

Oskar Entwickler_innen-Dokumentation

Johannes Střelka-Petz ©©© This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. https://creativecommons.org/licenses/by-sa/4.0/

31. Dezember 2020

Inhaltsverzeichnis

1	Was ist Oskar Zither?1.1QuellenQuellen	2 2
2	Für wen ist Oskar Zither?2.1Benötigte Fähigkeiten für die Herstellung	2 2
3	Brailleschrift	2
4	mobile, USB-HID Braille-Tastatur	3
5	Firmware 5.1 Gehäuse	5 6
6	Bauanleitung 6.1 Vorbereitung	$egin{array}{c} 7 \ 7 \end{array}$



1 Was ist Oskar Zither?

Oskar (Open Source Key ARrangement) Zither (Abbildung 1) ist eine mobile Hardware-Tastatur, welche die Eingabe der Blindenschrift Braille ermöglicht. Die Tastatur wird auf der Rückseite eines Smartphones oder eigenständig bedient. Oskar kann wie ein Smartphone, ohne feste Unterlage, im Stehen und Gehen verwendet werden.

1.1 Quellen

https://oskar.ddns.mobi/wiki/Zither

- Oskar Zither https://gitlab.com/teamoskar/oskar_zither
 - Gehäuse https://gitlab.com/teamoskar/oskar_zither_case
 - Platine https://gitlab.com/teamoskar/oskar_zither_pcb
 - Firmware https://gitlab.com/teamoskar/oskar_firmware_arduino

2 Für wen ist Oskar Zither?

Die mobile Braille-Tastatur Oskar Zither ist ein Hilfsmittel für sehbehinderte und blinde Smartphone-Nutzer_innen welche Braille beherrschen oder lernen wollen. Mit Oskar Zither können Blinde und Sehbehinderte Menschen 4 mal schneller und 2 mal genauer tippen als am Smartphonetouchscreen.¹

2.1 Benötigte Fähigkeiten für die Herstellung

Bei der Herstellung des Gehäuses wird ein 3D Drucker bedient. Falls die Platine nicht vorgefertigt bezogen wird, ist eine Platine zu ätzen. Bei der Bestückung der Platine wird gelötet. Das Gehäuse wird mit Schrauben befestigt.

3 Brailleschrift

1825 wurde die Brailleschrift von Louis Braille veröffentlicht. Braille ist die heute am weitesten verbreitete Blindenschrift. Das Braillezeichen (Braillezelle) der Blindenschrift Braille besteht aus sechs taktilen Punktpositionen. Die Positionen sind in einem aufrecht stehenden Rechteck aus drei Zeilen

¹https://oskar.ddns.mobi/mediawiki/images/a/af/Bac-oskar.pdf



und zwei Spalten angeordnet. Die Durchnummerierung der ersten 6 Punkte einer Braillezelle erfolgt von der linken Spalte oben nach unten und von der rechten Spalte von oben nach unten (Abbildung 2). Das Basis-System der deutschen Blindenschrift ermöglicht die Wiedergabe eines Textes in 6-Punkte-Blindenschrift (Abbildung 3).

Computer-Braille erweitert die 6-Punkte-Blindenschrift um zwei weitere Punkte unterhalb des 6-Punkte-Blindenschrift-Zeichens auf die 8-Punkte-Brailleschrift. Der siebte Punkt liegt in der linken und der achte Punkt in der rechten Spalte (Abbildung 2). Computer-Braille (auch als Euro-Braille bezeichnet) wird definiert durch die Deutsche Norm DIN 32 982 "8-Punkt-Brailleschrift für die Informationsverarbeitung".

4 mobile, USB-HID Braille-Tastatur

Oskar Zither ist ein Tastatur der USB HID Klasse (USB HID class keyboard). Die USB HID Klasse² (Human Interface Device) definiert eine Klasse von Geräten des USB-Standards welche direkt vom Menschen bedient werden. USB³ (Universal Serial Bus) ist ein Standard für Kabel, Stecker und Kommunikationsprotokolle für die Verbindung, Kommunikation und Stromversorgung von Endgerät und externen Geräten. Das USB HID Protokoll der USB HID Klasse kann über USB-Kabel und USB On-The-Go (USB-OTG) übertragen werden. Der USB-OTG-Standard, eine Erweiterung von USB, erlaubt die direkte Kommunikation zwischen externen Geräten (Smartphone) ohne Host (Computer).

