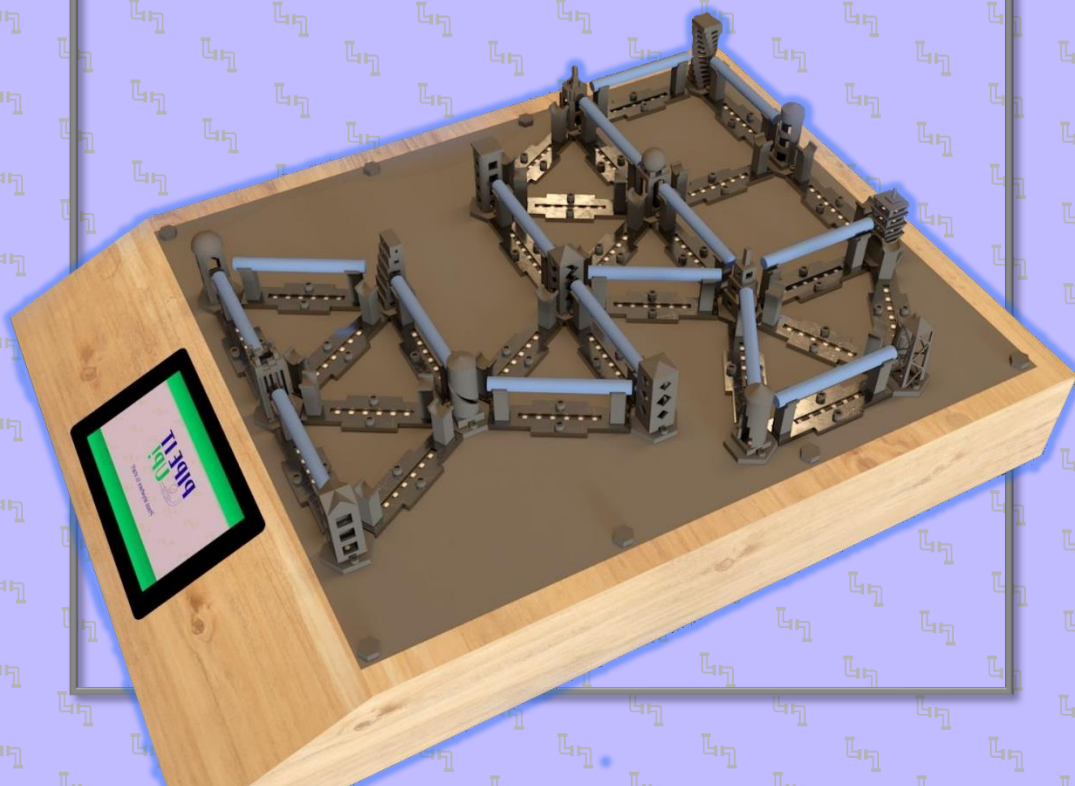


Bedienungs- und Wartungsanleitung

PIPE IT UP!



Inhalt

1	Was ist Pipe It Up?	4
1.1	Entstehungsgrund.....	4
1.2	Nutzen	4
1.3	Prim-Algorithmus.....	5
1.4	Kruskal-Algorithmus.....	5
2	Spielmechanik.....	6
2.1	Bedienung.....	6
2.2	Prim-Algorithmus.....	7
2.3	Kruskal-Algorithmus.....	7
3	Physische Komponenten	8
3.1	3D-Druck-Komponenten	8
3.2	3D-Druck-Dateien.....	9
3.3	Anpassungen am 3D-Modell.....	9
3.4	Druck einer Komponente	10
3.5	Sonderkomponente «Knoten».....	11
3.6	Knoten-Nummerierungen:.....	12
3.7	Befestigung der 3D-Komponenten	13
3.8	Verkabelung der Topplatte (LEDs)	14
3.9	Verkabelung der Level Shifter für LEDs.....	16
3.10	Verkabelung der Buttons (Kanten).....	17
3.11	I/O Expander & Button Verbindung.....	19

3.12	Verkabelung & ID der I/O Expander	21
3.13	Anschlüsse des Raspberry Pi	23
3.14	Lüfter	24
3.15	Raspberry Pi 7-Inch Touch Screen Display	25
4	Software-Komponenten	26
4.1	Installation	26
4.1.1	Image auf SD-Karte laden	26

1 Was ist Pipe It Up?

Das Brettspiel «Pipe It Up!» ist ein didaktisches Spiel, konzipiert für Informatik-Studierende im ersten Semester an der FHNW. Es thematisiert die elementaren Prim- und Kruskal-Algorithmen der Graphentheorie, mit welchen auf verschiedene Weisen ein minimaler Spannbaum berechnet werden kann.

Diese beiden Algorithmen sind ein fester Bestandteil des Mathematikstoffs jedes 1.Semester Informatik-Studierenden. Sie bilden unter anderem die Grundlage für die weiterführende informatische Graphentheorie.

