

Retro FUTURISTIC ENGINEER

USB  **Ret ro**  **ARDU** **Input**
UNIVERSAL SERIAL BUS



The three-headed
Monkey



Inhalt

1	Einleitung.....	- 1 -
2	Anschlüsse	- 5 -
3	Konfiguration.....	- 6 -
3.1	Erklärung	- 6 -
3.2	Tabellarische Schalter-Übersicht.....	- 7 -
3.2.1	Tastatur	- 7 -
3.2.2	Serielle Maus	- 7 -
3.2.3	Digital-Joystick-Port.....	- 7 -
3.2.4	A-Mouse	- 8 -
3.2.5	Analog-Joystick	- 8 -
3.2.6	Debug-Modus	- 8 -
4	Bekannte Limitierungen	- 9 -
5	Kompatibilitätsliste.....	- 10 -
6	Besondere Tasten-/Achsen-Mappings	- 12 -
6.1	Tastaturen	- 12 -
6.1.1	PC zu Amiga	- 12 -
6.1.2	PC zu Apple	- 12 -
6.2	Joysticks	- 13 -
6.2.1	Speedlink Competition Pro USB	- 13 -
6.2.2	Speedlink Phantom Hawk.....	- 14 -
6.2.3	Logitech Extreme 3D Pro	- 14 -
6.2.4	Logitech F-Serie Gamepads (F310, F710) Direct Input	- 15 -
6.2.5	Thrustmaster TC Sidestick Airbus Edition (evtl. auch FCS Hotas).....	- 16 -
6.2.6	Trio Linker Plus mit Playstation 2 Dual Analogue Controller	- 16 -
6.2.7	Trio Linker Plus mit Dreamcast Controller	- 17 -
6.2.8	Trio Linker Plus mit Gamecube Controller	- 18 -
6.2.9	Default Joystick (sofern es sowas gibt).....	- 18 -
7	Materialliste	- 19 -
8	Zusammenbau	- 23 -
8.1	Vorbereitung des USB Host Shields (Lötbrücken setzen).....	- 23 -
8.2	Bauteile auf der Oberseite des Three-Headed Monkey Shields	- 24 -
8.3	Bauteile auf der Unterseite des Three-Headed Monkey Shields	- 24 -
8.4	Slicing und Druck des 3D-Druck Gehäuses	- 26 -
8.5	Zusammenbau der 3D-Druck-Teile für das Gehäuse	- 27 -
8.5.1	Tipp vorab – Zerlegen und Neueindreihen von Schrauben in Kunststoffgehäusen...	- 27 -

8.5.2	Jetzt aber – Gehäuse zusammenbauen.....	- 27 -
9	Programmieren des Arduinos	- 30 -
10	Informationen zu den Interfaces.....	- 32 -
10.1	CN1: Analog-Joystick	- 32 -
10.2	CN2: PC Serial Mouse	- 33 -
10.3	CN3: C64/Atari Joystick und 1350 Joystick Mouse.....	- 34 -
10.4	CN4: Commodore/Atari Mouse.....	- 35 -
10.5	CN5: Keyboard.....	- 36 -
10.5.1	Timings XT.....	- 37 -
10.5.2	Timings AT und PS/2.....	- 37 -
10.5.3	Timings und Besonderheiten Amiga.....	- 37 -
10.5.4	Timings Tandy 1000.....	- 38 -
10.6	CN6: PS/2 Mouse.....	- 38 -
10.7	CN7: ADB	- 38 -
11	Code-Quellen.....	- 40 -
12	Abbildungsverzeichnis.....	- 41 -

1 Einleitung

Der USB-RetroARDUInput ist ein auf dem Arduino Mega2560 basiertes Projekt zur Übersetzung von modernen USB-Eingabegeräten auf alte Standards.

Das Schweizer Armeemesser unter den Retro-Tools ermöglicht es, immer die passenden Eingabegeräte zur Hand zu haben.

Die Inspiration hierzu stammt von Adrian Black (Youtube: Adrians Digital Basement), der basierend auf einem Arduino einen ähnlichen Wandler für PS/2 zu Tandy 1000 entwickelt hat. Sein Code (Link im Kapitel Quellen) ist auch der, auf dem sich die Tandy-Ausgabe des RetroARDUInput stützt und der, auf dessen Basis ich durch Veränderung von Timings und Ansteuerungsverhalten die anderen Tastaturprotokolle entwickelt habe. Auf die bekannte USB Host Shield Library von felis stützt sich vermutlich jeder, der mit dem Arduino als USB Host arbeitet. Ein paar weitere Code-Snippets, insbesondere im Bereich ADB, wurden aus anderen Open Source Projekten übernommen, sowie jede Menge Eigenarbeit investiert.

Hinzu kam das grundsätzliche Problem, dass alte Eingabegeräte immer seltener werden, einige USB-Eingabegeräte mit älteren PCs mit USB und PS/2 nicht richtig laufen wollten (vorab: Da Gaming-Tastaturen und Mäuse ein paar Besonderheiten aufweisen, die sie zu dem Arduino USB Host Shield inkompatibel machen, war dieses Dilemma so nicht lösbar), aber auch der Traum von der „Eines für Alles“ KVM mit einem HDMI/USB KVM-Switch, einem HDMI Monitor und USB-Eingabegeräten, die mit Hilfe von Adaptern wie dem RGB2HDMI, VGA zu HDMI, MCE2VGA, RetroTink und eben für die Eingabegeräte den RetroARDUInput von den frühen 80ern bis heute jeden Computer über die gleiche Tastatur, die gleiche Maus, den gleichen Joystick und den gleichen Monitor steuerbar machen.

