Bibliotheksordner **Bibliothek.pretty** abgelegt werden. Da dieser nicht im Projekt-Ordner vorinstalliert werden kann, wird er hier mit dem **Footprinteditor** erzeugt:

Datei => Neue Bibliothek ...

6. **KiCad** gibt standardmäßig **Library.pretty** vor. Neu soll für die Footprintbibliothek jedoch für dieses Projekt

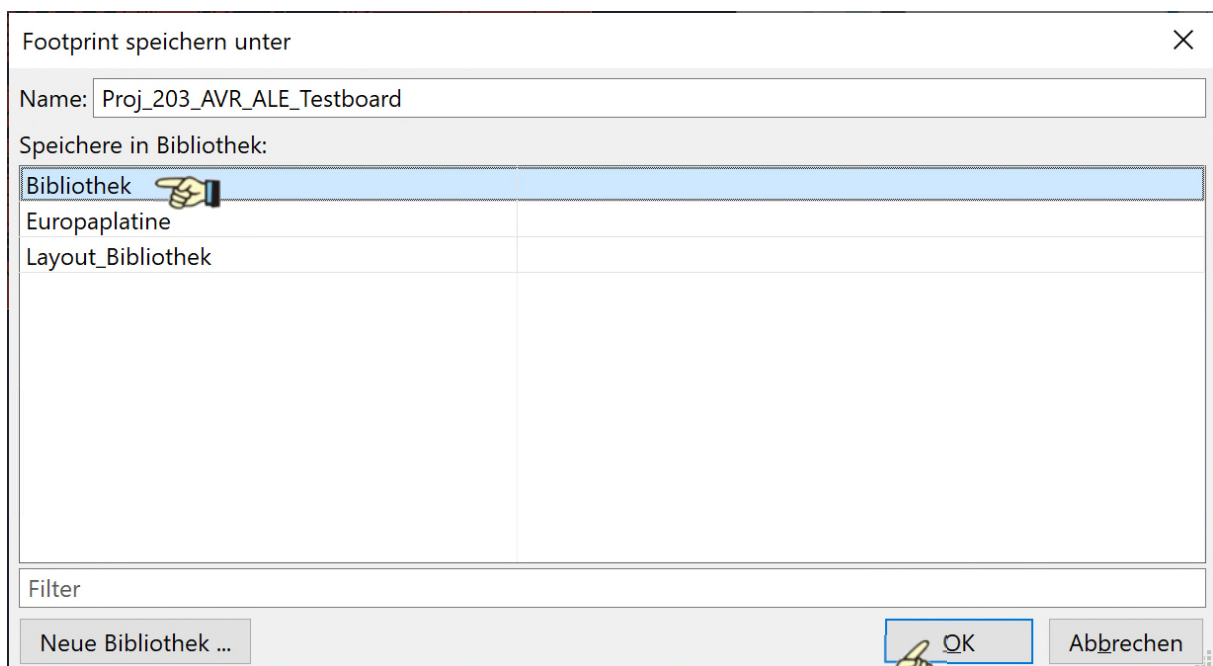
Bibliothek.pretty

gewählt werden. Im Feld **Dateiname:** wird der Ordner umbenannt:



Bibliotheksdatei mit als **Bibliothek.pretty** => **Speichern**

7. Jetzt die Footprintdatei **Proj_203_AVR_ALE_Testboard.kicad_mod** im **Footprinteditor** mit **Datei** => **Speichern unter ...** in der Bibliotheksdatei **Bibliothek.pretty** sichern:



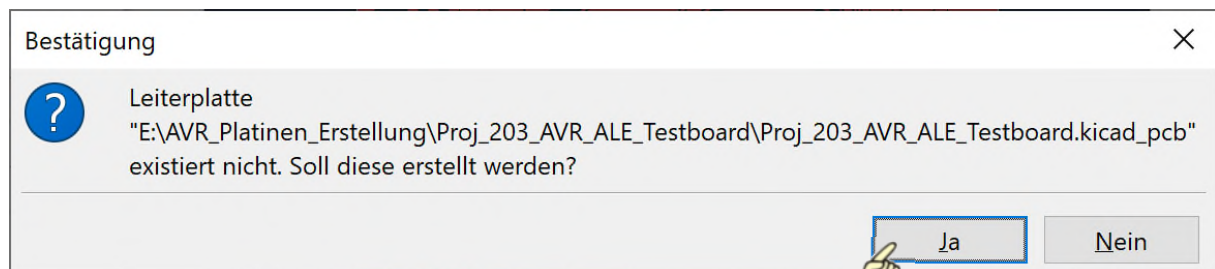
Speichern in **Bibliothek** (das ist die **Bibliothek.pretty**) => **OK**

06: Leiterplatteneditor: Footprint-Datei bearbeiten

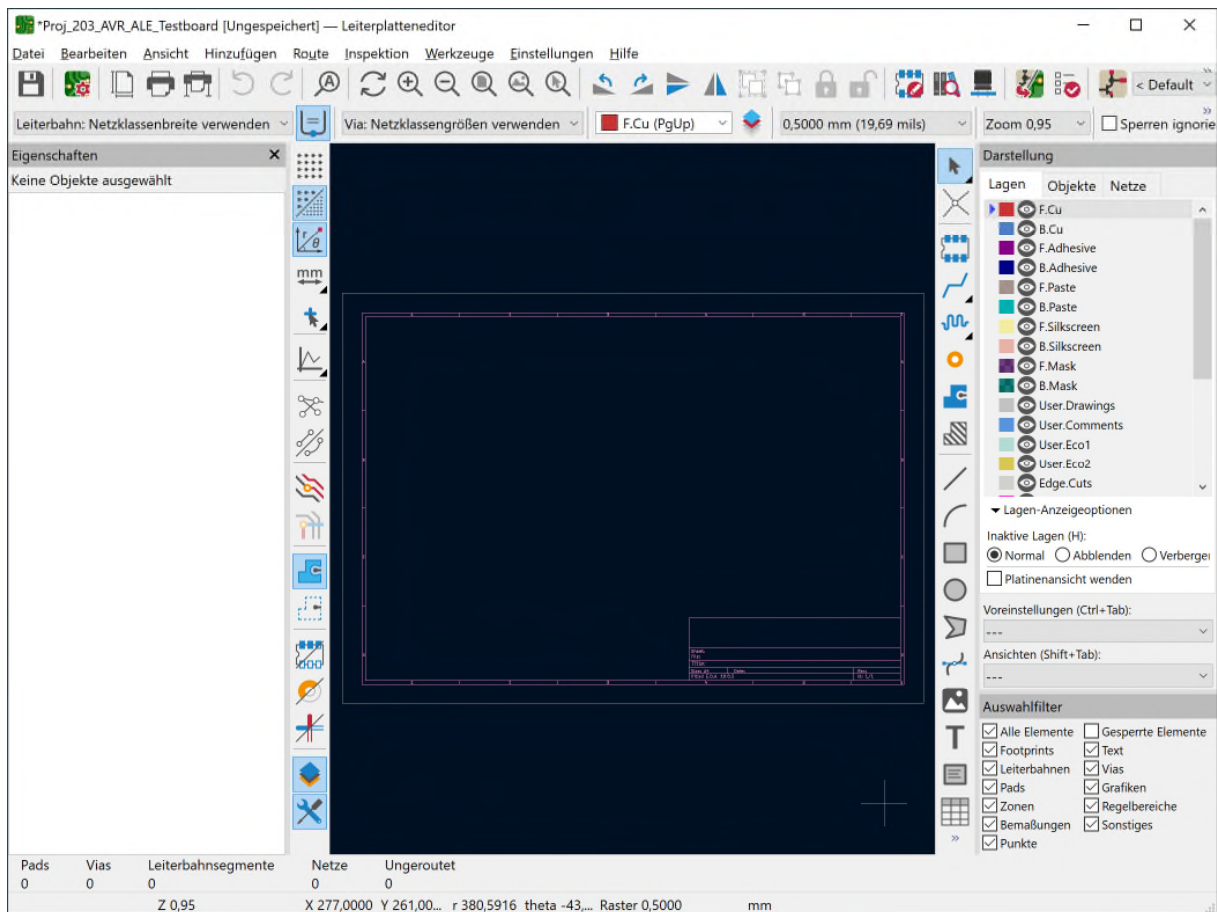
1. Aufrufen des **Leiterplatteneditors** aus dem **KiCad-Hauptfenster**



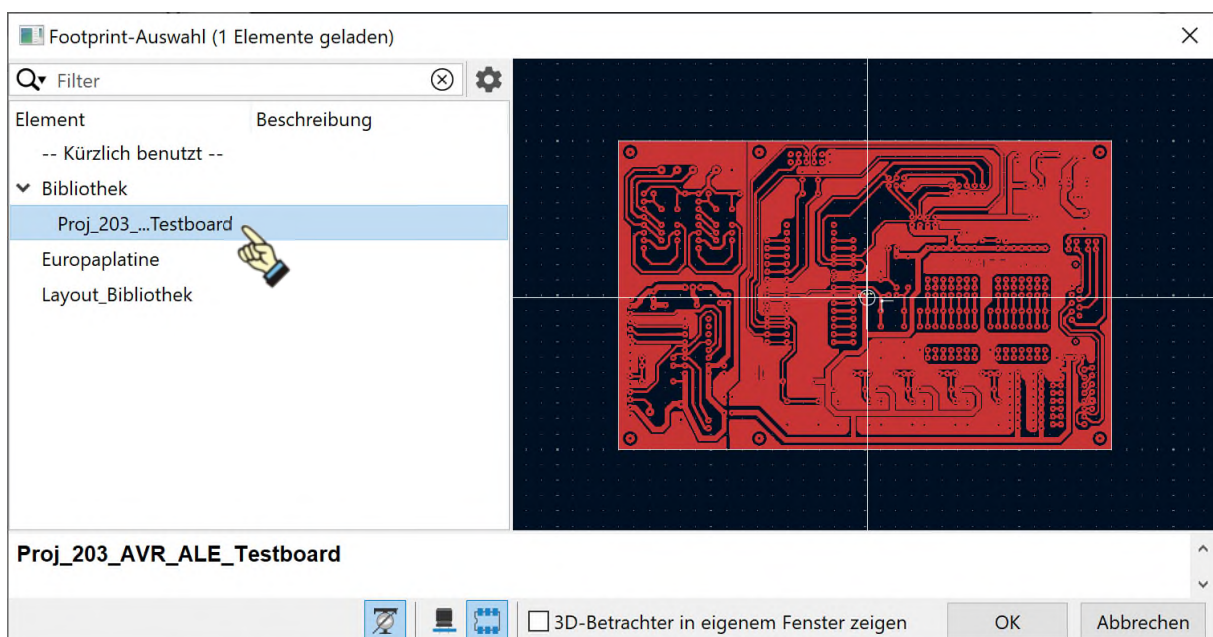
2. Wie man im linken Bereich sehen kann, existiert noch kein sog. **Printed Circuit Board**, d.h. keine Datei mit der Endung **.kicad_pcb**. Beim erstmaligen Aufrufen des **Leiterplatteneditors** wird sie verlangt:



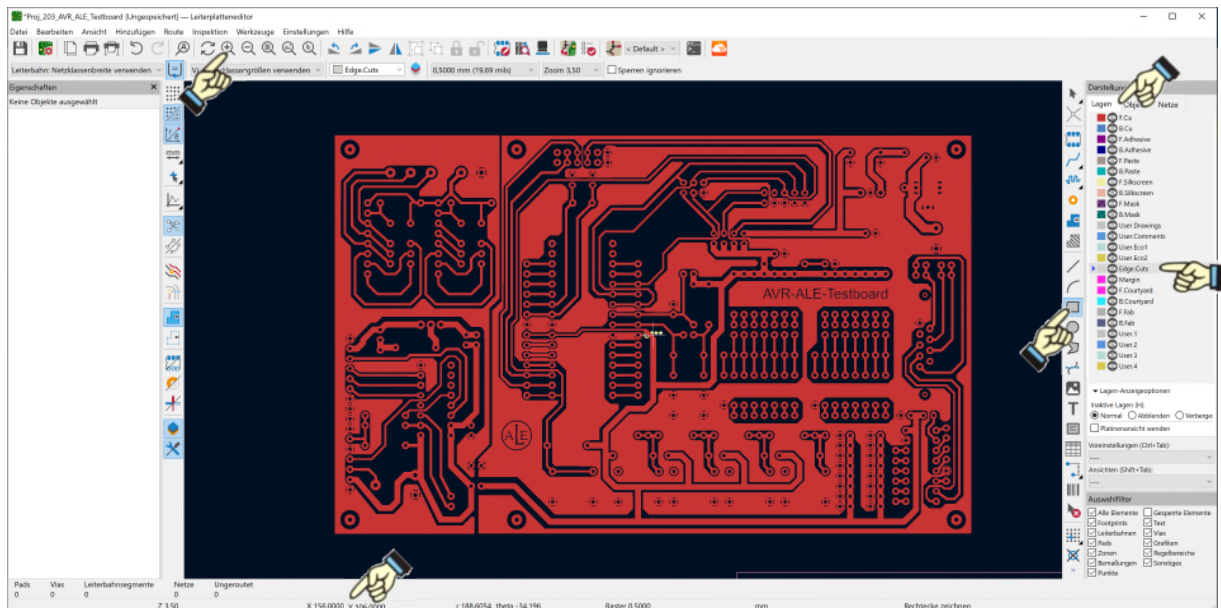
3. Der **Leiterplatteneditor** wird mit einer leeren, schwarzen Arbeitsfläche gestartet. => **Ja**
4. Der **Footprinteditor** wird nicht mehr gebraucht und kann geschlossen werden.



5. Die **Footprint-Datei** **Proj_AVR_ALE_Testboard.kicad_mod** wird mit **=> Hinzufügen => Footprint platzieren** (oder Drücken der Taste **A** auf der Tastatur) aufgerufen. Das **Platzieren** ist, wie man in den folgenden Schritten sehen kann, sehr wichtig. Es öffnet sich die **Footprint-Auswahl**, die beim Anklicken des **Bibliothekseintrags** **Proj_203...Testboard** die **Footprint-Datei** anzeigt:



6. Mit einem **Doppelklick** auf den Bibliothekseintrag **Proj_203...Testboard** "hängt" das Layout am Mauszeiger und soll nun im Arbeitsfensters vom **Leiterplatteneditor** an einem definierten Ort platziert werden. Zunächst wird der Ort mit seinen Koordininnaten **X** und **Y** im **Leiterplatteneditor** in der untersten Zeile angezeigt. Hat man z.B. die genauen Werte für den Mittelpunkt des Layouts mit **X 156,0000** und **Y 106,0000** mit ruhiger Hand erreicht, wird mit einem **Linksklick** - ohne zu Zittern - das Layout dort abgesetzt. Für die weiteren Schritte ist es sinnvoll, die Ansicht des Layouts mit dem Plus-Werkzeug **Hinein zoomen** zu vergrößern:



Bedeutung der "Handzeiger"

Links Unten: Der Zeiger zeigt auf die genauen Koordinaten-Werte für den Mittelpunkt des Layouts: **X 156,0000** und **Y 106,0000**. Im Prinzip sind die absoluten Werte nicht von Bedeutung. Aber je genauer sie gewählt wurden, um so leichter und genauer lassen sich die nachfolgenden Masken platzieren. Das Layout mit der Größe **160 x 100 mm** hat bezogen auf die oben gewählte Lage genau folgende Eckpunkt-Koordinaten:

Oberer linker Eckpunkt: **X 76,0000 und Y 56,0000**

Unterer rechter Eckpunkt: **X 236,0000 und Y 156,0000**

Entsprechend werden die Koordinaten für die Rechtecke der Masken genau so platziert.

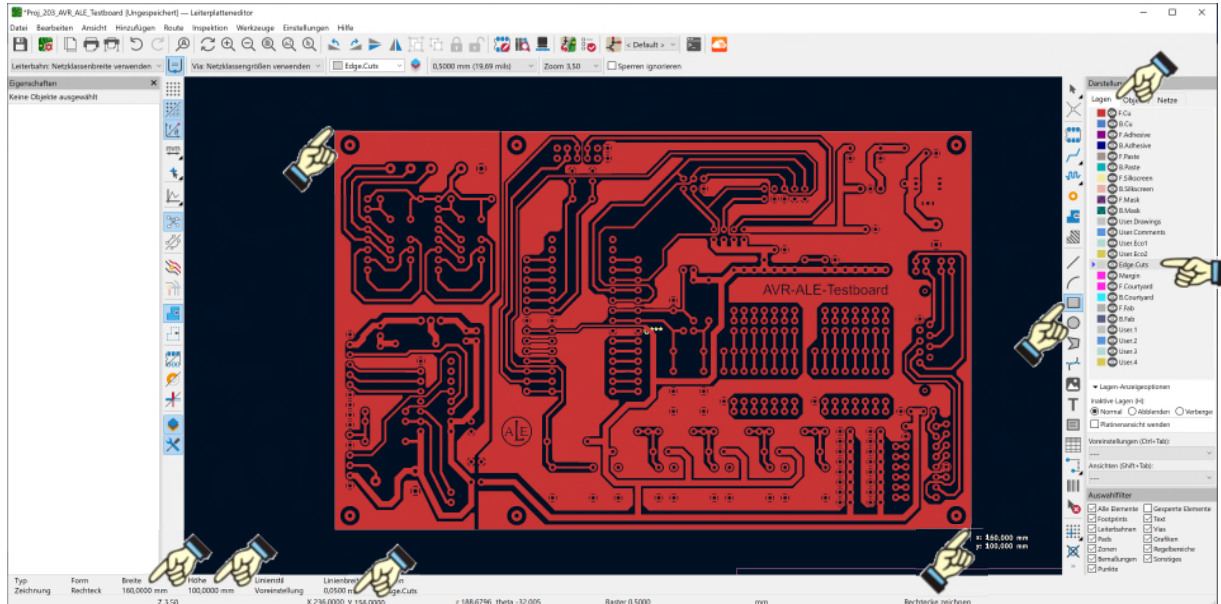
Links Oben: Der Zeiger zeigt auf die "Vergrößerungslupe", um die Darstellung des Layouts zur besseren Einstellung der Koordinaten-Werte zu vergrößern.

Rechts Außen: Die beiden Zeiger verweisen auf die Einstellung für den Layout-Begrenzungsrahmen (**Edge.Cuts**).

Rechts Mitte: Der Zeiger zeigt auf das Rechteck-Werkzeug, das für die Erstellung des Layout-Begrenzungsrahmens (**Edge.Cuts**) benötigt wird.

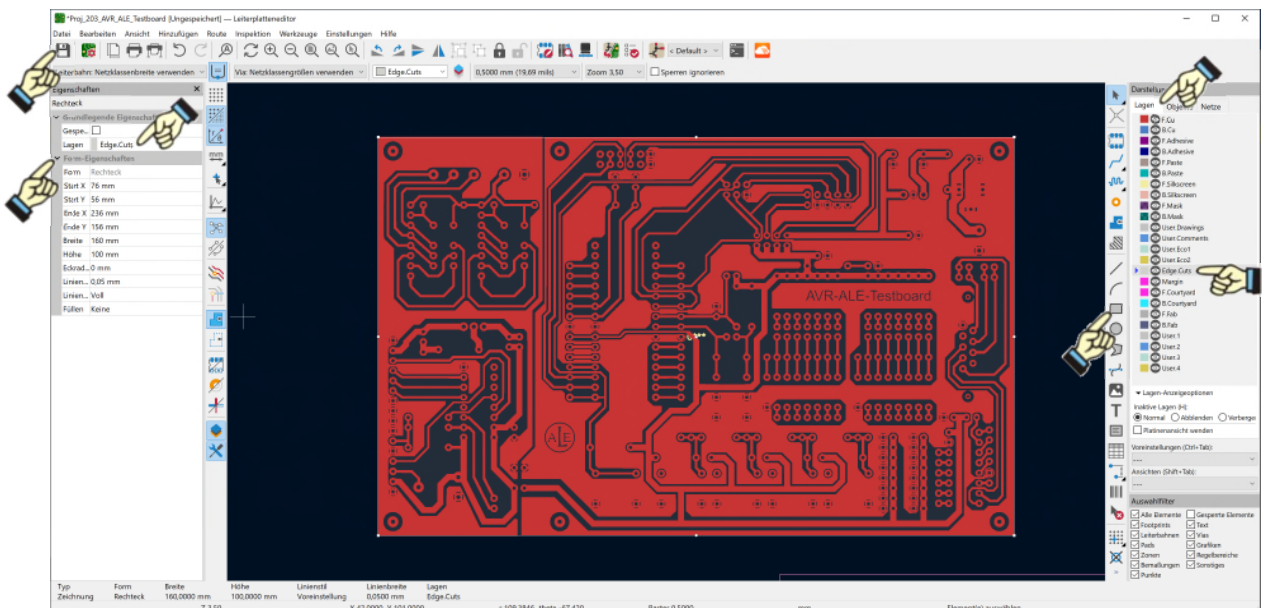
06.1: Layout-Begrenzungsrahmen festlegen

1. **Lagen** ändern in **Edge.Cuts**, Werkzeug **Rechteck zeichnen** einstellen (siehe oben)
2. Exaktes Rechteck um das Layout (Footprint) legen:

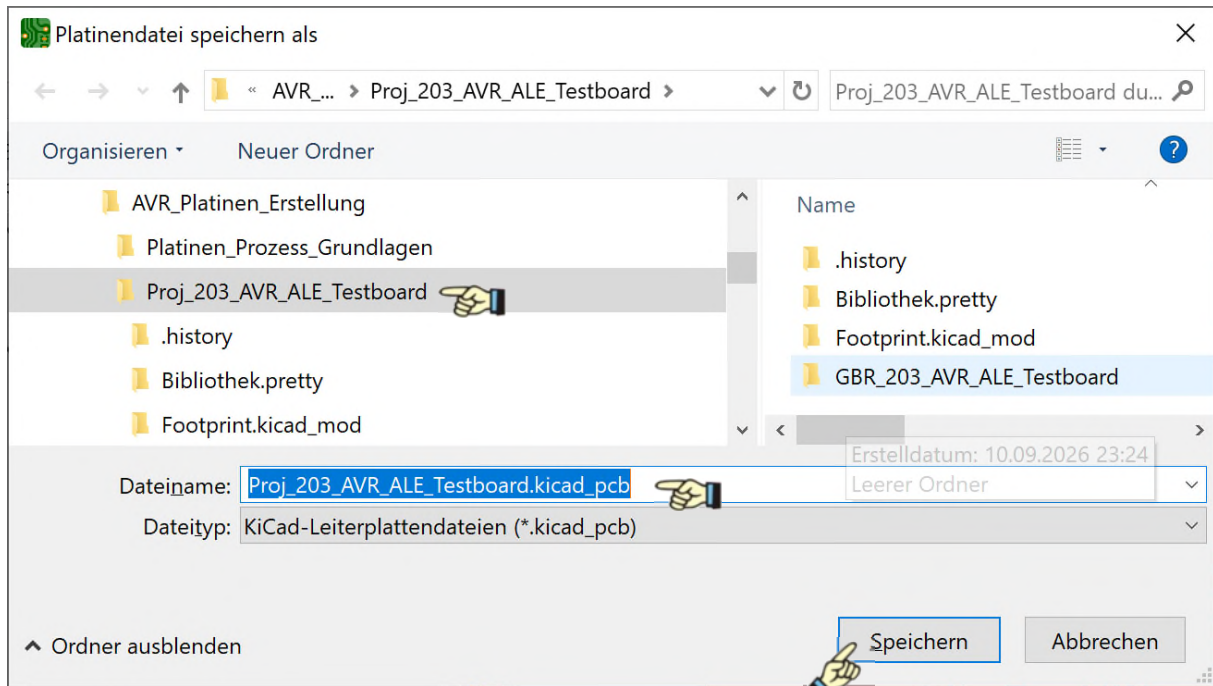


- a. Mit der Maus an die **obere linke Ecke** des Layouts fahren. Erst wenn die Koordinaten bei der Mausbewegung genau **X 76,0000** und **Y 56,0000** auf der untersten Zeile des **Leiterplatteneditors** angezeigt werden, dann mit einem **Linksklick** den Punkt fixieren.
- b. Nun mit der Maus (dem Rechteck) auf die **untere rechte Ecke** des Layouts zusteuern. Erst wenn die Koordinaten genau **X 236,0000** und **Y 156,0000** (wieder auf der untersten Zeile des **Leiterplatteneditors** ablesbar) anzeigen, dann mit einem **Linksklick** den Punkt - und damit das Rechteck - fixieren.

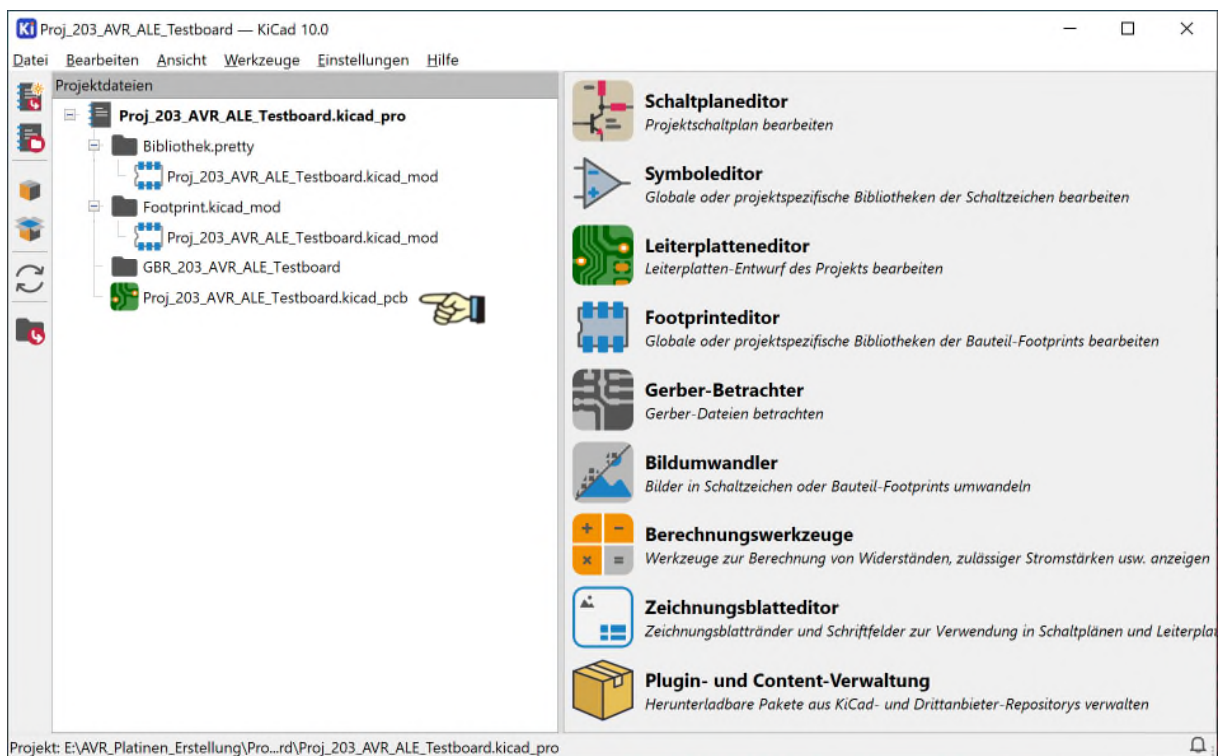
3. Mit der Taste **Esc** wird das Werkzeug **Rechteck zeichnen** beendet.



4. In der linken Tabelle **Form-Eigenschaften** kann man die Koordinaten prüfen
5. Mit **Datei => Kopie speichern ...** wird die Datei **Proj_203_AVR_ALE_Testboard.kicad_pcb** mit dem **Platinen-Rahmen (Edge.Cuts)** in dem **Printed Circuit Board** gesichert:



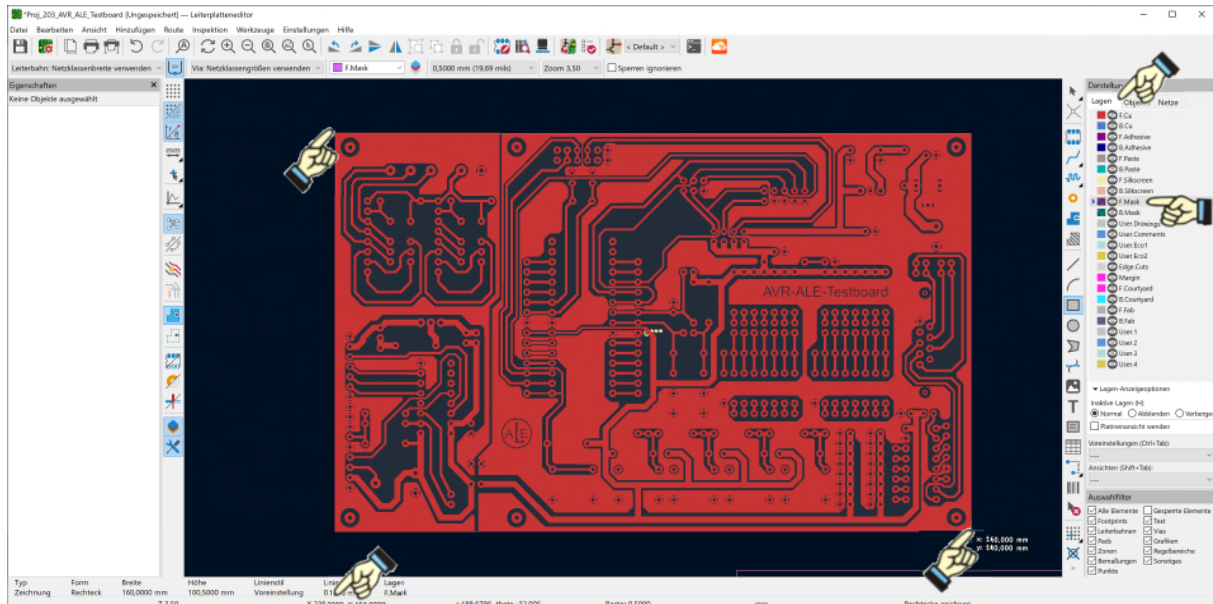
=> Speichern



6. Ein Klick in das schwarze Leerfeld bereitet den **Leiterplatteneditor** auf den nächsten Arbeitsschritt vor.

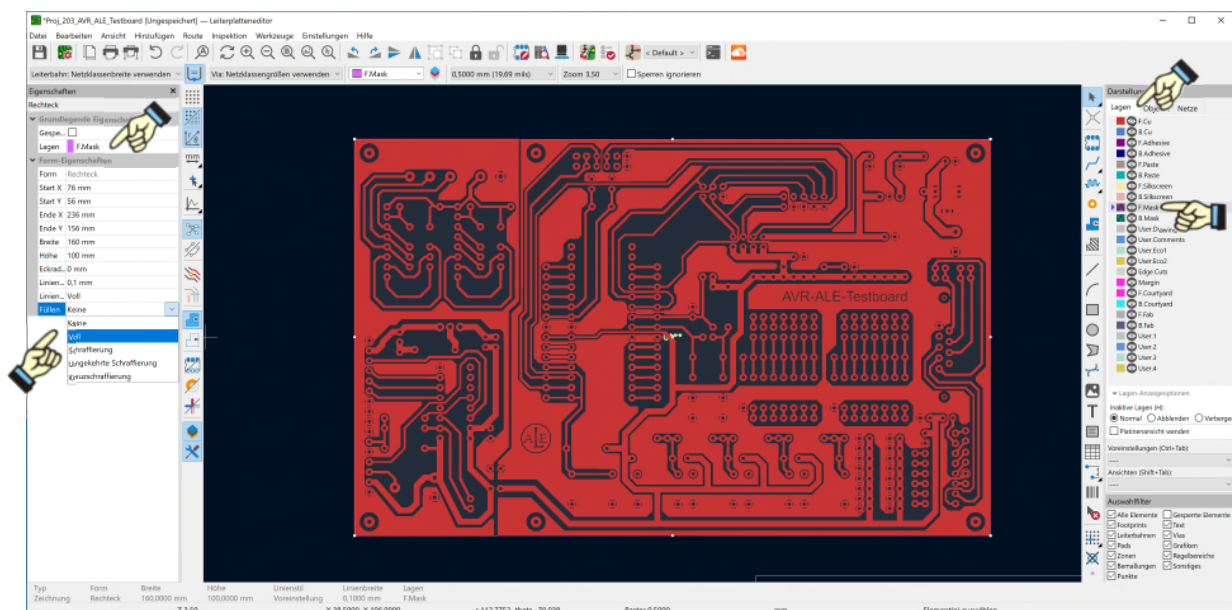
06.2: Lötstopppmaske für Vorderseite (Frontsite) festlegen

1. Die Auswahl **Edge.Cuts** wird aufgehoben und **Lagen** auf **F.Mask** umgestellt. Das Werkzeug **Rechteck zeichnen** wird wieder markiert.
2. Exaktes Rechteck um das Layout (Footprint) legen (vergl. Punkt **06.1**):