1.1 Entstehungsgrund

Der ursprüngliche Beweggrund für die Entwicklung von «Pipe It Up!» war die Absolvierung der ersten Informatik-Projektschiene. Dabei entwickelt eine zufällig generierte Gruppe aus Informatik-Basisjahr- und iCompetence-Studierenden ein Informatik-Produkt zu einer von sieben unterschiedlichen Aufgabenstellungen. In unserem Fall wurde es vorgegeben die beiden Algorithmen anhand von Pipelines zu erklären, um so einen Bezug zum echten Leben zu schaffen und die Thematik für die Studierenden greifbarer zu machen.

1.2 Nutzen

Die grundlegende Funktion und Sinn hinter dem Brettspiel «Pipe It Up!» ist das spielerische Vermitteln der zwei Algorithmen von Prim und Kruskal an Informatik-Studierende, die das «mgli»- Modul

besuchen. Dieser Lehreffekt wird durch den Bezug zum echten Leben erzeugt, der das Spiel bietet.

Anstelle von Fachbegriffen aus der Graphentheorie, werden die Studenten*innen mit ihnen bereits vertrauten Begriffen wie Pipelines und Städten konfrontiert. Dadurch erhoffen wir uns, dass jeder Nutzer*in unseres Spiels nach der Verwendung die Funktionsweise des Prim- und Kruskal-Algorithmus versteht, sowohl wie die Algorithmen anschliessend in einem konventionellen Kontext anwenden kann.

1.3 Prim-Algorithmus

Die Anwendung des Algorithmus von Prim ermöglicht die Berechnung eines minimalen Spannbaumes in einem zusammenhängenden, ungerichteten, kantengewichteten Graphen. Der tschechische Mathematiker Vojtěch Jarník entwickelte den Algorithmus im Jahr 1930.

1.4 Kruskal-Algorithmus

Der Algorithmus von Kruskal berechnet einen minimalen Spannbaum in einem ungerichteten Graphen. Voraussetzungen für den Graphen sind, dass er zusammenhängt, gewichtete Kanten hat und endlich ist. Im Jahr 1956 veröffentlichte Joseph Kruskal erstmals seinen Algorithmus in der Zeitschrift «Proceedings of the American Mathematical Society».

2 Spielmechanik

Die treibende Kraft hinter unserem Brettspiel ist der Raspberry Pi, der mithilfe unserer selbstentwickelten Software die gelegten Kanten erkennen und validieren kann. Unter der Topplattform mit den Pipelines ist eine weitere Plattform vorhanden, auf welcher für jede Kante je zwei Buttons verlötet wurden. Wenn nun eine Pipeline gelegt und hinuntergedrückt wird, registriert unsere Software die ausgewählte Kante anhand der untenliegenden betätigten Buttons und überprüft deren Richtigkeit.

2.1 Bedienung

1. «Pipe It Up!» an den Strom anschliessen und anschalten.
2. Warten, bis das Spiel gestartet hat.
3. Mithilfe des Touchscreens durch das Menu navigieren und den gewünschten Modus auswählen und starten.
4. Wichtig: Vor dem Stecker ziehen, das Gerät über die Einstellungen im Hauptmenü zuerst abschalten.

2.2 Prim-Algorithmus

Um den Spielmodus für den Prim-Algorithmus zu starten, selektieren Sie zuerst «Spielmodus» und dann «Prim». Geben Sie Ihren Spielernamen ein, anschliessend startet das Spiel.

2.3 Kruskal-Algorithmus

Um den Spielmodus für den Kruskal-Algorithmus zu starten, selektieren Sie zuerst «Spielmodus» und dann «Kurskal». Geben Sie Ihren Spielernamen ein, anschliessend startet das Spiel.

3 Physische Komponenten

Das Brettspiel «Pipe It Up!» besteht aus einer Vielzahl an physischen Komponenten, die durch eine Wechselbeziehung das Spielen ermöglichen. Im Folgenden erhalten Sie einen Überblick über die einzelnen Bausteine und deren Eigenschaften.

3.1 3D-Druck-Komponenten

Ein grosser Teil des physischen Produkts besteht aus Komponenten, welche mittels 3D-Druck hergestellt wurden.

Deshalb ist es wichtig, falls die 3D-gedruckten Komponenten beschädigt sind und ersetzt werden müssen, zu wissen, wo die STL-Dateien (Datei-Format für den 3D-Drucker) zu finden sind und wie das Vorgehen zum 3D-Druck aussieht.