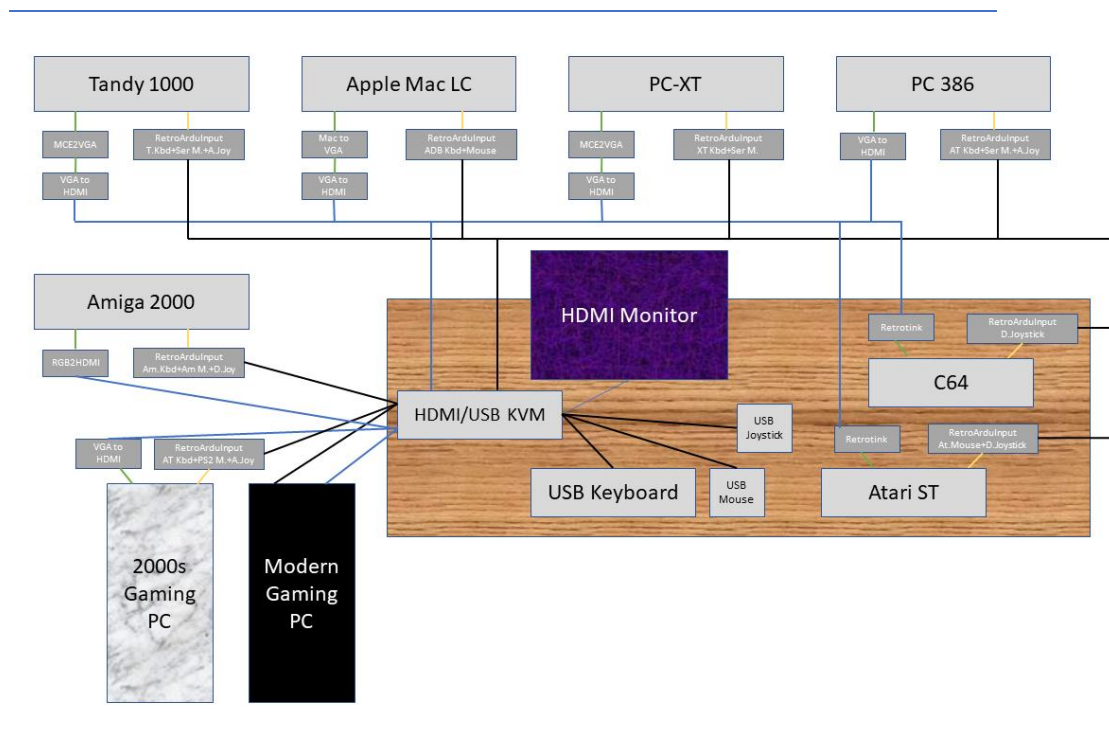


Abbildung 1: Die Vision – Eine KVM für alles!

Ob das je so funktionieren wird, oder ob die Limitierungen des USB Host Shields dies vereiteln, sei mal offen gelassen.

Aber auch so soll das Gerät in jeder Situation hilfreich sein. So werden folgende Retro-Devices emuliert und ausgegeben:

- Tastaturen:
 - o PC/XT
 - o PC/AT und PS/2 inkl. Scancode-Sets 1 (XT-Style), 2 (AT Default) und 3 (für einige PS/2) sowie Typematic Rate und Ansteuerung der Keyboard-LEDs, PS/2 via Adapterkabel DIN->Mini-DIN
 - o Amiga 500/2000/3000, wobei es unwahrscheinlich ist, dass jemand den internen Tastaturanschluss eines Amiga 500 auf den DIN-Stecker adaptiert, für die Desktop-Amigas 2000 und 3000 aber eine sehr sinnvolle Sache
 - o Tandy 1000 (dank mechanischer Pinout-Umschaltung ohne Adapter)
 - o Apple (via ADB)
- Mäuse:
 - o Serielle Maus mit 2-Tasten Microsoft-Modus und 3-Tasten Logitech-Modus
 - o PS/2 mit 3-Tasten-Modus (Scaling und Sample Rate werden derzeit ignoriert)
 - o Amiga / Atari (elektronische Pinout-Umschaltung)
 - o C64 1350 Joymouse (eher eine Spielerei), derzeit keine 1351 Proportional Mouse
 - o Apple (via ADB)
- Joysticks:
 - o Analog-Joystick für PCs (Poti-Ausgang offen)
 - o Analog-Joystick für Tandy TRS-80 CoCo, Tano/Dragon und BBC Micro (Poti-Ausgang gegen Ground, mechanische Umschaltung), benötigen Adapterkabel vom PC-Joystick-Port, auch für den BBC Micro, der zwar den gleichen Stecker, aber ein anderes Pinout hat
 - o Commodore/Atari Digital-Joystick, inkl. Berücksichtigung einer „Dead Zone“ bis zum Auslösen der Achsen bei Verwendung von Analog-Joysticks

Das sollte für jeden reichen, genau wie die berühmten 640kB.