Abbildung 4 zeigt den schematischen Aufbau von Oskar. Der Aufbau besteht aus den Tasten, welche mit dem Mikrocontroller eines Arduino Micro⁴ verbunden sind und einem Smartphone, zu dem die Texteingaben mit einem USB-Kabel übertragen werden. Arduino⁵ ist eine open-source (quelloffene) Mikrocontroller Entwickler Plattform aus Hard- und Software.

Der Mikrocontroller eines Arduino Micro Entwicklerboards übersetzt die Akkorde in einen Datenstrom, der von den Endgeräten als Tastatureingaben interpretiert wird. Im Arduino Micro ist der Mikrocontroller ATmega32U4 verbaut. Mit dem Mikrocontroller ATmega32U4 lässt sich das USB HID Protokoll über USB und USB-OTG übertragen.

Den Akkorden (Tastenkombinationen) sind USB HID class keyboard Zeichencodes zugeordnet. Bei Oskar Zither entspricht die Zuordnung der Zei-

²https://en.wikipedia.org/wiki/USB_human_interface_device_class

³http://usb.org

⁴https://store.arduino.cc/arduino-micro

⁵https://www.arduino.cc/





Abbildung 1: Hände an der Oskar Zither



Abbildung 2: 6-Punkte-Blindenschrift und die Erweiterung durch die Punkte 7 und 8 auf die 8-Punkt-Brailleschrift

● 0 0 0 0 0	● 0 ● 0 0 0								$\begin{array}{c} \circ \bullet \\ \bullet \bullet \\ \circ \circ \end{array}$
A	В	С	D	E	F	G	Η	Ι	J
	● 0 ● 0 ● 0				• • • 0 • 0	• • • • • 0			$\begin{array}{c} \circ \bullet \\ \bullet \bullet \\ \bullet \circ \end{array}$
	_			~	-	-	_	~	
K	L	Μ	Ν	0	Р	Q	R	S	Τ
K •••		M ∷	N °°°	O ::	P ∷		R 	S •••	T •••

Abbildung 3: Das Alphabet des Basis-Systems der deutschen Blindenschrift

Seite 4 von 15



chencodes zu den Akkorden der Deutsche Norm DIN 32 982 "8-Punkt-Brailleschrift für die Informationsverarbeitung" angelehnten Erich Schmids 10 Keys Braille. (Siehe https://gitlab.com/teamoskar/oskar_firmware_arduino/-/blob/ master/brailletable.md#braille) Damit Gelegenheit bleibt, die Akkorde vorzubereiten, werden erst beim Lösen der Tasten die Zeichencodes des USB HID class keyboards gesendet.

Verwendet wird Oskar mit den Tasten vom Körper abgewandt. 8 Tasten auf der Rückseite von Oskar Zither entsprechen jeweils einem Braillepunkt eines Braille-Zeichens. Die Finger ruhen auf den Tasten. Die Handfläche hälten das Gerät. Zwei zusätzliche Tasten für die Daumen ermöglichen die Eingabe des Leerzeichens und weiterer Tastenkombinationen. So kann mit jeweils vier Fingern von jeder Hand getippt werden, ohne eine Unterlage zu verwenden (Abbildung 1)

5 Firmware

Die Zuordnung der Zeichencodes zu den Akkorden erfolgt in der Bibliothek braille_german⁶ in den Zuordnungs-Tabellen keymap_space1, keymap_space2 und keymap_function. Die Zuordnungs-Tabellen bestehen aus Einträgen mit den Zeichencode Feldern Taste, Modifikator, Tottaste und Tottaste Modifikator die nach dem Byte-Wert des Akkord gereiht sind. Jede einzelen Taste wird einem Bit in einem Byte (8 Bit in einem Byte), entsprechend der Durchnummerierung der Punkte in der Braillezelle (Abbildung 2), zugeordnet. Durch das Drücken der Tasten im Akkord werden die zugeordneten Bits im Byte von 0 auf 1 gesetzt und ergeben den Byte-Wert (0-255). Jedes Zeichencode Feld eines Eintrags entspricht einem Tastendruck einer Tastatur.