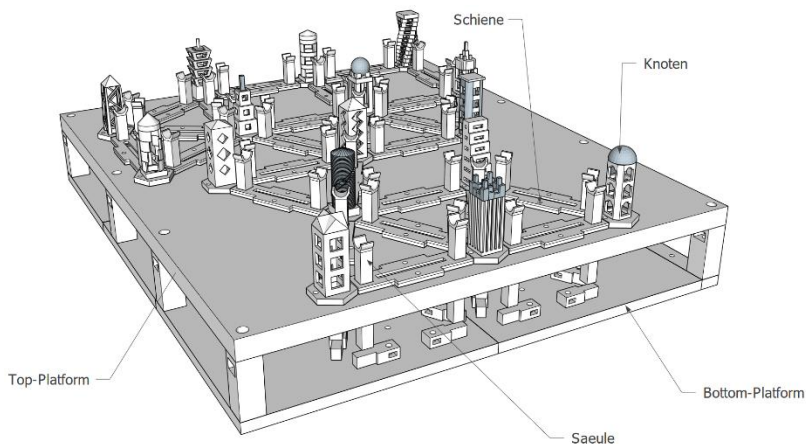


Abbildung 1: Übersicht über alle 3D-gedruckten Komponenten

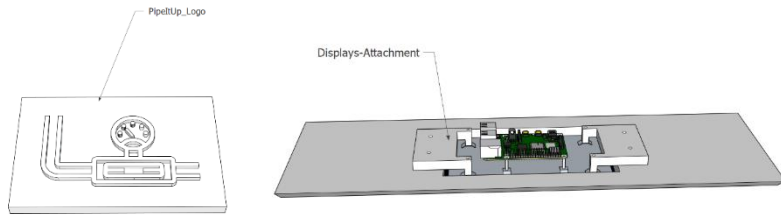


Abbildung 2 Pipe It Up! Logo & Display-Attachment

3.2 3D-Druck-Dateien

Alle Dateien für den 3D-Druck sind auf dem GitHub unter dem folgenden Link:

https://github.com/RononDex/PipeltUp/tree/master/3D_Print zu

finden. Jede Komponente hat seinen eigenen Ordner, welche den Namen der Komponente trägt. Die Namen der Komponenten sind in der Abbildung 1 ersichtlich. In jedem Ordner sind zwei Unterordner mit den Namen «CAD» und «STL» zu finden.

CAD steht für Computer-Aided Design, eine Software, welche für das Konstruieren/Designen von 3D-Modellen benötigt wird.

STL steht für Standard Triangle Language und ist die Datei-Format, welche nach der Fertigstellung des 3D-Modells von der CAD-Software exportiert wird. Die STL-Datei wird schlussendlich für den 3D-Druck benötigt.

3.3 Anpassungen am 3D-Modell

Für das Konstruieren von 3D-Modellen wird eine CAD-Software benötigt. Für das Projekt «Pipe It Up!» wurde die Software

«SketchUp» verwendet. Es gibt eine Gratis-Version, mit der man die nötigsten Werkzeuge für das Modellieren besitzt. Ein Tutorial ist auf der Website von SketchUp zu finden unter dem folgenden Link:

<https://help.sketchup.com/de/sketchup/getting-started-self-paced-tutorials>.

Die CAD-Dateien sind jeweils im Unterordner «CAD» zu finden; sie tragen die Datei-Format «.skp». Falls eine andere CAD-Software verwendet wird, müssen die Dateien entsprechend in die gewünschte Datei-Format umgewandelt werden.

3.4 Druck einer Komponente

Für jede Komponente finden Sie im Unterordner «STL» die finale Datei (mit dem Format «.stl») für den 3D-Druck. Falls eine Komponente angepasst wurde, muss die Datei in entsprechenden CAD-Software unter Dateityp «STL» exportiert werden.

Für den finalen 3D-Druck wird eine zusätzliche Software benötigt, um die STL-Dateien zu slicen (Fachbegriff für die Anpassung der Einstellungen des 3D-Drucks für das entsprechende Modell). Empfohlene Slicer-Software sind folgende:

- PrusaSlicer
(https://www.prusa3d.com/de/page/prusaslicer_424/)
- Ultimaker Cura
(<https://ultimaker.com/de/software/ultimaker-cura>)

Falls Sie die Komponente im Maker Studio an der FHNW 3D-drucken möchten, müssen Sie vorhin ein Einführungskurs belegen:

<https://makerstudio.fhnw.ch/digitales-atelier/> .

3.5 Sonderkomponente «Knoten»

Im Gegensatz zu den anderen 3D-gedruckten Komponenten, können die Knoten (oder Türme) nicht beliebig an die «Top-Plattform» befestigt werden.

Aufgrund der verschiedenen Winkel der Schienen (oder Kanten) besitzen die Knoten unterschiedliche Sockel-Varianten.

Die untenstehenden Abbildungen gibt eine Übersicht über das Anlegen der Knoten an die richtigen Stellen auf die «Top-Plattform».

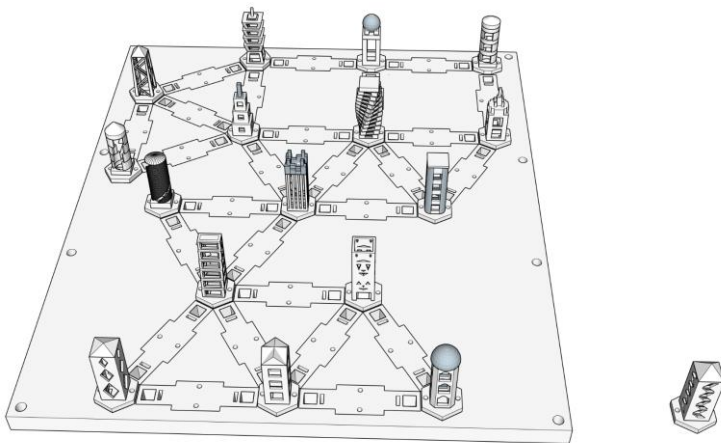










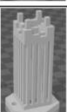








Abbildung 3: Alle Knoten/Türme

Der Knoten rechts von der «Top-Plattform» ist als Reserve gedacht und wurde im finalen Spiel nicht implementiert.