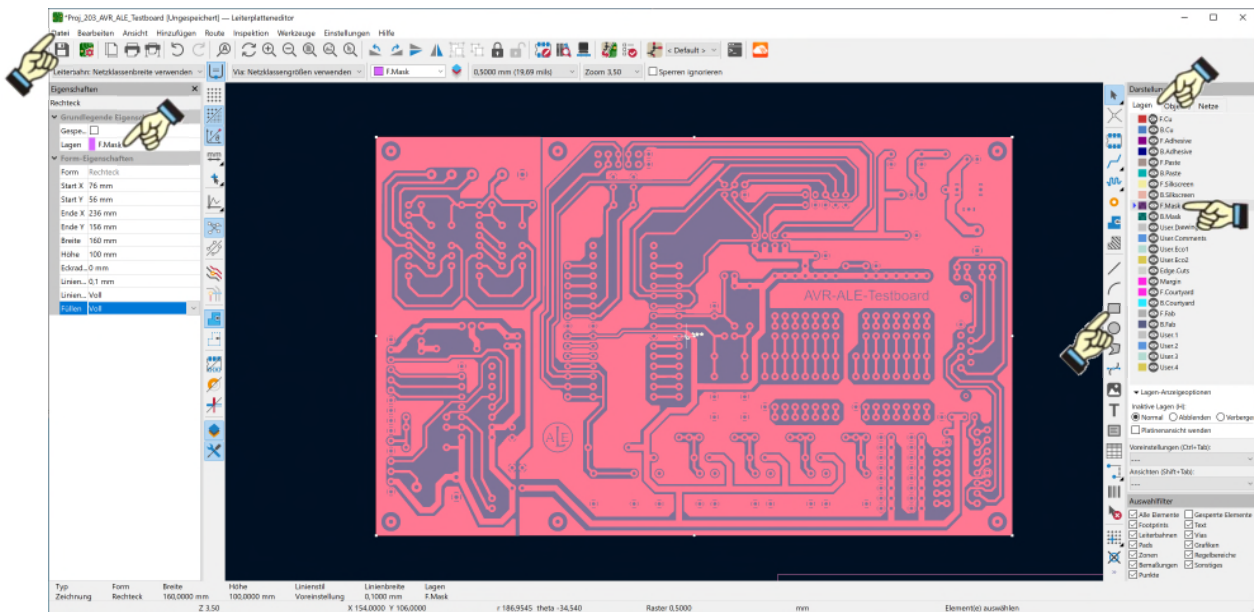


- a. Mit der Maus an die **obere linke Ecke** des Layouts fahren. Erst wenn die Koordinaten bei der Mausbewegung genau **X 76,0000** und **Y 56,0000** auf der untersten Zeile des **Leiterplatteneditors** angezeigt werden, dann mit einem **Linksklick** den Punkt fixieren.
- b. Nun mit der Maus (dem Rechteck) auf die **untere rechte Ecke** des Layouts zusteuern. Erst wenn die Koordinaten genau **X 236,0000** und **Y 156,0000** (wieder auf der untersten Zeile des **Leiterplatteneditors** ablesbar) anzeigen, dann mit einem **Linksklick** den Punkt - und damit das Rechteck - fixieren.

3. Mit der Taste **Esc** wird das Werkzeug **Rechteck zeichnen** beendet.



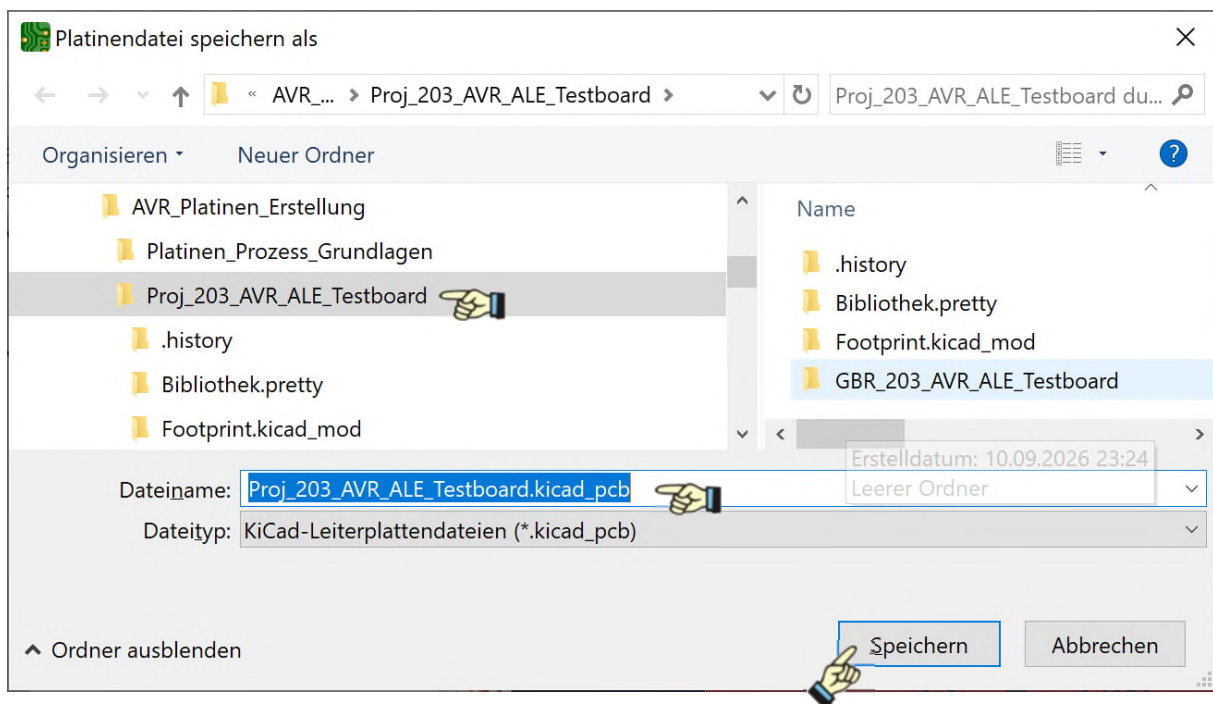
4. Mit **Füllen** => **Voll** wird das Rechteck der Maske **F.Mask** mit Farbe gefüllt



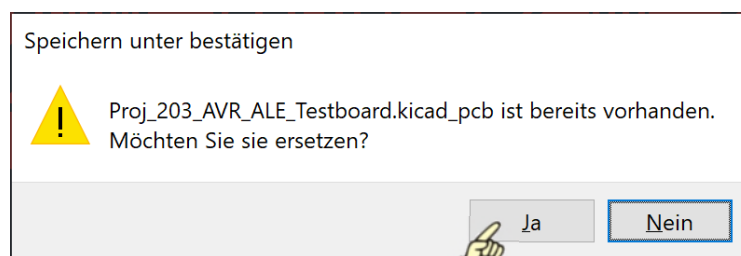
5. Mit **Datei** => **Kopie speichern ...** wird die Datei

Proj_203_AVR_ALE_Testboard.kicad_pcb

mit der **Lötstopmmaske (F.Mask)** in dem **Printed Circuit Board** gesichert:



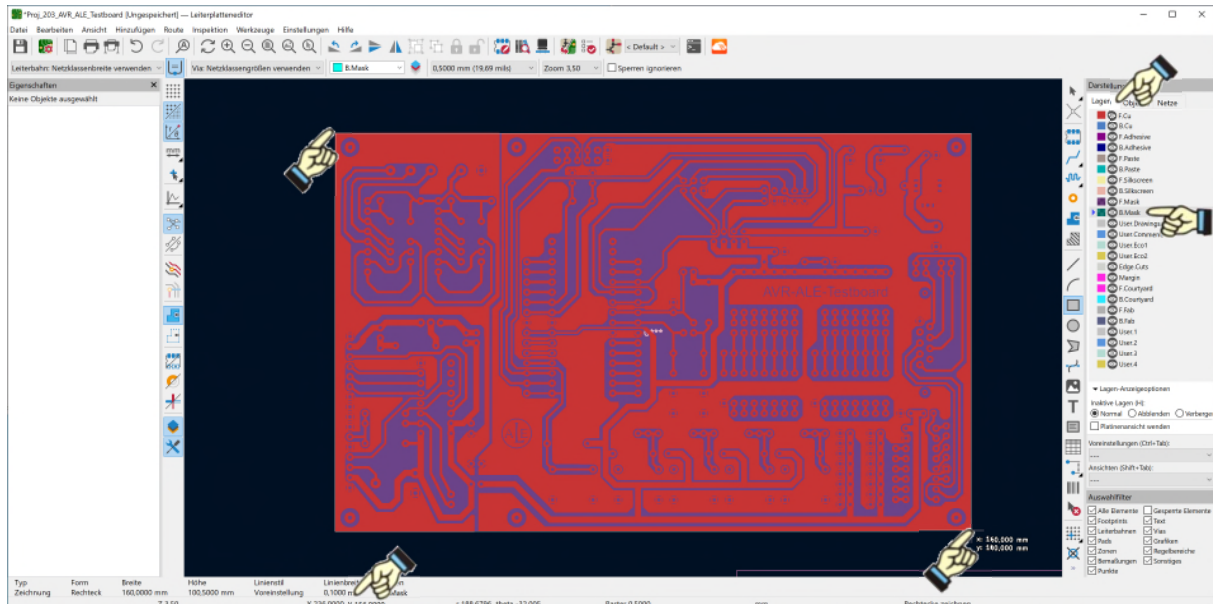
=> **Speichern**



=> **Ja**

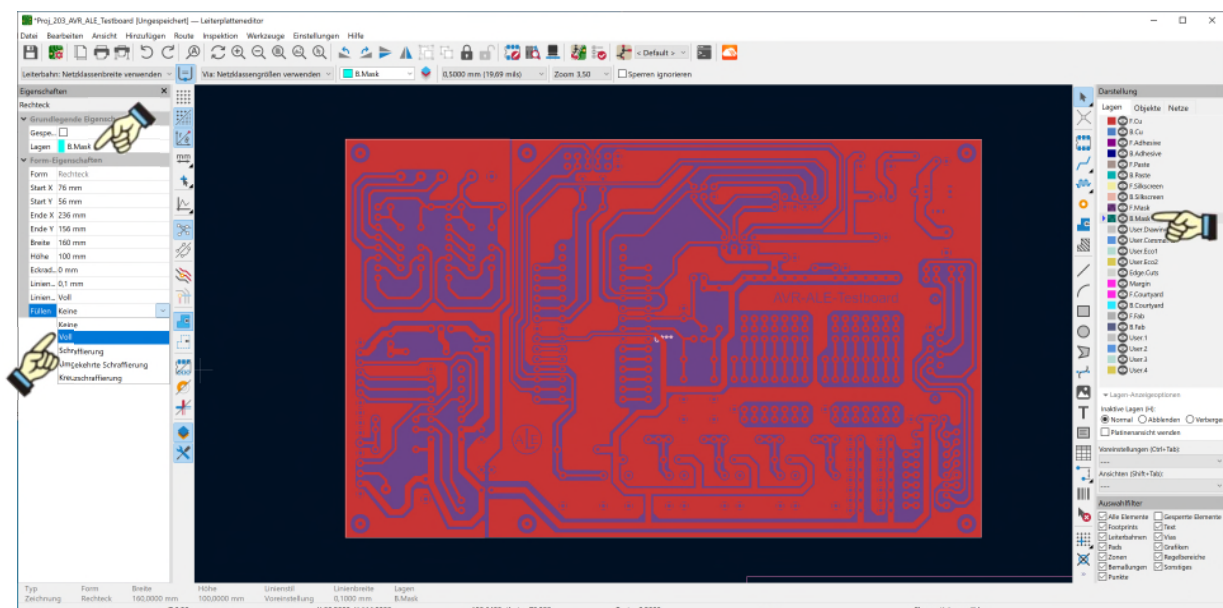
06.3: Lötstopppmaske für Rückseite (Backsite) festlegen