- Das Feld Taste entspricht einem einfachen Tastendruck, z.B (i).
- Die Bedeutung eines Tastendrucks kann durch die Kombination mit einen Modifikator verändert werden, z.B. Umschalttaste und (i) ergibt (I).
- Die Bedeutung eines Tastendrucks kann weiters durch den vorrausgehenden Tastendruck einer Tottaste verändert werden, z.B. Akut (') und anschließend (i) ergibt (í).

⁶https://gitlab.com/teamoskar/oskar_firmware_arduino/-/blob/master/ libraries/braille_german/braille_german.cpp



• Auch die Bedeutung des vorrausgehenden Tastendrucks kann durch die Kombination mit einerm Modifikator verändert werden, z.B. Umschalttaste und Akut (') ergibt Gravis (') und anschließend (i) ergibt (ì).

Weiters kann zwischen den Zuordnungs-Tabellen gewechselt werden um mehrere Ebenen der Zuordnung zu erhalten. In der Funktion modes _n_write in oskar_firmware_arduino.ino 7 wird geprüft ob eine Tastenkombination gedrückt wurde welche die Zuordnungs-Tabellen wechselt. Der Wechsel der Zuordnungs-Tabelle erfolgt durch

- Tastenkombination mit einer Leertastem
- einmalig für den nächsten Tastendruck wie bei den Tottasten durch eine Umschalt-Tastenkombination (ßticky keysklebende Tasten").
- oder dauerhaft durch doppelte eingabe der Umschalt-Tastenkombination bis zur erneuten doppelten eingabe der Umschalt-Tastenkombination. (Modi)

Die Bedeutung der Zeichencodes in den Zuordnungs-Tabellen ändert sich mit den Länder spezfischen Tastaturlayouts. Wird zum Beispiel zwischen englischem (QWERTY) und deutschem (QWERTZ) Tasaturlayout gewechselt, werden die Tasten (y) und (z) vertauscht. Für jedes Tastaturlayout muss die Zuordnungstabelle angepasst sein. Derzeit wird mit braille_german das deutsche Tastaturlayout unterstützt. Für die Unterstützung weiterer Sprachen und länderspezifischen Tastaturlayouts sind Bibliotheken wie braille_german anzulegen und diese wie in oskar_firmware_arduino.ino⁷ mit #include "braille_german.h" einzubinden.

5.1 Gehäuse

In der OpenSCAD Datei zither.scad ⁸ befindet sich neben dem Gehäuse (Modul case) auch Modelle der Komponenten im Gehäuse (Modul keypad, Modul arduino, Module pdb, Modul thumbs2). Die Modelle der Komponenten helfen die Mindestabmessungen einzuhalten und unterstützen Toleranzangaben. Am Anfang der Datei werden grundlegende Parameter und Komponentenabmessungen welche in mehreren Modulen benötigt werden. Nach den Parametern folgen Module welche zum teil in der angeführten Reihenfolge aufeinander aufbauen.

⁷https://gitlab.com/teamoskar/oskar_firmware_arduino/-/blob/master/ oskar_firmware_arduino.ino

⁸https://gitlab.com/teamoskar/oskar_zither_case/-/blob/master/zither. scad



Das Modul keypad eine Anordnung von kailh choc PG1350 Modulen. Das Modul thumbs2 ist eine Anordung von Modul thumb unter zuhilfenahme der Hilfsfunktion Modul casecenter mirror. Modul casecenter mirror spiegelt Objekte um die Längsachse des Keypad bzw. Gehäuse. Modul bttom und support plane erschaffen eine gerade Grundfläche für den 3D-Druck, die sich an der Position der Daumentasten ausrichtet. Modul case inner hüllt die inneren Komponenten ein und schafft so die Gehäuseform die von Modul case outer von einer Gehäusewand umgeben wird. Modul case erweitert das Gehäuse um Schraubenführungen und entfernt case inner damit Platz für die Komponenten im inneren ist und die Ausnehmungen für Keypad, Daumentasten und Arduino USB-Anschluss entstehen. (Abbildung 5) Modul case n thumbplateholder fügt Führungen für Daumentastenrahmen hinzu. Die Daumentastenrahmen Modul thumbplate werden einzeln gedruckt da diese eine höhere Toleranz als das Gehäuse verlangen. Die höhere Toleranz der Daumentastenrahmen wir dadurch erreicht, dass die Rahmen direkt auf der Bodenplatte gedruckt werden und nicht überhängend im Gehäuse. Modul case front cut schneidet die Rückseite vom Gehäuse damit zwei 3Ddruckbare Hälften entstehen. Modul case back cut entspricht dem Gegenstück von case front cut. In Modul print front, print back, print tplate wird für eine ebene Auflage auf der Bodenplatte beim Druck gesorgt.