3.6 Knoten-Nummerierungen:

Knoten_NR	Bild
1	
2	
3	
4	
5	
6	

Knoten_NR	Bild
7	
8	
9	
10	
11	
12	

Knoten_NR	Bild
13	
14	
15	
16	
17	

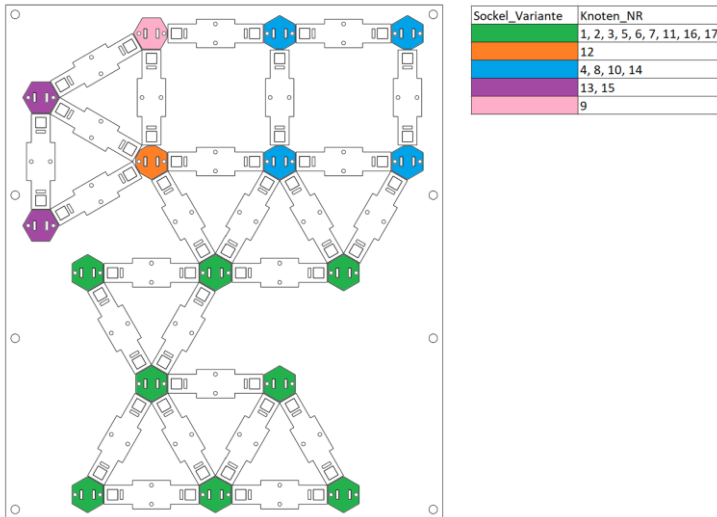


Abbildung 4: Platzierung der Knoten mit den unterschiedlichen Sockelvarianten

3.7 Befestigung der 3D-Komponenten

Auf dem GitHub finden Sie unter dem folgenden Link:

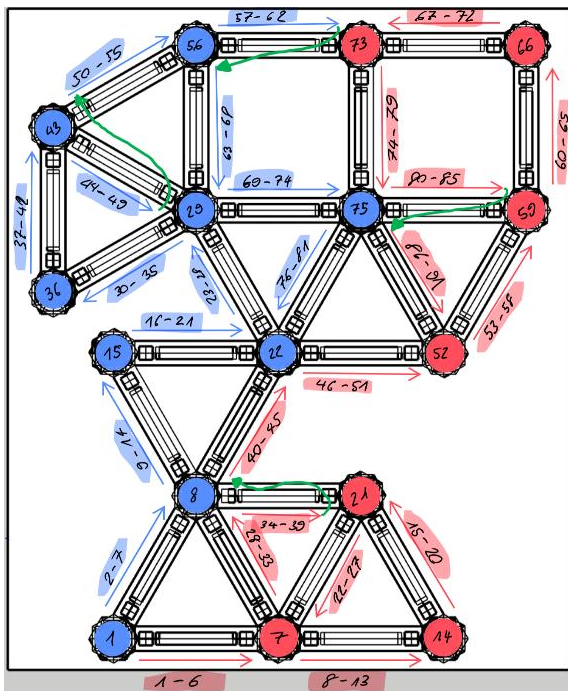
https://github.com/RononDex/PipeltUp/blob/master/3D_Print/Befestigung_3D%20Prints_Schraubenliste.xlsx die Details zur Befestigung der 3D-gedruckten Komponenten.

3.8 Verkabelung der Topplatte (LEDs)

Die LED-Verkabelung verläuft über zwei Linien. Auf der Abbildung 1 mit Blau und Rot eingezeichnet.

Diese Linien bestehen aus Knoten und Kanten und sind in einer spezifischen Reihenfolge angeordnet (Zahlenreihenfolge auf Abbildung 1).

Die Grünen Pfeile sollen Verbindungen, bei welchen kein Knoten dazwischen ist, einfacher veranschaulichen.



Die zwei Linien haben jeweils unterschiedliche Farben für ihre Verbindungskabel. Diese sind in der folgenden Tabelle zu entnehmen. Zudem ist die totale Anzahl von LEDs auch festgehalten.

Signalart	Blaue Linie	Rote Linie
GND	Weiss	Gelb
DIN	Grau	Grün
5 Volt	Violett	Blau
Total Anzahl LEDs	81	91

Von der Topplatte führt eine Steckverbindung auf das Controlboard. Die Belegung des Steckers ist auf der folgenden Abbildung zu sehen. Beide Linien werden über diese Verbindung angeschlossen.

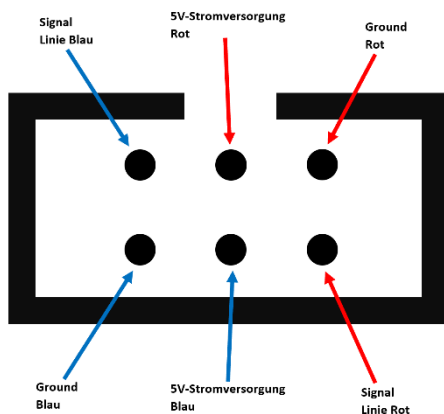


Abbildung 6: Belegung Stecker LEDs

3.9 Verkabelung der Level Shifter für LEDs

Für jede Linie gibt es einen Level Shifter. Dieser wird mit 5 Volt gespeissen. Die Anschlüsse sind auf der Abbildung 3 zu entnehmen.