2. Die Auswahl **F.Mask** wird aufgehoben und **Lagen** auf **B.Mask** umgestellt. Das Werkzeug **Rechteck zeichnen** wird wieder markiert.
6. Exaktes Rechteck um das Layout (Footprint) legen (vergl. Punkt **06.1**):

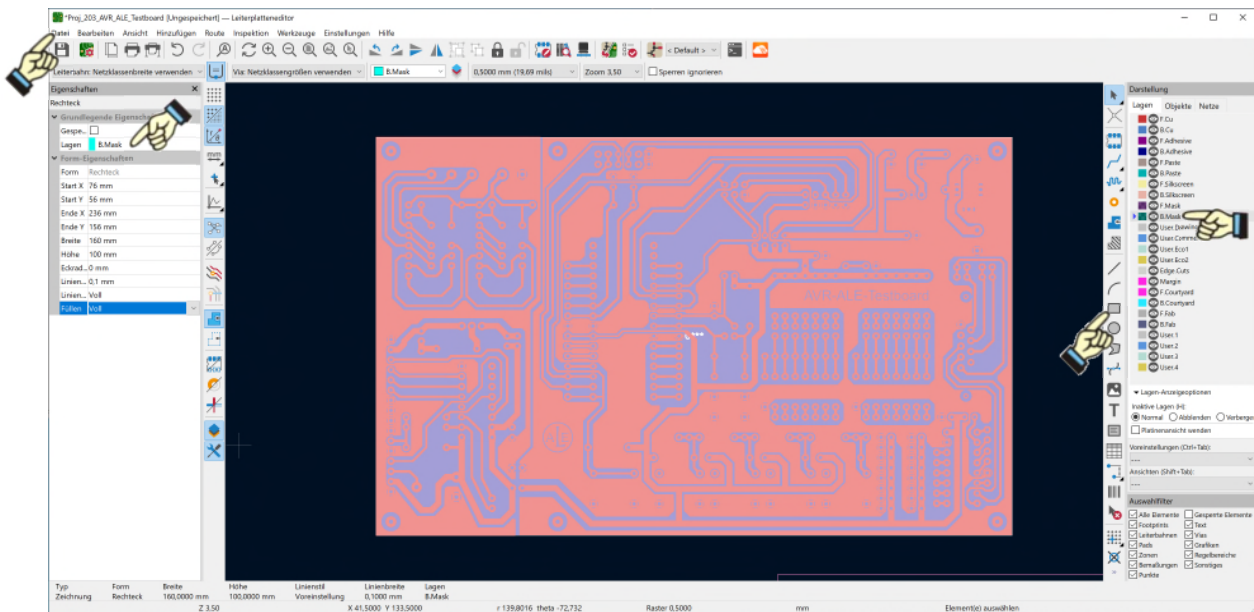


- a. Mit der Maus an die **obere linke Ecke** des Layouts fahren. Erst wenn die Koordinaten bei der Mausbewegung genau **X 76,0000** und **Y 56,0000** auf der untersten Zeile des **Leiterplatteneditors** angezeigt werden, dann mit einem **Linksklick** den Punkt fixieren.
- b. Nun mit der Maus (dem Rechteck) auf die **untere rechte Ecke** des Layouts zusteuern. Erst wenn die Koordinaten genau **X 236,0000** und **Y 156,0000** (wieder auf der untersten Zeile des **Leiterplatteneditors** ablesbar) anzeigen, dann mit einem **Linksklick** den Punkt - und damit das Rechteck - fixieren.

7. Mit der Taste **Esc** wird das Werkzeug **Rechteck zeichnen** beendet.



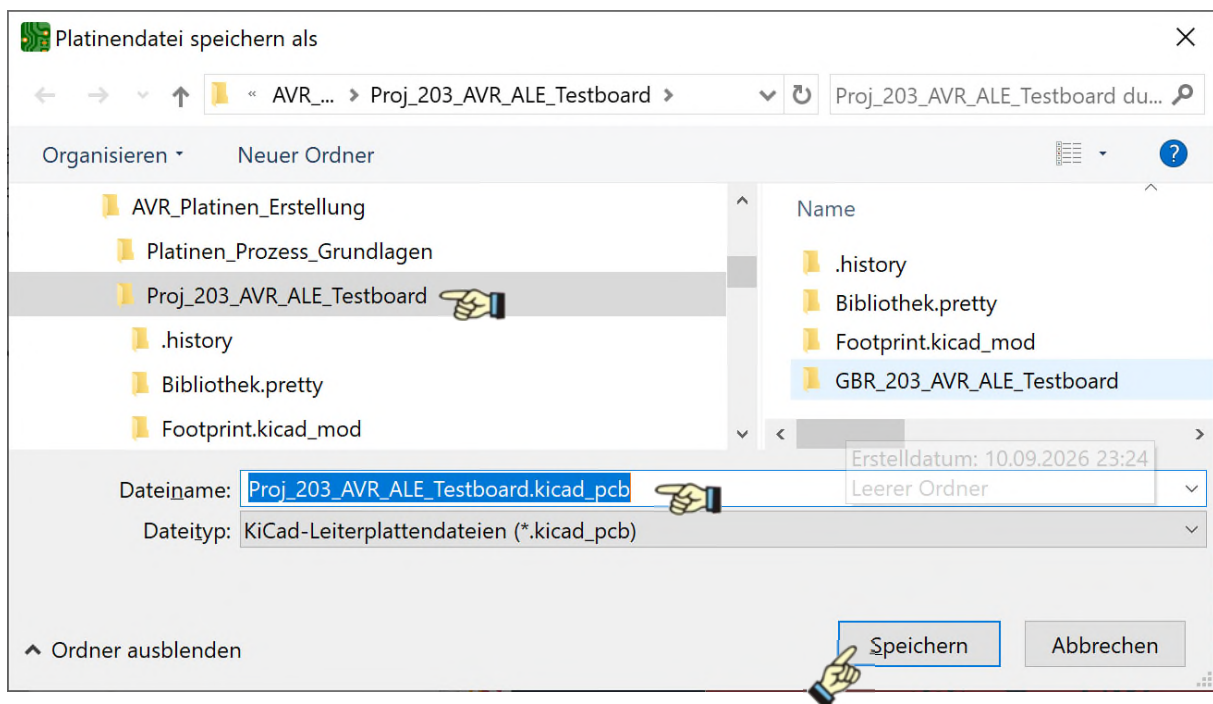
8. Mit **Füllen** => **Voll** wird das Rechteck der Maske **F.Mask** mit Farbe gefüllt



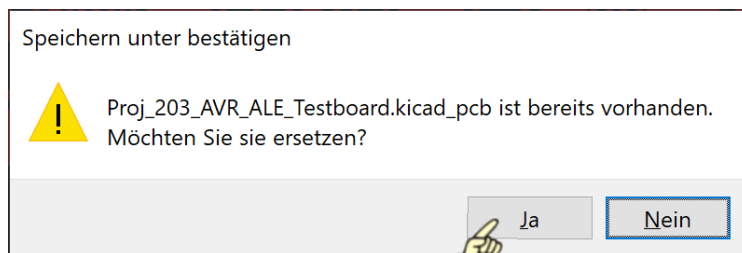
9. Mit **Datei** => **Kopie speichern ...** wird die Datei

Proj_203_AVR_ALE_Testboard.kicad_pcb

mit der **Lötstopmmaske (B.Mask)** in dem **Printed Circuit Board** gesichert:



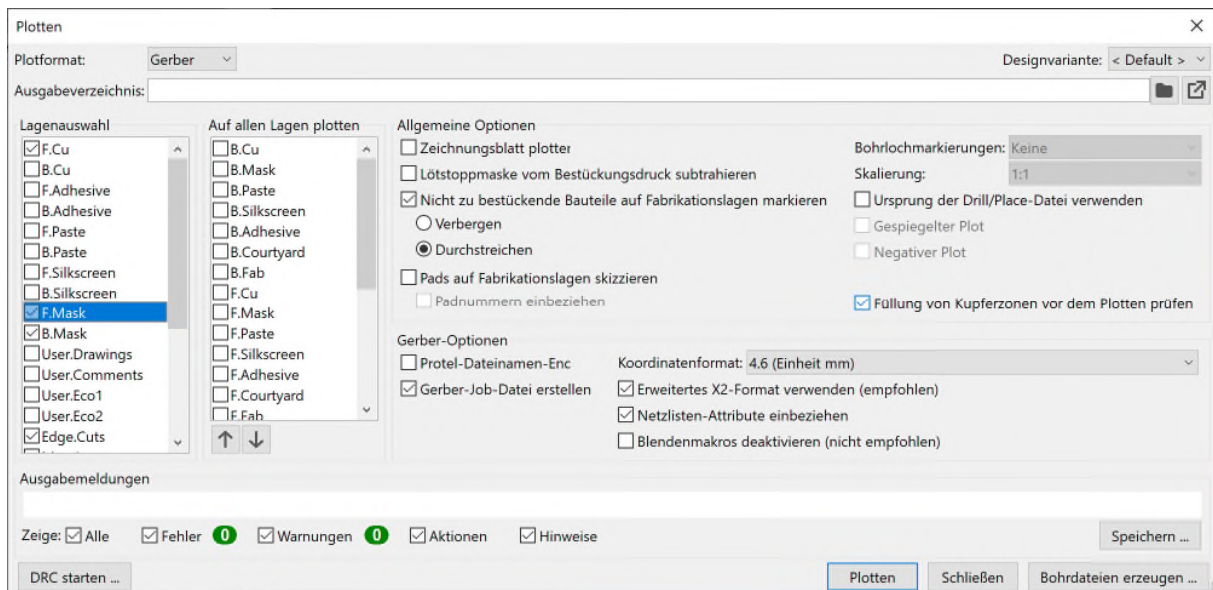
=> **Speichern**



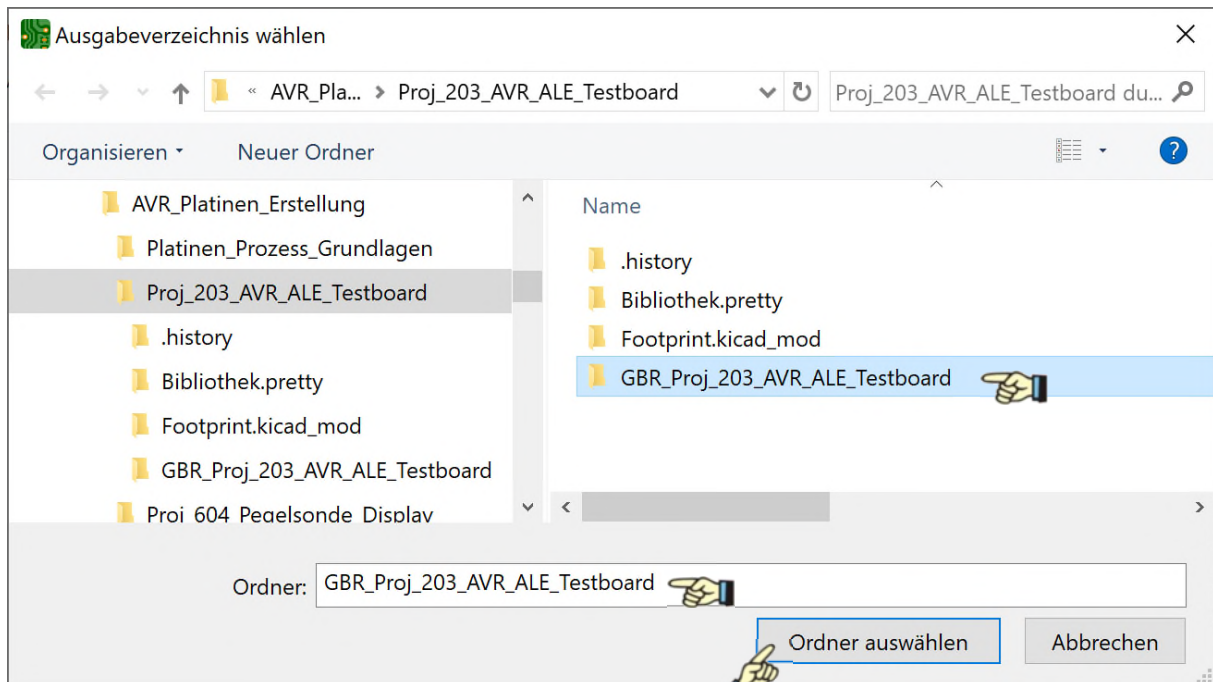
=> **Ja**

06.4: Layout für fertige Leiterplatte "plotten"