6 Bauanleitung

6.1 Vorbereitung

- PCB Fertigung ⁹
 - -Zum Beispiel die Platine bei Aisler.
net bestellen 10
- 3D Druck ¹¹
 - resolution 0.2 mm
 - infill density 30%
 - filament PLA
- $\bullet\,$ Firmware 12

⁹https://gitlab.com/teamoskar/oskar_zither_pcb

¹⁰https://aisler.net/p/OPSGPOBN

¹¹https://gitlab.com/teamoskar/oskar_zither_case

¹²https://gitlab.com/teamoskar/oskar_firmware_arduino





Abbildung 4: Schematischer Aufbau von Oskar Zither



Abbildung 5: Oskar Zither in OpenSCAD

Seite 8 von 15



6.2 Werkzeug

Abbildung 6 Werkzeug

- 3D Drucker
- Lötkolben
- Seitenschneider
- PZ1 Schraubendreher
- Computer

https://gitlab.com/teamoskar/oskar_zither/-/blob/master/tools.txt

6.3 Material

Abbildung 7 Material

material	count
Kailh Choc low profile Switch tactile, brown CPG135001D02	10
Kailh Choc low profile Kecaps	10
Arduino Micro	1
Axial Widerstand 1 kOhm, $1/4$ Watt	1
Spanplatten Schrauben 3 x 16 mm	4
Platin	1
Gehäuse 3D-Druck PLA Filament	1
- Vorderseite	
- Rückseite	
- Daumentastenrahmen	
Lötzinn, S n 99.3%, Cu 0.7% mit Flussmittel 2.2%	1
optional Lötfett	
Lötlitze 22AWG	1
USB-OTG Kabel USB-Micro-B-Stecker	1

https://gitlab.com/teamoskar/oskar_zither/-/blob/master/BOM.txt

6.4 Durchführung

• Platine Löten Um den Arduino Micro auf Vorderseite der Platine zu löten sind die Pins 2 bis 9, MOSI, SCK und Ground anzulöten. Für die mechanisch Stabilität empfiehlt es sich noch die 3.3V oder benachbarte Pins anzulöten. Ebenfalls werden die 8 Tasten und der 1k Ohm





Abbildung 6: Werkzeug



Abbildung 7: Material

Seite 10 von 15



Widerstand auf die Vorderseite der Platine gelötet. Die zwei Tasten für die Daumen werden in die zwei Daumentastenrahmen gesteckt und mit Schaltlitze an die mit R1kGND"beschrifteten Pins verbunden. Außerdem werden die zwei Daumentasten noch mit den Pins MISO und SCK verbunden. Wobei die näherliegenden Pins verwendet werden, linke Taste zum linkeren Arduino Pin und rechte Taste zum rechteren Pin. Linke Daumentaste mit SCK verbinden (kürzere Verbindungsstrecke) und rechte Daumentaste mit MOSI verbinden (längere Verbindungsstrecke) (Abbildung 8).