Der Input 1A kommt von dem Raspberry Pi und ist abhängig davon, auf welche Linie dieser geht. Das Signal wird über 1Y auf die LEDs übertragen.

LED Linie	Anschluss am Raspberry Pi		
Rote Linie	24	GPIO 24 PCM_FS/PWM1	35
Blaue Linie	12	GPIO 1 PCM_CLK/PWM0	1

SN74AHCT125

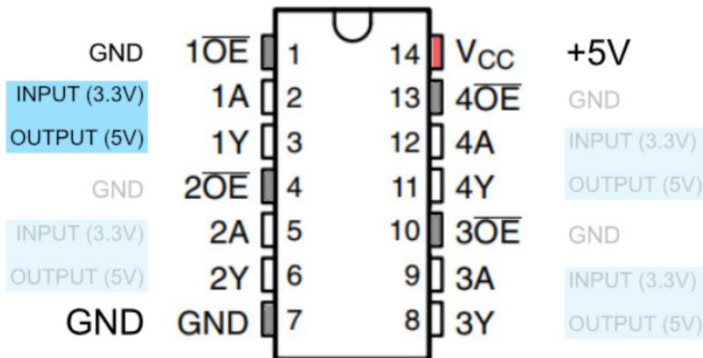


Abbildung 7: Pinout Level Shifter

3.10 Verkabelung der Buttons (Kanten)

Jeder Knoten hat seine eigene Nummer und die sich dazwischen befindenden Kanten sind mit den Nummern der zwei anliegenden Knoten beschriftet.

Die Buttons, auf den Kanten, werden jeweils mit einer I/O-Signalverbindung und einem Ground verbunden. Die zwei I/O-Kabel bilden ein Paar und sind mit der gleichen Nummer beschriftet, wie die Kante selbst.

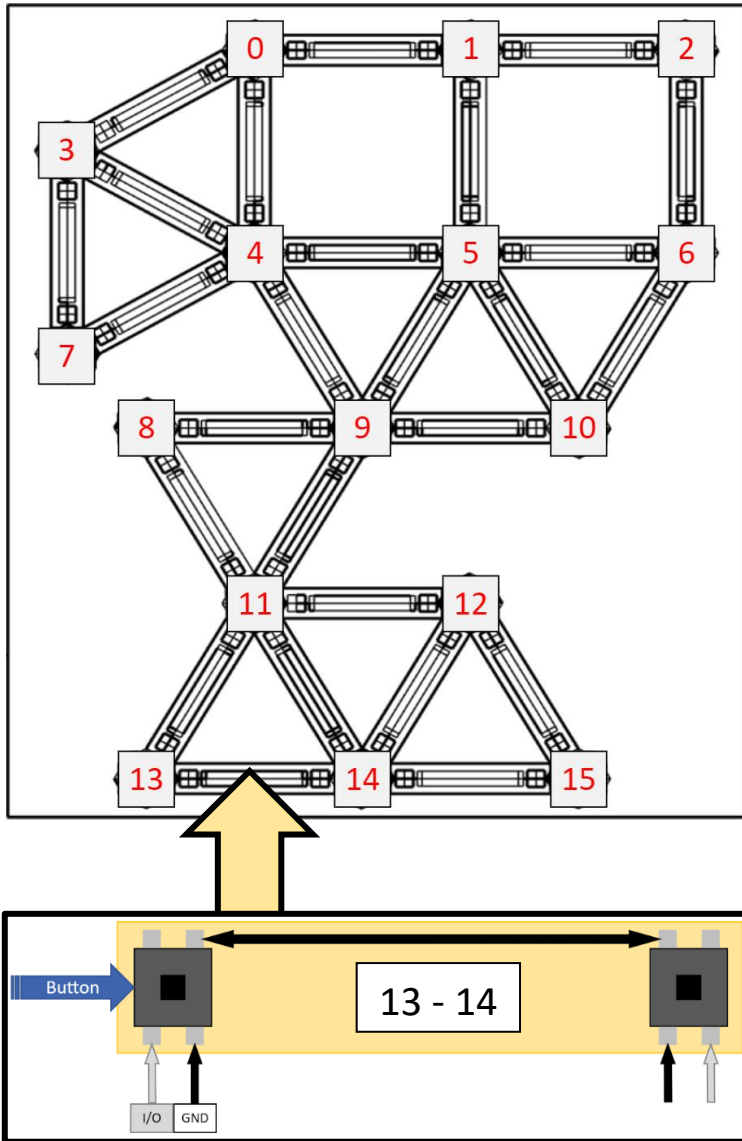
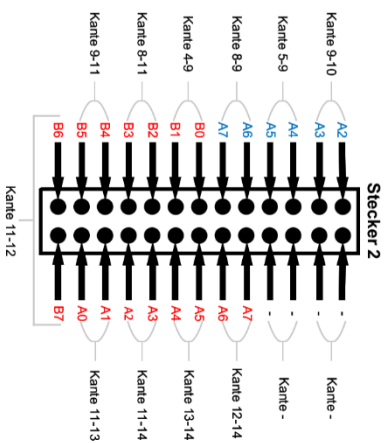
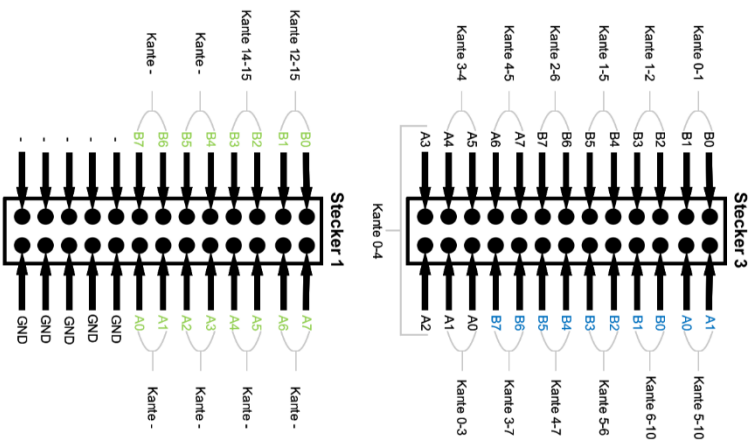