Joystick defekt? Kein Problem, ein USB-Joystick funktioniert nun an jedem PC, Amiga oder C64. Ob analoger Joystick oder der gute alte Competition Pro, alles ist neu als USB-Gerät erhältlich und kann so mit jedem Gerät verbunden werden.

Einen IBM XT, Tandy 1000 oder Desktop-Amiga auf dem Trödelmarkt gekauft und keine passende Tastatur oder Maus zur Hand? Kein Problem, einfach eine USB-Tastatur passend übersetzen.

Der USB-RetroARDUInput besteht aus drei Komponenten. Der Arduino Mega2560 selbst, das USB Input Shield und das sogenannte Three-Headed Monkey Shield. Unverkennbar die Anspielung auf den dreiköpfigen Affen aus „The Secret of Monkey Island“, passend dazu, dass eine Tastatur, eine Maus und ein Joystick jeden der drei Köpfe symbolisieren. Auch, wenn jeder Kopf mehrere Dialekte spricht und somit auch über mehrere Anschlüsse verfügt.

Das Gehäusedesign in meiner persönlichen Farbgebung soll eine kleine Hommage an die ersten PCs von IBM sein, den 5150 PC, 5160 PC-XT und 5170 PC-AT. Diese haben eine hellgraue Front, einen mittelgrauen Deckel (Top/Seite) und eine schwarze Rückwand mit weißer Beschriftung. Daher habe ich mich beim 3D-Druck des Gehäuses für folgende Farben entschieden:

- Mittelgrau (IBM Gehäusedeckel): Geeetech PLA Grau (Mittelgrau)
- Hellgrau (IBM Frontpanel): Amazon Basics PLA Naturgrau (Hellgrau)
- Schwarz (IBM Rückseite): Geeetech PLA Schwarz
- Rot (IBM Schalter): Geeetech PLA Rot, alternativ 3DFils TPU Rot für Soft-Touch Schalter

- Transparent (Akzentuierung, Optional): Geeetech PLA Transparent

Dies kommt den IBM-Farben nicht perfekt aber schon sehr nahe und passt sehr gut zu diversen Computern aus den „Off-White“ 1980er Jahren. Da ich die Front mit transparenter Basis mit graviertem Schriftzug (solide hellgraues Frontpanel mit ausgespartem Schriftzug, der transparent durchscheint), wird die Schrift leuchten, befeuert von 6 LEDs, die, wie es sich für Power-LEDs aus den 80ern und frühen 90ern gehört, natürlich grün sind.

Gedruckt werden:

- Basisgehäuse: Mittelgrau
- Top-Cover und rechtes Cover: Basis Mittelgrau, Wechsel auf Schwarz für Beschriftungen
- Hinteres Cover: Schwarz, Wechsel auf Hellgrau für Beschriftungen
- Linkes Cover (Bauartbedingt unvermeidliche Mulde für den USB-Host-Port, angelehnt an die bei IBM rechts sitzende Schaltermulde): Start mit Rot für die Abdeckung im USB-Port-Bereich, Wechsel auf Schwarz für die Beschriftung des USB-Ports und den Rahmen der Mulde, Wechsel auf Mittelgrau für äußere Blende, Wechsel auf Schwarz für die Beschriftung
- Vorderes Cover „gravierte Schrift“: Start mit Transparent, falls die LED-Beleuchtung eingebaut wird oder Schwarz, falls das Gerät nicht leuchten soll, dann Wechsel auf Hellgrau
- Vorderes Cover „erhabene Schrift“: Start mit Hellgrau, Wechsel auf Schwarz für die Beschriftung (keine Transparenz vorgesehen). Ich empfehle nicht, bei diesem Front-Cover Typen auf eine transparente Basis zu setzen, da die Ausleuchtung vermutlich sehr ungleichmäßig wäre
- Schiebeschalter-Cover, Reset-Schalter Pin und Reset-Schalter Sleeve: Rot

Das Gehäuse entstand in FreeCAD und wird mit Cura gesliced. Gedruckt wird auf einem Creality Ender 3 V2.

Die bekannten China-PCB-Fertiger, die meist auch 3D-Druck anbieten, drucken in SLA-Technik, die im Gegensatz zu FDM keine Farbwechsel erlaubt. Zudem bieten sie meist nur ein Hellgrau an. Wer zuhause mit einem FDM-Drucker unterwegs ist, kann im Prinzip jedes beliebige Farbschema anwenden, günstig aber mit einem größeren Zeitaufwand (je nach Teil 1 bis 8 Stunden Druckzeit).