1. Im **Leiterplatteneditor** Menü **Plotten** aufrufen: **Datei => Plotten ...**
2. **Lagenauswahl** bestimmen auf: **F.Cu, F.Mask, B.Mask** und **Edge.Cuts**

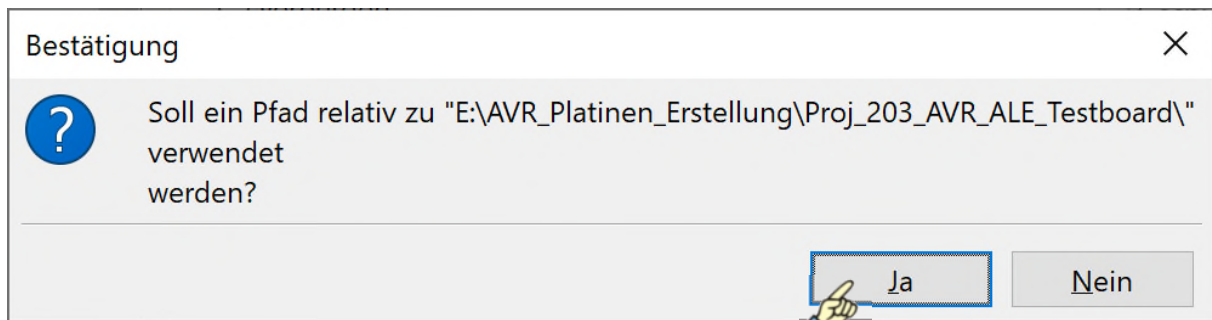


3. Um die Ausgabedateien gezielt in den GBR-Ordner abzulegen, muss das gewünschte Ausgabeverzeichnis **GBR_Proj_203_AVR_ALE_Testboard** mit dem rechten Ordner-Symbol  aufgerufen werden:

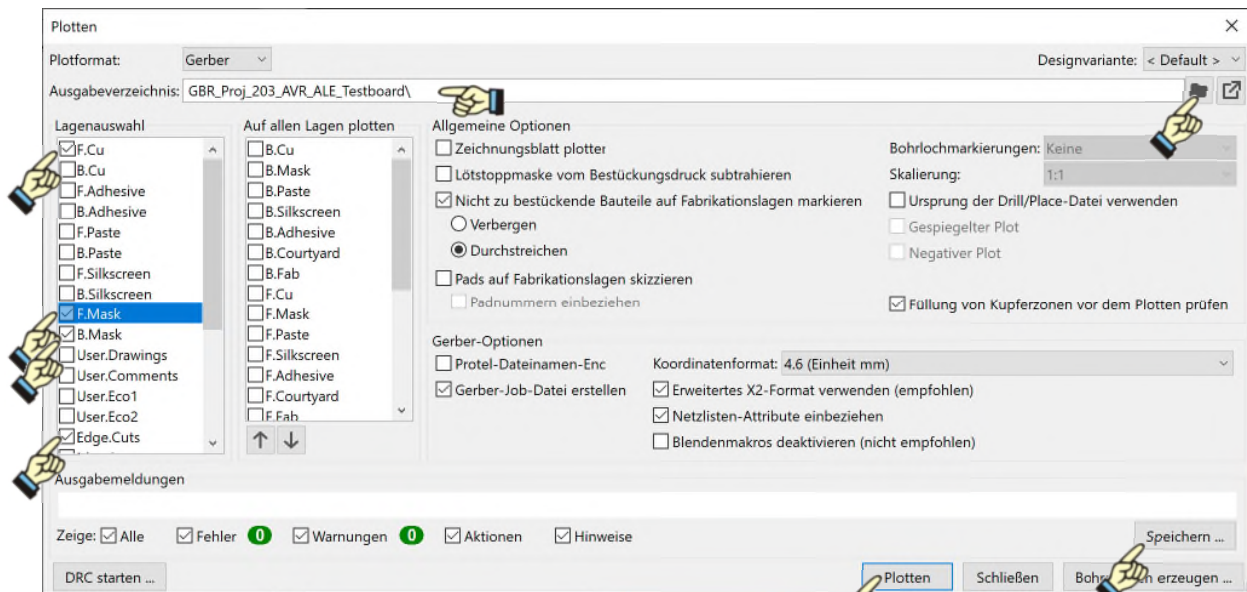


=> Ordner auswählen

Ggf. bedarf es noch folgender Bestätigung:



=> Ja

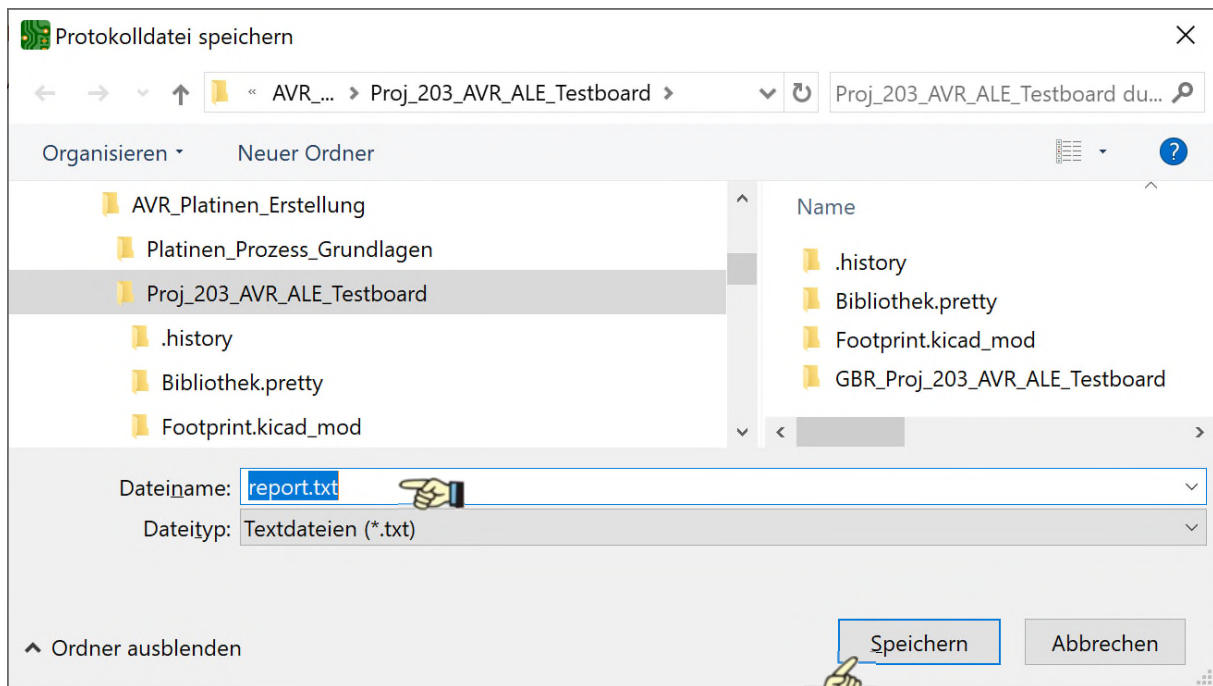


Zum Schluss dürfen auf der linken Seite nur noch folgende Häkchen gesetzt sein:

- F.Cu** Layer für die Kupferbahnen
- F.Mask** "gefüllte" Lötstopmmaske auf der Vorderseite
- B.Mask** "gefüllte " Lötstopmmaske für die Rückseite
- Edge.Cuts** Begrenzungsrahmen der Platine

3. Dann auf **Plotten** klicken.

4. Mit **Speichern ...** wird in einer Text-Datei **report.txt** ein Protokoll festgehalten:

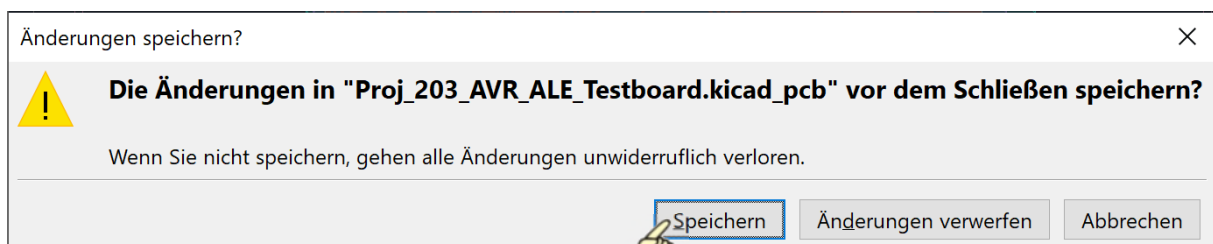


= > **Speichern**

Im Ausgabeordner **GBR_Proj_203_AVR_ALE_Testboard** liegen jetzt 5 neue GBR-Dateien:

Proj_203_AVR_ALE_Testboard-B_Mask.gbr	downloadbar
Proj_203_AVR_ALE_Testboard-Edge_Cuts.gbr	downloadbar
Proj_203_AVR_ALE_Testboard-F_Cu.gbr	downloadbar
Proj_203_AVR_ALE_Testboard-F_Mask.gbr	downloadbar
Proj_203_AVR_ALE_Testboard-job.gbrjob	downloadbar

5. Nun kann der **Leiterplatteneditor** mit **Datei => Schließen** geschlossen werden. Wenn die Änderungen noch nicht alle gesichert wurden, warnt **KiCad** mit dem wichtigen Hinweis:



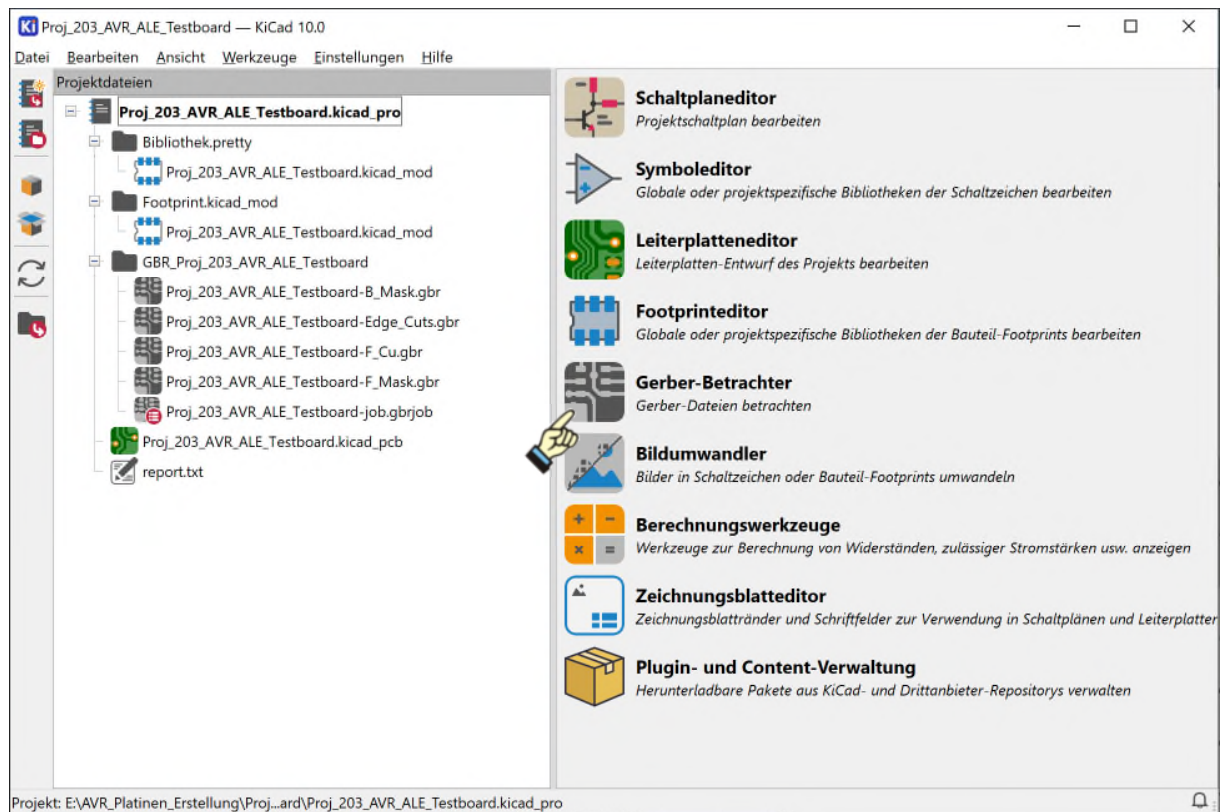
6. Mit **= > Speichern** wird der **Leiterplatteneditor** geschlossen und alle Änderungen des Projektes werden in folgenden Dateien gesichert:

Proj_203_AVR_ALE_Testboard.kicad_pcb
Proj_203_AVR_ALE_Testboard.kicad_prl
Proj_203_AVR_ALE_Testboard.kicad_pro

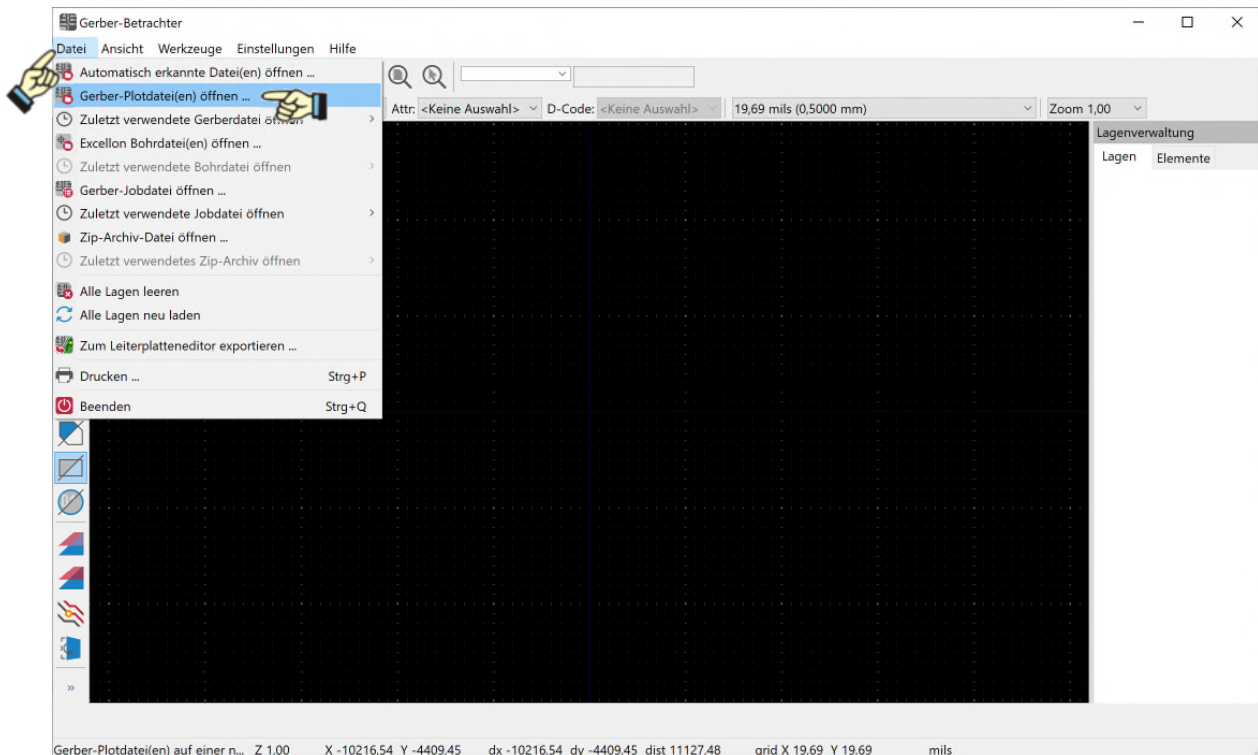
Andernfalls gehen alle Änderungen unwiderruflich verloren.

07: Gerber-Betrachter: Layout-Kontrolle

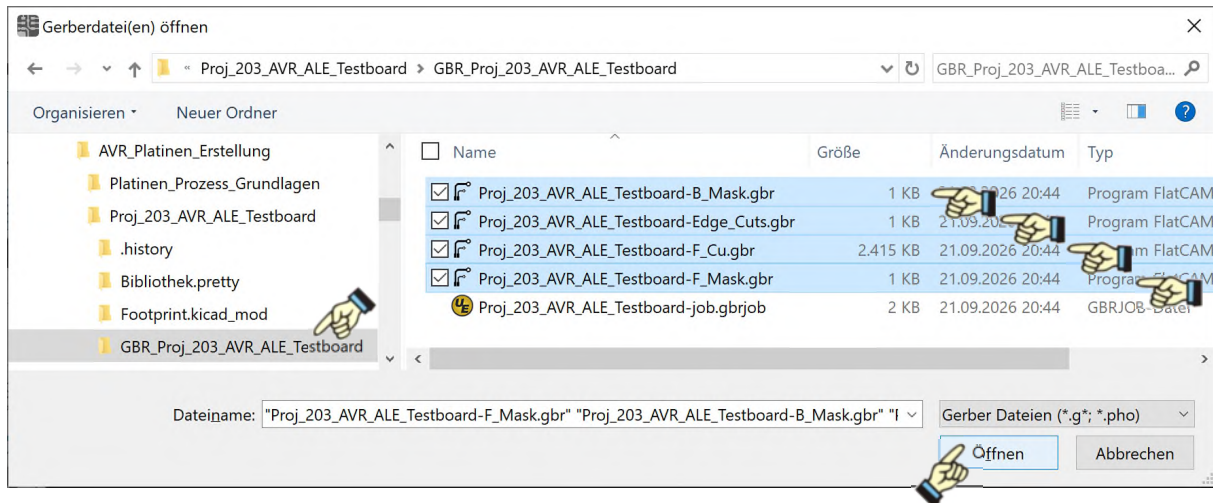
1. Der Projekt-Ordner **Proj_203_AVR_ALE_Testboard** hat nun folgende Struktur:



2. Zur Kontrolle wird der Gerber-Betrachter geöffnet:

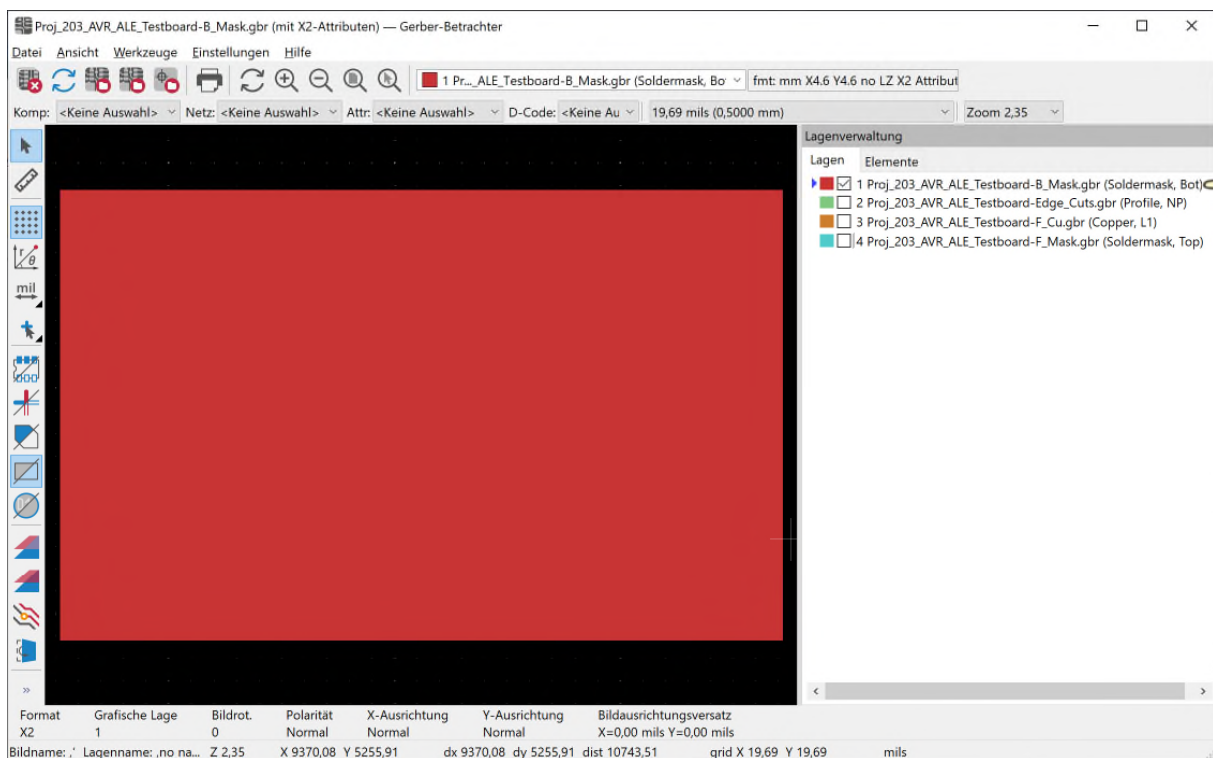


3. ... und mit **Datei => Gerber-Plotdatei(en) öffnen ...** werden die gerade erstellten Gerber-Dateien einzeln oder als Gruppe aufgerufen:

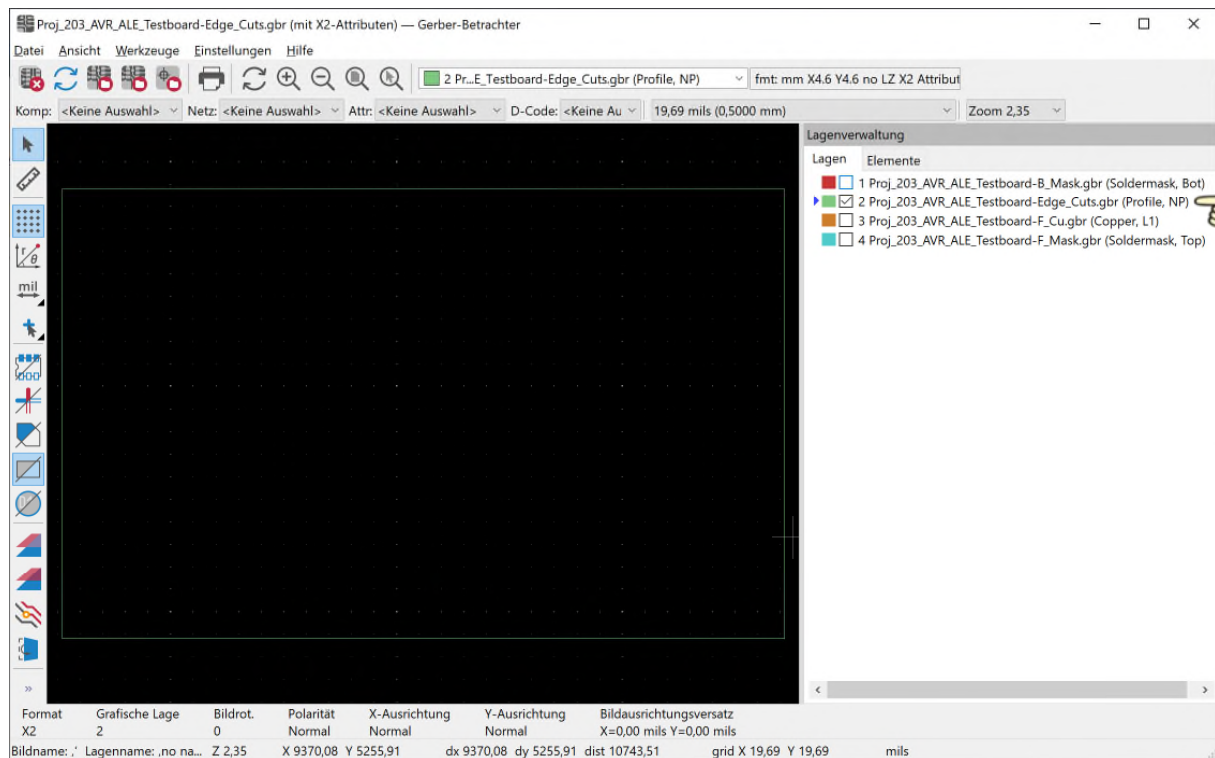


4. Gerber-Dateien der Reihe nach kontrollieren:

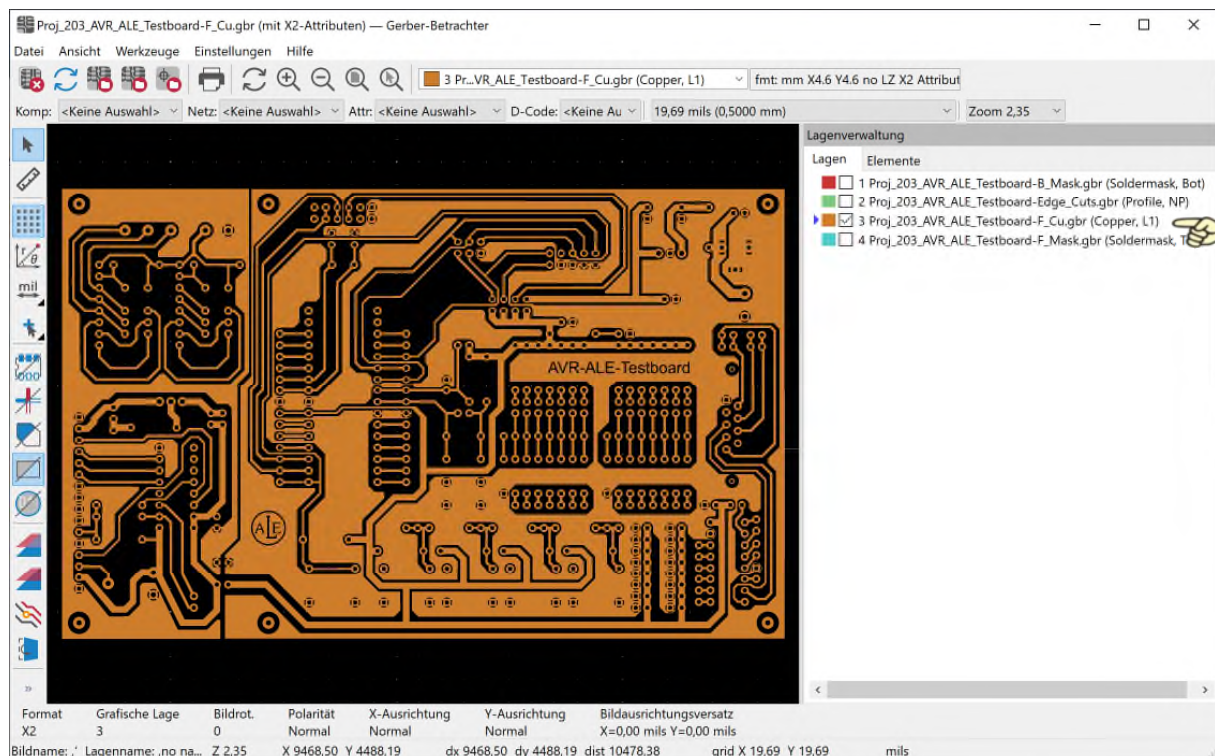
a. Lötstopppmaske für Rückseite (B.Mask)



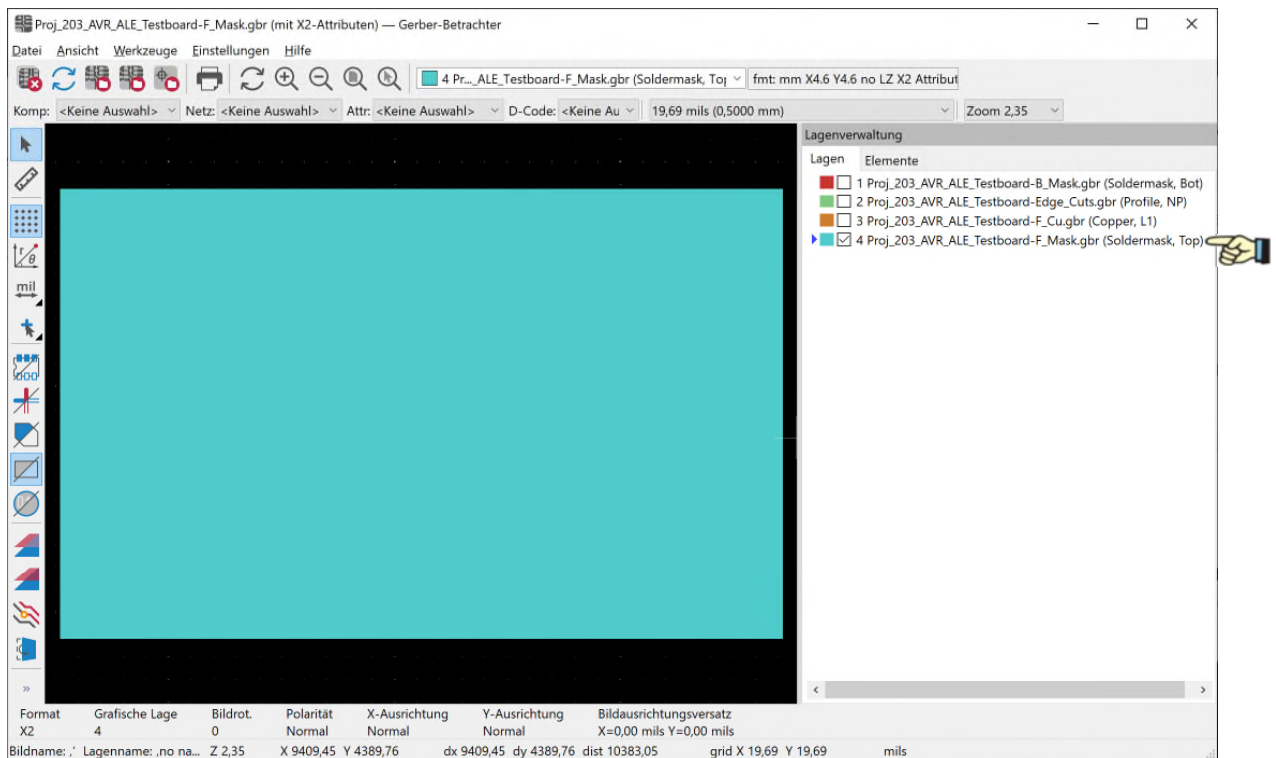
b. Layout-Begrenzungsrahmen (Edge.Cuts)



c. Cu-Layout (F.Cu)



d. Lötstopppmaske für Vorderseite (F.Mask)



08: Die ZIP-Datei für den Hersteller fertigen

Wenn das Layout im Gerber-Viewer korrekt aussieht, werden die GBR-Dateien für den Upload an den Hersteller in einer ZIP-Datei komprimiert:

1. Mit einem **Rechtsklick** auf den Ausgabeordner wird gewählt:

GBR_Proj_203_AVR_ALE_Testboard

Senden an => ZIP-komprimierter Ordner.

2. Es entsteht eine Datei namens **GBR_Proj_203_AVR_ALE_Testboard.zip**.
(hier downloadbar: [GBR_Proj_203_AVR_ALE_Testboard.zip](#))
3. Diese eine ZIP-Datei wird auf die Webseite des gewählten Platinenherstellers hochgeladen. Die dortigen Online-Vorschau-Tools zeigen dann meist sofort ein Bildergebnis der fertigen Platine an.

Das ZIP-Archiv ist damit perfekt bestückt.

Die Datei mit der Endung **.gbrjob** wird von **KiCad** automatisch erstellt. Sie enthält technische Metadaten über das Projekt (z.B. welche Layer verwendet werden). Die meisten modernen Platinenhersteller lesen diese Datei beim Hochladen automatisch mit aus, um die Einstellungen für Ihre Platine (wie z.B. die Anzahl der Lagen) direkt richtig zu konfigurieren.

Die ZIP-Datei kann man jetzt unverändert beim Hersteller hochladen. Da die Platine keine Lötstopppmaske (den typisch grünen Lack) erhält, wird die Platine in der Fabrik als sogenanntes **"Nacktes Kupfer" (Bare Copper)** oder ggf. mit HASL-Silber (chemische Versilberung) gefertigt – perfekt für SMD-Lötungen und das anschließende manuelle Bohren.