Abbildung 8: Knotenbeschriftung und Kantenaufbau

3.11 I/O Expander & Button Verbindung

Jede Kante hat zwei I/O-Kabel, die auf die Steckverbindung führen. Auf der folgenden Abbildung ist zu sehen, wie sich die Verbindungen zusammensetzen.

Es gibt 3 Stecker mit den Nummern 1, 2 und 3. Jeder hat seine spezifische Belegung und steuert andere Kanten an.



Die Farben geben an, welche Pins es des jeweiligen Expanders sind:

Expander 0

Expander 1

Expander 2

Expander 3

3.12 Verkabelung & ID der I/O Expander

Die Anschlüsse der I/O Expander ist auf der folgenden Abbildung zu sehen.

Die Anschlüsse SCL und SDA werden durchgeschleift und führen auf den gleichbeschrifteten Anschluss des Raspberry Pis.

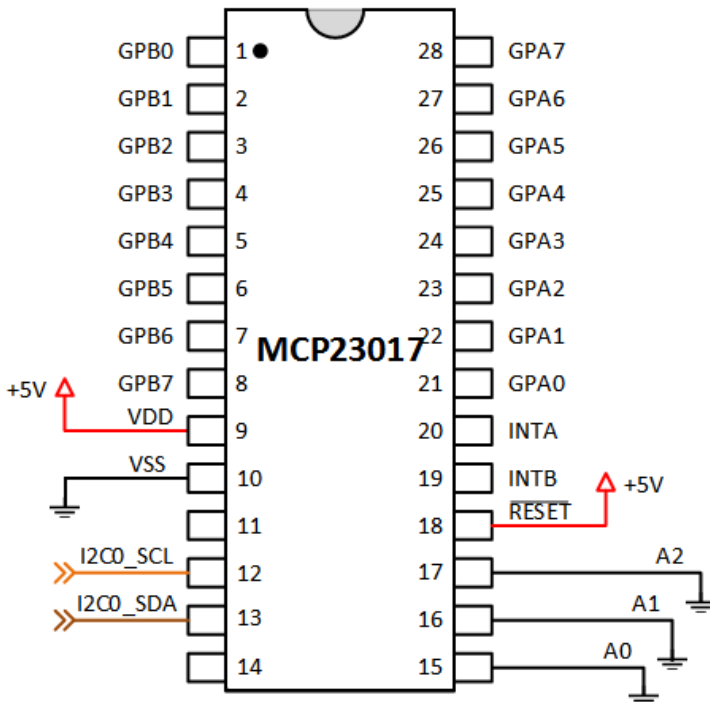


Abbildung 9: Pinout I/O Expander

Jeder Expander hat seine spezifische ID. Diese wird durch die Anschlüsse A0, A1 und A2 und wie diese angeschlossen werden, bestimmt.

Auf der unten aufgeführten Tabelle ist zu sehen, wie welcher Expander angeschlossen ist.

Board ID	A0	A1	A2
0	-	-	-
1	+	-	-
2	-	+	-
3	+	+	-

3.13 Anschlüsse des Raspberry Pi

Raspberry Pi 3 Model B (J8 Header)					
GPIO#	NAME			NAME	GPIO#
	3.3 VDC Power	1		2	5.0 VDC Power
8	GPIO 8 SDA1 (I2C)	3		4	5.0 VDC Power
9	GPIO 9 SCL1 (I2C)	5		6	Ground
7	GPIO 7 GPCLK0	7		8	GPIO 15 TxD (UART)
	Ground	9		10	GPIO 16 RxD (UART)
0	GPIO 0	11		12	GPIO 1 PCM_CLK/PWM0
2	GPIO 2	13		14	Ground
3	GPIO 3	15		16	GPIO 4
	3.3 VDC Power	17		18	GPIO 5
12	GPIO 12 MOSI (SPI)	19		20	Ground
13	GPIO 13 MISO (SPI)	21		22	GPIO 6
14	GPIO 14 SCLK (SPI)	23		24	GPIO 10 CE0 (SPI)
	Ground	25		26	GPIO 11 CE1 (SPI)
30	SDA0 (I2C ID EEPROM)	27		28	SCL0 (I2C ID EEPROM)
21	GPIO 21 GPCLK1	29		30	Ground
22	GPIO 22 GPCLK2	31		32	GPIO 26 PWM0
23	GPIO 23 PWM1	33		34	Ground
24	GPIO 24 PCM_FS/PWM1	35		36	GPIO 27
25	GPIO 25	37		38	GPIO 28 PCM_DIN
	Ground	39		40	GPIO 29 PCM_DOUT

Attention! The GPIO pin numbering used in this diagram is intended for use with WiringPi / Pi4J. This pin numbering is not the raw Broadcom GPIO pin numbers.