Falls sich jemand über die Anordnung der Kapitel wundert, und warum die bedienungsrelevanten Kapitel noch vor dem Zusammenbau kommen, obwohl man das Gerät ja eigentlich zuerst zusammenbauen muss: Falls sich in der Community genug Leute finden, die den Kollegen mit weniger Lötkenntnissen fertige Geräte bauen, ist das Kapitel Zusammenbau für die meisten Leute sicher uninteressanter und wurde daher nach den Kapiteln „Konfiguration“, „bekannte Limitierungen“, „Kompatibilitätsliste“ und „Besondere Tasten-/Achs-Mappings“ positioniert. Diese sind für den Alltagsbetrieb interessanter. Für die Nerds unter uns, die sich das Gerät, wie es sich für einen Retro-Nerd gehört, selbst zusammenschrauben möchten, folgt darauf das Kapitel „Materialliste“, „Zusammenbau“ und „Programmierung“, und wer zudem noch etwas über die Interfaces lernen möchte, beispielsweise um aktiv an dem Community-Projekt zu arbeiten und das Produkt weiter zu verbessern, bekommt als Abschluss noch die „Informationen zu den Interfaces“. Zum Schluss, da vieles zwar auf eigenem Hirnschmalz basiert, aber nicht alles, noch eine Quellenangabe zu anderen Projekten, aus denen ich etwas Code übernommen habe.

Anmerkung: Das Projekt startete Mitte 2020 und neben einer Vision und Notwendigkeit auch ein wenig als Ablenkung von einer gewissen Langeweile im „Corona-Lockdown“. Zudem sah ich es als tolle Chance, meine nach fast 15 Jahren in einem Beruf ohne diese Skills etwas eingerosteten Kenntnisse in Hardware- und Software-Entwicklung noch mal etwas aufzufrischen und neue Felder wie PCB-Design und 3D-Druck zu erforschen. Da mein Job aber auch während des Lockdowns

glücklicherweise viel Zeit in Anspruch nahm, war nicht immer viel Freizeit da und somit streckte sich die Entwicklungszeit über mittlerweile doch mehrere Jahre. Zwischenzeitig wurde die Idee von einem anderen Projekt namens „USB4VC“ auf Basis des Raspberry Pi, derzeit mit PC/XT und PC/AT/PS/2 Tastatur und serieller und PS/2-Maus sowie Analog-Joystick auf einem PC-Modul und ADB auf einem Apple-Modul, eingeholt. Da aber Open Source Community-Projekte nicht konkurrieren, sondern sich ergänzen, und ich es immer noch als Lernprojekt für mich sowie als anderen Ansatz (Atmega-Microcontroller auf Arduino statt ARM-SOC auf RasPi) und gute Ergänzung (mehr und andere Protokolle) sehe, mache ich weiter – und hier ist nun das vorläufige Ergebnis, als Community-Banane (eigentlich fertig, muss aber in der Community nachreifen).

Das Parallel-Projekt „USB for Vintage Computers“ oder kurz „USB4VC“ von dekuNukem:

Auf Github: [GitHub - dekuNukem/USB4VC: USB Keyboard/Mouse/Gamepads on Retro Computers!](#)

Auf Kickstarter: [USB4VC: USB Inputs on Retro Computers! by dekuNukem — Kickstarter](#)

2 Anschlüsse

Das Gerät verfügt über folgende Anschlüsse:

Auf dem Arduino:

- USB Typ B für Programmierung, Stromversorgung und Debugging
- Power: Stromversorgung – Im Betrieb gegenüber USB vorzuziehen, gerade wenn USB-Hubs verwendet werden. Polarität Positive Pin/Negative Sleeve. Der Arduino verfügt über einen Spannungsregler und fordert über diesen Port üblicherweise 7-12V ein. Zwar läuft er mit 5V, allerdings springt die Erkennung erst bei 7V an. Darunter wird er per USB, falls ein USB-Programmierkabel angeschlossen ist (meist ausreichend, evtl. aber nicht immer) oder parasitäre Spannungen, wie sie sich hier z.B. über die Referenzspannungen im Joystick-Port einschleichen (reichen definitiv nicht!) versorgt.

Auf dem USB Host Shield:

- USB Typ A – Hier können ein oder mehrere USB-Geräte (per USB-Hub) angeschlossen werden. Es sollte jedoch immer nur ein Gerät eines Typs angeschlossen werden, mehrere Geräte gleichen Typs wie z.B. zwei Joysticks werden nicht unterstützt. Ausnahme: Ein USB Composite HID, üblicherweise Wireless Tastatur/Maus-Sets mit nur einem Receiver-Dongle, können gemeinsam mit einer Maus und Tastatur angeschlossen werden. Die Geräte konkurrieren dann miteinander und liefern beide Eingaben.

Auf dem „Three-Headed Monkey“ Shield:

- Analoger Joystick (PC): Hier werden die Standard X/Y Achse als X/Y-Achse von Joystick 1 und eventuelle Schubregler oder 3D-Funktion (Stick drehen) als X/Y Achse von Joystick 2 übertragen. Für die Tasten (Taste 1 und 2 als Taste 1 und 2 von Joystick 1 und 3 und 4 als 1 und 2 von Joystick 2) gilt das Gleiche. So wurde es auch in frühen Joysticks mit mehr als zwei Achsen oder mehr als zwei Knöpfen umgesetzt, z.B. CH Products Flightstick / Flightstick Pro. Die digitalen Potentiometer lassen sich auf Masse schalten (z.B. BBC Micro, Tandy TRS-80 oder Dragon mit passendem Adapter) oder offen betreiben (Standard beim PC).
- Serielle Maus (PC): Umschaltbar zwischen 2-Tasten Microsoft und 3-Tasten Logitech Protokoll
- Digitaler Joystick (C64, Amiga, Atari): Der Port kann umgeschaltet werden, sodass er statt auf den USB-Joystick auf die USB-Maus hört. Dies soll die Commodore 1350 „Joystick-Maus“ für den C64 emulieren. Eine Implementierung für die 1351 „Proportional Maus“ ist derzeit nicht geplant und würde vermutlich ein größeres Redesign voraussetzen.
- Amiga/Atari Maus: Maus basierend auf den Radimpulsen der optomechanischen Impulsgeber alter Kugelmäuse. Umschaltbar zwischen Commodore (Amiga, PC1, PC10-III/PC20-III/Colt) und Atari ST (hat einige Pins getauscht)
- Tastatur: 8-Pin DIN-Anschluss um volle Kompatibilität zu allen Systemen zu ermöglichen. Mit einem 8-Pin Kabel für den Tandy 1000 und einem 5-Pin Kabel, das mechanisch ebenfalls in diesen Stecker passt, für IBM PC-XT, IBM PC-AT und PS/2 (mechanischer Adapter von DIN auf Mini-DIN erforderlich) und Commodore Amiga aus der Desktop/Tower-Serie
- PS/2 Maus: Standard PS/2 Maus
- ADB: Für ältere Apple-Geräte (Macintosh Classic / Color Classic, LC, Performa, frühe Power Macs und Ähnliche)

3 Konfiguration

3.1 Erklärung

Das Gerät verfügt über folgende Schalter:

- SW01-SW04 und SW05: SW01 und SW04 schalten mechanisch das Pin-Layout des Tastatur-Anschlusses um. In der Position 1-2 werden die Pins für Tandy 1000 ausgerichtet. In der Position 2-3 für PC (XT/AT/PS/2), Amiga und Apple (Ausgabe über ADB). Das Pin-Layout des Tandy 1000 weicht so stark vom PC-Standard ab, dass eine Umschaltung im Arduino nicht möglich war (5V und Masse-Leitungen). SW05 schaltet zwischen Tandy und PC/Amiga Protokoll um. Ein großer, alle fünf Schalter umfassender 3D-Druck-Hebel soll dazu beitragen, dass alle Schalter immer gleichzeitig umgelegt werden. Steht SW05 in Position 1-2 (Tandy), werden SW06 und SW07 ignoriert.
- SW06: PC/Amiga+Apple Umschalter: Stehen SW01-SW05 in Position 2-3 (PC/Amiga) wählt dieser Schalter zwischen Amiga/Apple (1-2) und PC (2-3). SW07 wählt dann zwischen PC-XT/AT und Amiga/Apple
- SW07: XT/AT Schalter: Schaltet im PC-Modus (SW01-SW05 und SW06 in Position 2-3) zwischen PC-XT (Position 1-2) und PC-AT und PS/2 (Position 2-3) bzw. im Amiga/Apple-Modus zwischen Amiga (Position 1-2) und Apple ADB (Position 2-3) um, je nachdem, ob bei SW06 der PC oder Amiga/Apple Modus vorgewählt ist. Im ADB-Modus wird der DIN-Tastatur-Port deaktiviert und ADB gibt Tastatur und Maus aus. In jedem anderen Modus gibt der DIN-Port Tastatursignale des gewählten Systems aus und ADB bleibt deaktiviert.
- SW08: Serielle Maus Modus: 1-2 (3 Button Logitech) und 2-3 (2-Button Microsoft)
- SW09: Digital Joystick Modus: 1-2 (C64 Commodore 1350 Joystick Maus) und 2-3 (Digital-Joystick für C64, Amiga und Atari). Der Port macht das Gleiche, er hört allerdings im 1350-Modus auf die Maus und nicht auf den Joystick.
- SW10: Amiga/Atari-Maus: 1-2 (Atari, einige Pins gedreht, passiert im Arduino) und 2-3 (Commodore Amiga, Commodore PC1/PC10-III/20-III/COLT)
- SW11: Debug: Position 1-2 Debug Ein, wartet nach dem Einschalten auf den seriellen UART der Debugging-Konsole der Arduino Entwicklungsumgebung und gibt dann Meldungen aus, 2-3 Debug Aus, Gerät initialisiert (dauert einige Sekunden) und ist dann auch ohne serielle Konsole einsatzbereit. Keine Ausgabe über die serielle Konsole
- SW12-SW15: Joystick-Potentiometer gegen Masse gezogen (Position 1-2) für Kompatibilität zu Geräten ohne Widerstand am Eingang (Tandy TRS-80, Dragon) oder am Ende offen (Position 2-3) für Geräte mit Widerstand am Eingang (PC Joystick Interface)
- Reset: Hier wurde für den Reset-Knopf auf dem USB Host Shield ein Loch gelassen, durch das ein 3D-Druck Knopf gesteckt werden kann, um so den Reset im Notfall auch von außen betätigen zu können.