- Firmware flashen
 - $-\,$ Arduinio IDE 13 installieren
 - * Oskar Zither Firmware¹² installieren.git clone https://gitlab. com/teamoskar/oskar-firmware-arduino.git
 - Verwendung der Arduino IDE Das Projektverzeichnis oskar_firmware_arduino in das Arduino Scatchbook Verzeichnis (/home/user/Arduino") kopieren. Die Bibliothek aus dem Projektverzeichnis/libraries in die Arduino Scatchbook Bibliothek ("home/user/Arduino/libraries") kopieren. Das Scatchbook oskar_firmware_arduino in Arduino öffnen. Board Ärduino Microäuwählen dann "Verify/Compile". Arduino Micro mit USB-Kabel mit dem Rechner verbinden und mit Üpload"hoch laden.
 - Alternativ zur Arduino IDE Benutzeroberfläche kann Arduino-Makefile ¹⁴ verwendet werden. Zusätzlich zur Arduino IDE noch Arduino-Makefile installieren. Die Pfade ARDUINO_DIR, ARDMK_DIR und AVR_TOOLS_DIR im Makefile an die tatsächlichen Pfade anpassen. Nach der Anpassung des Makefiles mit makecompilieren und mit make upload"das Programm auf den Arduino Micro hoch laden.
- Gehäuse
 - Daumentasten befestigen Die Tasten rasten in die Daumentastenrahmen (siehe Baumaterial, zwei Daumentastenrahmen in Orange, links neben den Tasten) genau ein. Da die Tasten eine quadratische Grundfläche haben ist die Orientierung der Tasten in

¹³https://www.arduino.cc/en/Main/Software

¹⁴https://github.com/mjoldfield/Arduino-Makefile



den Daumentastenrahemn für die mechanische Stabilität unbedeutend. Daumentastenrahmen mit den enthaltenen Tasten werden mit der Seite des einzelnen breiten Rahmens voran in die Führungen im Gehäuseboden eingeschoben. Der schmale Teil des Daumentastenrahmen sieht damit zur Oberseite bzw. zum Gehäusedeckel mit der Tastenfeldausnehmung.

- Zusammenbau Vor dem Zusammenbau die Tastenkappen auf die Tasten stecken und die ICSP Stifte (2x3 Stifte neben Reset Knopf am Arduino Micro) mit dem Seitenschneider möglichst knapp am Arduino Micro abzwicken. (Abbildung 9) Schraubenlöcher der Platine über den Schraubenlöchern des Gehäuseboden positionieren. Gehäusedeckel aufsetzen und zusammen schrauben. (Abbildung 10 11)
- Anschluss Smartphone mit USB-Micro-B-Stecker USB-OTG Cable und passendem Stecker (USB-Micro-B, USB-Typ-C) verbinden. iPhones benötigen Apple-Lightning-Stecker zu USB-Typ-A-Buchse und Apple-Lightning-Buchse "Apple Camera Adapter Lightning to USB 3" mit Energieversorgung über Apple-Lightning-Buchse. Rechner mit USB-Micro-B-Stecker zu USB-Typ-A-Stecker verbinden.

Kabel	Erfolgreich
OTG USB-Micro-B-Stecker zu USB-Typ-A-Stecker	PC Linux, Mac iOS
OTG USB-Micro-B-Stekcer zu USB-Micro-B-Stecker	Android Smartphone
RC-Cable USB-Micro-B-Stecker zu USB-Micro-B-Stecker	Android Smartphone
OTG USB-Micro-B-Stecker zu USB-Typ-A-Buchse	
mit USB-Micro-B-Buchse zu USB-Typ-C-Stecker Adapter	Android Smartphone
Apple-Lightning-Stecker zu USB-Typ-A-Buchse	
und Apple-Lightning-Buchse mit Netzteil	
über Apple-Lightning-Buchse	iPhone
Apple-Lightning-Stecker zu USB-Typ-A-Buchse	
und Apple-Lightning-Buchse mit Powerbank	
über Apple-Lightning-Buchse	iPhone





Abbildung 8: Lötstellen



Abbildung 9: Frontansicht ohne Abdeckung

Seite 13 von 15





Abbildung 10: Schrauben im Gehäuse



Abbildung 11: Oskar Zither Vorderansicht

Seite 14 von 15



Kabel	Erfolglos
Apple-Lightning-Stecker zu USB-Typ-A-Buchse	
und Apple-Lightning-Buchse mit Oskar Zither	
über USB-Typ-A-Buchse	iPhone
	(Cannot Use Accessory,
	USB IO Board:
	This accessory
	requires too much power.)
Apple-Lightning-Stecker zu USB-Typ-A-Buchse	
und Apple-Lightning-Buchse mit Oskar Zither	
über Apple-Lightning-Buchse	iPhone
Apple-Lightning-Stecker zu USB-Typ-A-Stecker	iPhone
RC-Cable USB Micro to Lightning	iPhone
RC-Cable USB-Micro-B-Stecker zu USB-Typ-C-Stecker	Android Smartphone