<http://www.pi4j.com>

Abbildung 10: Pinout Raspberry Pi

3.14 Lüfter

Im Gehäuse sind zwei Lüfter verbaut, welche über ein USB Netzteil gespeist werden. Über einen Schalter kann man zwischen Off und 3 verschiedenen Stufen der Lüftungsstärke umschalten.

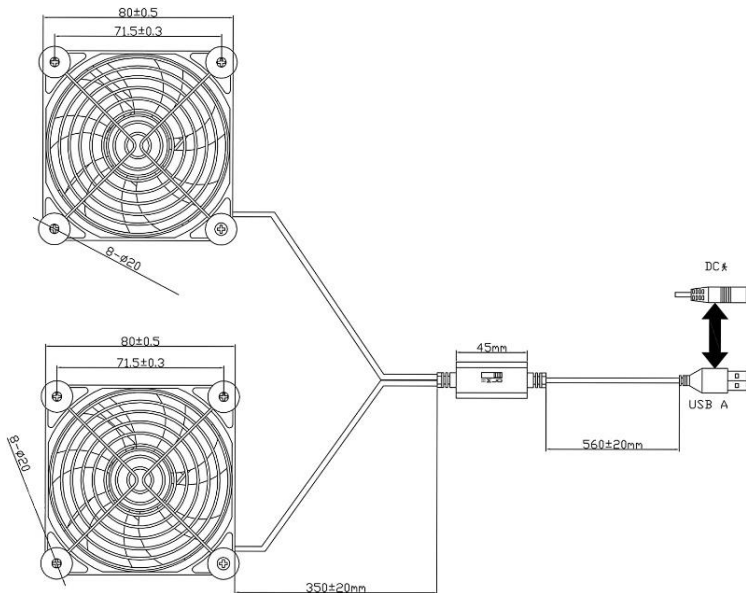


Abbildung 11: Lüfterschema mit Mass

3.15 Raspberry Pi 7-Inch Touch Screen Display

Das Display wird ebenfalls über seinen Micro-USB Anschluss von einem USB Netzteil gespeisen. Das Flachbandkabel wird direkt auf den Raspberry Pi angeschlossen.

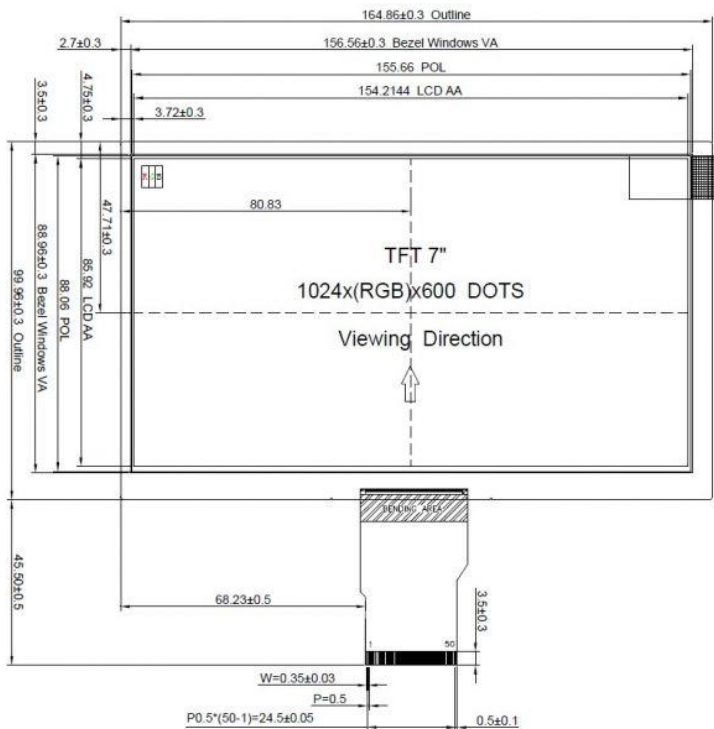


Abbildung 12: Display Mass

4 Software-Komponenten

4.1 Installation

4.1.1 Image auf SD-Karte laden

Benötigt für Installation:

- Raspberry Pi
- Micro-SD-Karte mit mindestens 32GB
- PC oder Laptop, der ein Micro-SD-Karte-Anschluss oder Adapter hat.
- WLAN oder Handy-Hotspot

Der Raspberry Pi läuft auf «Raspberry Pi OS». Der Installer für das Image kann von <https://www.raspberrypi.com/software/> heruntergeladen und installiert werden.