3.2 Tabellarische Schalter-Übersicht

3.2.1 Tastatur

Betrifft: CN5 – Keyboard – DIN-5/180 bzw. DIN 8/270 und CN7 – ADB – Mini-DIN-4

1 = An bzw. Rechts, 0 = Aus bzw. Links, / = Schalter ignoriert

Großer Schalter Tandy					PC/A*	AT/XT		
SW01	SW02	SW03	SW04	SW05	SW06	SW07	Modus	Anschluss
0	0	0	0	0	0	0	AT, PS/2 (mit Adapter)	DIN-5/8
0	0	0	0	0	0	1	PC/XT	DIN-5/8
0	0	0	0	0	1	0	Apple	ADB
0	0	0	0	0	1	1	Amiga	DIN-5/8
1	1	1	1	1	/	/	Tandy 1000	DIN-5/8

Bemerkung: Nur Tandy 1000 nutzt die Pins 6 bis 8 des 8-Pin-DIN-Anschlusses, wobei die dort liegenden Pins nicht relevant sind, Verwendung eines Lötpin-kompatiblen 5-Pin DIN-Anschlusses ist möglich. Die Anordnung der Pins 1-5 ist beim Tandy 1000 anders, als bei den pinkompatiblen IBM PC/XT, AT, PS/2 (mit Adapter DIN auf Mini-DIN) und Desktop-Amigas.

Eine Verwendung des Tandy-Pinout an einem anderen System oder des PC/Amiga-Pinouts am Tandy funktioniert nicht und kann im schlimmsten Falle zur Beschädigung der Hardware führen. Daher müssen bei Verwendung eines Tandy 1000 immer SW05 (Protokoll) sowie SW01-SW04 (Pinout) gleichermaßen auf 1 stehen.

Im Apple-Modus wird der Speicherbereich für die Scancode-Tabelle für die Apple-Scancodes belegt und die Tastaturbefehle werden über ADB ausgegeben und zudem auch die ADB-Maus aktiviert. Der DIN-Anschluss bleibt hier ohne Funktion. In jedem anderen Modus werden die Tastaturbefehle über den DIN-Anschluss ausgegeben. Die ADB-Maus bleibt in den PC/Amiga/Tandy Modus inaktiv.

3.2.2 Serielle Maus

Betrifft: CN2 – PC Serial Mouse 9-Pin D-SUB Female

1 = An bzw. Rechts, 0 = Aus bzw. Links

SW08	Modus
0	Serielle Maus in Microsoft 2-Button Protokoll
1	Serielle Maus in Logitech 3-Button Protokoll

3.2.3 Digital-Joystick-Port

Betrifft: CN3 – Commodore/Atari Joystick oder C64 1350 Joystick Mouse – 9-Pin D-SUB Female

1 = An bzw. Rechts, 0 = Aus bzw. Links

SW09	Modus
0	Joystick – Setzt USB-Joystickbefehle (mit Deadzone für analoge Joysticks) in Digital-Joystick um
1	1350 – Setzt USB-Mausbefehle in Digital-Joystick um (C64 „1350 JoyMouse“)

3.2.4 A-Mouse

Betrifft: CN4 – Commodore/Atari Mouse – 9-Pin D-SUB Female

1 = An bzw. Rechts, 0 = Aus bzw. Links

SW10	Modus
0	Commodore-Pinout. Kompatibel zu Commodore Amiga, COLT, PC-1, PC10-III, PC-20-III, PC-30-III
1	Atari-Pinout. Kompatibel zur Atari ST Serie

3.2.5 Analog-Joystick

Betrifft: CN1 – Analogue Joystick – 15-Pin D-SUB Male

1 = An bzw. Rechts, 0 = Aus bzw. Links

Schalter Joystick 1		Schalter Joystick 2		Modus
SW12	SW13	SW14	SW15	
0				Potentiometer-Ausgang offen für: <ul style="list-style-type: none">- PC (natives Pinout, kein Adapter)
1				Potentiometer-Ausgang auf GND für <ul style="list-style-type: none">- BBC Micro (mit Adapter, anderes D-SUB 15 Pinout)- Tandy TRS-80 CoCo (mit Adapter)- Tandy 1000 (mit Adapter)- Tano/Dragon (mit Adapter)

Da der Schalter nur pro Joystick-Achse (oder für den gesamten Joystick-Port) Sinn macht, wurde in Betracht gezogen, entweder alle 4 Schalter (verworfen) oder zumindest die 2 Schalter für die X- und Y-Achse eines einzelnen Joysticks (also SW12+SW13 und SW14+SW15) umzulegen, daher wurden in dem Gehäuse Schalterkappen mit Überlappung für je 2 Schalter verbaut

3.2.6 Debug-Modus

1 = An bzw. Oben, 0 = Aus bzw. Unten

SW11	Modus
0	Aus: Keine Initialisierung der USB-Serial-Schnittstelle, keine Ausgabe von Debug-Informationen über serial.println
1	An: Initialisiert die USB-Serial-Schnittstelle und wartet auf Kommunikation mit der Arduino IDE. Gibt Debug-Informationen über serial.println aus, sofern diese vom Programmierer eingegeben wurden

Da der Schalter für den Alltagsbetrieb nicht relevant ist, wurde er nicht über eine Schalterkappe nach außen geführt. Zum Umschalten muss die rechte Gehäusewand, an der die Tastaturanschlüsse liegen, abgenommen werden.

Das Warten auf den USB-UART „Serial“ kann dazu führen, dass kritische Timings auseinanderlaufen. Die Idee war daher, die globale Variable debugMessage an die Schalterposition zu knüpfen und immer per if-Befehl abzufragen, bevor Aktionen auf dem UART durchgeführt werden (Verbindung aufsetzen, Serial.println) und durch diesen Schalter bequem zwischen stabilem Alltagsbetrieb und Debugging per Nachrichten umschalten zu können.