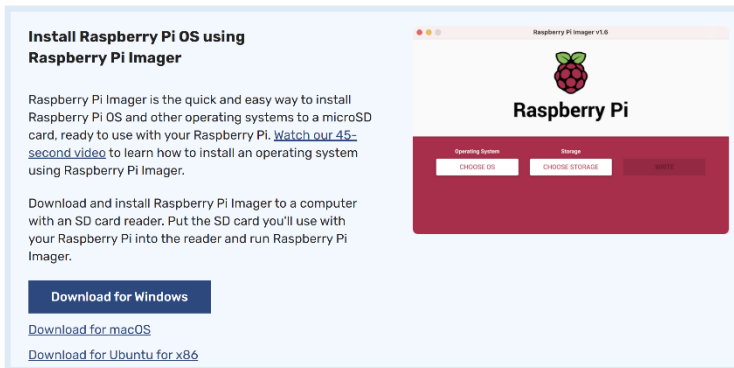


Abbildung 13: Screenshot von Download-Seite

1. Verbinde die Micro-SD-Karte mit dem PC.
2. Starte den Raspberry Pi Imager.
3. Wähle «OS wählen» → «Raspberry Pi OS (other)» → «Raspberry Pi OS (64-Bit)»



Abbildung 14: Screenshot von benötigtem Raspberry Pi OS

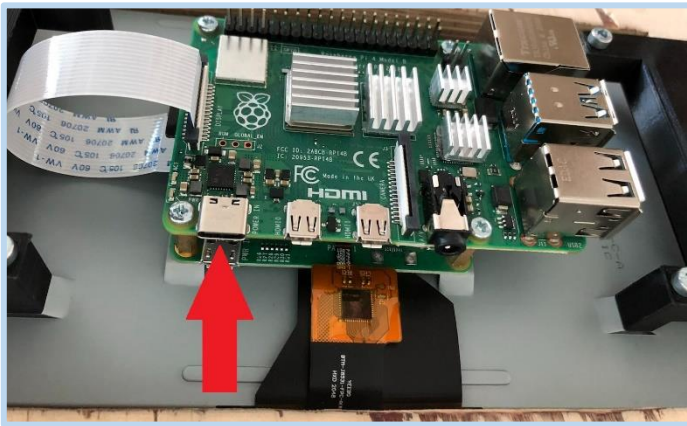
4. Wähle bei «SD-Karte wählen» die SD-Karte.
5. Wähle das Einstellungssymbol unten rechts.
6. Folgende Einstellungen sind einzutragen:
 - i. Hostname: pipe-it-up-pi
 - ii. SSH aktivieren (anwählen)
 - iii. Benutzername: pipeitup
 - iv. Passwort: pipe-it-up!3
 - v. WIFI SSID: [Name des WLANs oder Hotspots]
 - vi. WIFI Passwort: [Passwort des WLANs oder Hotspots]
 - vii. WIFI Land: CH
 - viii. «Speichern» anwählen
7. «Schreiben» anwählen. Das Image wird nun vom Internet heruntergeladen und auf die SD-Karte geschrieben. Nachdem der Prozess fertig ist, kann die SD-Karte vom PC entfernt werden.

4.2 Raspberry Pi starten

1. Die SD-Karte in den Raspberry Pi einlegen.



2. Verbinde den Raspberry Pi über den USB-C-Port mit Strom.



3. Der Raspberry Pi sollte danach hochfahren. Der erste Bootprozess kann mehrere Minuten dauern.
Nach dem Bootprozess sollte der Login-Screen erscheinen.

4.3 Installation von Pipe It Up!

Die nächsten Schritte können über das Display des Spiels (benötigt zusätzlich Tastatur, welche an den USB-Anschlüssen des Raspberry Pi verbunden werden kann) oder per SSH ausgeführt werden. Falls SSH verwendet wird, können folgende Daten angegeben werden:

- Hostname: pipe-it-up-pi (oder IP-Adresse des Raspberry Pi)
- Nutzernamen: pipeitup
- Passwort: pipe-it-up!3

1. Öffne das Terminal
2. Nutze den Befehl
«git clone https://github.com/RononDex/PipeltUp.git» um das Pipe-It-Up Repository zu klonen.
3. Öffne das Verzeichnis mit dem Setupskript mit dem Befehl
«cd pipe-it-up/pi»
4. Führe das Setupskript mit dem Befehl «sudo bash setup.sh» aus.
Das Skript nimmt die nötigen Konfigurationen vor und installiert das Spiel. Dies kann einige Minuten dauern.
5. Das Spiel lässt sich über die Datei «PipeltUp.jar» auf dem Desktop starten.

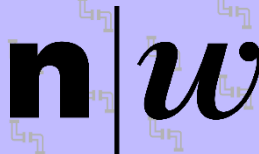
PIPE IT UP!



Hergestellt in Zusammenarbeit durch:

Aaron Bodenmann, Carl von Burg, Clara Tran, Luca Polzner,
Marc Föry, Nicola Liechti, Tino Heuberger, Samir Hauri

Im Auftrag von:



Im Rahmen des IP1/2 2021/22 für Frau Lucia di Caro