4 Bekannte Limitierungen

Von folgenden Dingen ist bekannt, dass sie nur eingeschränkt oder nicht funktionieren:

- USB-Hubs: Funktioniert nicht in jeder Kombination. Stromzufuhr für den Arduino über den 7-12V Barrel Jack (Polarität: Positive Pin/Negative Sleeve) statt über USB-Programmier-Port und Verwendung eines USB-Hubs mit eigener Stromversorgung empfohlen. Inkompatibilitäten mit bestimmten Hubs sind Limitierungen der Library, evtl. müssen ein paar Hubs ausprobiert werden.
- Mäuse: Die Library unterstützt nicht das Auslesen des Mausrades, daher ist auch eine Weitergabe an PC-Mäuse nicht möglich, obwohl sowohl PS/2 wie auch sogar Seriell im Logitech-Modus dies hergeben würden.
- Gaming-Eingabegeräte mit steuerbarer RGB-Beleuchtung: Diese verhalten sich nicht immer wie ein normales USB-HID. Das kann dazu führen, dass die USB-Library diese nicht erkennt. Diese Geräte sind aber auch zu älteren PCs nicht 100% kompatibel, so funktioniert z.B. die Corsair K68 Tastatur nicht mit dem Arduino USB Host Shield, aber auch Boards wie das Asus P5KR (Sockel 775) reagieren im BIOS nicht darauf, selbst wenn die Tastatur in den BIOS-Modus versetzt wird. Mehr dazu in der Kompatibilitätsliste.
- Joystick: Da derzeit keine automatische Erkennung funktioniert (es wird daran gearbeitet, die Limitierungen der Host Shield Library behindern aber mehr, als zu helfen) und in der aktuellen Revision keine manuelle Umschaltung vorgesehen ist, muss zum Wechsel des Joysticks die Firmware geändert (JoyType in der globals.cpp basierend auf den #define Werten in der globals.h) und neu hochgeladen werden.
- Joystick: Jeder Joystick hat ein anderes Mapping oder gar unterschiedliche Auflösungen. Es wird versucht, so viele Joystick-Typen wie möglich zu implementieren. Ist ein Joystick nicht vorhanden und funktioniert nicht mit dem Default-Joystick-Template, muss er entsprechend implementiert werden.
- Aufgrund des limitierten Speichers des Mega2560 können keine zwei Scan Code Mappings gleichzeitig genutzt werden. Deshalb ist die Nutzung einer Apple ADB Tastatur explizit zu wählen. Der DIN Tastatur-Port wird dann deaktiviert. Konsequenterweise läuft auch die ADB-Maus nur im Apple-Tastatur-Modus, um Rechenleistung zu sparen und die ADB-Libraries in jedem anderen Modus komplett deaktiviert zu lassen.
- Die Tastatur-Modi XT, Tandy 1000 und Amiga sehen keine bidirektionale Kommunikation zwischen Host und Eingabegerät vor. Daher kann der Arduino nur den Status der Caps Lock, Num Lock und Scroll Lock LEDs vermuten. Bei fehlender 5V-Versorgung am Tastaturport werden die LEDs abgeschaltet (Computer aus), um mit dem korrekten Status zu starten, bei Abstürzen mit Soft Reset hat der Rechner evtl. einen anderen Lock-Status, als die Tastatur. Dies ist jedoch normal, „echte“ Tastaturen für diese Rechner haben das gleiche Problem.

5 Kompatibilitätsliste

- Eingabegeräte:
 - USB-Hubs:
 - Generell: Einige Hubs funktionieren nur dann reproduzierbar, wenn ein eingestecktes Gerät ein USB-Wireless-Receiver eines Logitech Tastatur/Maus Sets (hier: K270) ist, andere gar nicht
 - Composite HIDs (Maus/Tastatur Sets, meist Wireless über einen Receiver-Dongle):
 - Logitech K270 Wireless Maus/Tastatur **PASS**
 - Tastaturen:
 - Corsair K68 (Gaming Tastatur mit roter Beleuchtung, auch nicht im BIOS-Modus) **FAIL**
 - Microsoft Wired Keyboard 600 **PASS**
 - Mäuse:
 - Logitech G502SE HERO (Gaming Maus mit RGB) **FAIL**
 - Microsoft Basic Optical Mouse **PASS**
 - Logitech MX500 **PASS**
 - Joysticks:
 - Speedlink Phantom Hawk (Inkl. Kalibrierung aufgrund ungenauer Zentrierung ab Werk) **PASS**
 - Speedlink Competition Pro USB (nicht reproduzierbar gelegentliche Abstürze mit meinem Competition Pro, ggf. ein Hardware-Defekt, läuft aber auch teilweise Stundenlang durch) **PASS**
 - Trio Linker Plus (Playstation 1+2, Dreamcast, Gamecube -> USB)
 - Playstation 2 Dual Analogue Controller **PASS**
 - Dreamcast Controller **PASS**
 - Gamecube Controller (Kann ich nicht testen, vermutlich gleiches Mapping wie PS2 Dual Analogue) **Partial Pass**
 - Logitech Extreme 3D Pro **PASS**
 - Thrustmaster TC Sidestick Airbus Edition (vermutlich auch FCS Hotas) **PASS**
 - Logitech F-Serie Gamepads (gelegentlich Verbindungsprobleme mit dem F710 Wireless, aber auch das funktioniert, kabelgebundene Gamepads wie das F310 nutzen das gleiche Mapping) – Achtung, Schalter am Gamepad muss auf D (DirectInput) stehen, X (XInput) funktioniert nicht **PASS**
- Zu steuernde Retro-PCs:
 - Tastatur-Interface:
 - IBM PC/XT Klasse:
 - Commodore PC20-II **PASS**
 - IBM PC/AT und PS/2 Klasse:
 - Headland HT12 286: LEDs in CheckIt nicht richtig gesetzt, sonst okay **Partial Pass**
 - KMC-A419-3 386: LEDs in CheckIt nicht richtig gesetzt, sonst okay **Partial Pass**
 - Tyan Trinity S1590: LEDs von BIOS und in CheckIt nicht richtig gesetzt, sonst okay **Partial Pass**
 - Commodore Amiga Desktop/Tower und Amiga 500 mit Adaptierung auf Platinenstecker:
 - Alle Tasten funktionieren, Reset (Ctrl+Amiga+Amiga) und Selbsttest/Handshake sind implementiert, aber A2000 spezifische

Besonderheiten und mit A500 nicht testbar **PASS** (Amiga 500)/**Ungetestet** (Amiga 2000)

- Tandy 1000:
 - Adaptierung von PS/2 to Tandy von Adrian Black vollständig implementiert, aktuell keine Möglichkeit zu testen **Ungetestet**
- Serielles Interface (Mainboard/Maustreiber):
 - Im 2-Button Microsoft Modus:
 - Commodore PC20-II / Cutemouse **PASS**
 - Headland HT12 286 / Cutemouse **PASS**
 - KMC-A419-3 386 / Cutemouse **PASS**
 - Im 3-Button Logitech Modus:
 - Commodore PC20-II / Cutemouse **PASS**
 - Headland HT12 286 / Cutemouse **PASS**
 - KMC-A419-3 386 / Cutemouse **PASS**
- PS/2 Maus Interface (Mainboard/Maustreiber):
 - Tyan Trinity S1590: Funktioniert mit Logitech-Treiber aber nicht mit Cutemouse und Windows **Partial Pass**
- Analoger Joystick:
 - ISA-Karten (5V Referenz)
 - Creative Labs Soundblaster 2.0, Aztech Sound Galaxy NXII (CPS Soundblaster 2.5, diverse Multi-I/O Karten) **PASS**
 - PCI-Karten (3.3V Referenz)
 - **Test Offen**
 - Diverse Heimcomputer, die das Poti auf GND benötigen
 - **Test offen**
- Commodore/Atari Maus:
 - Amiga 500 **PASS**
 - Atari ST (**PASS? Ungetestet mangels Gerät, jedoch reine Pinout-Änderung**)
- Commodore/Atari Joystick:
 - Joystick-Modus (reagiert auf USB-Joystick):
 - Amiga 500 **PASS**
 - 1350 „Joymouse“ Modus (reagiert auf USB-Maus):
 - Amiga 500 als Joystick, keine C64 Software verfügbar **PASS**
- Mac Classic und Co (ADB Maus und Tastatur):
 - **Implementierung Offen**

6 Besondere Tasten-/Achsen-Mappings

6.1 Tastaturen

PC-Tastaturen sind PC-Tastaturen, unabhängig vom Interface und das Layout unterscheidet sich bei PS/2 und USB nicht. Hier ergeben sich keine Besonderheiten. Beim XT waren die rechten Alt und Ctrl Tasten nicht vorgesehen, daher findet eine Doppelnutzung der Scancodes statt (Linke und rechte Tasten auf der USB-Tastatur verwenden den gleichen XT Scancode). Ansonsten gibt es keine Besonderheiten.

Spezieller wird es beim Amiga und Apple.

6.1.1 PC zu Amiga

Einige Tasten des Amigas existieren nicht auf der PC-Tastatur und umgekehrt. Daher wurde eine möglichst praxisnahe Belegung gewählt, um alle Tasten abbilden zu können.

- Modifier-Tasten:
 - o Alt-Links und Alt-Rechts 1:1 umgesetzt
 - o Ctrl kann über beide Ctrl-Tasten der USB-Tastatur ausgelöst werden
 - o Closed A (linke Amiga-Taste): Linke Windows-Taste
 - o Open A (rechte Amiga-Taste): Kann über rechte Windows-Taste oder Kontextmenü-Taste ausgelöst werden, da nicht mehr jede Tastatur zwei Windows-Tasten hat
- Obere Zahlenreihe:
 - o Backslash/Pipe links von der Backspace-Taste umgelegt auf die „Einfügen“-Taste des Inverse-T Pads
- Del/Help:
 - o Del bleibt auf der Entfernen-Taste des Inverse-T Pads
 - o Help: Umgelegt auf Pause/Break
- 10-Key Pad:
 - o Open Bracket auf der Bild-Hoch Taste des Inverse-T Pads
 - o Closed Bracket auf der Bild-Runter Taste des Inverse T-Pads

6.1.2 PC zu Apple

Einige Tasten der ADB-Tastatur existieren nicht auf der PC-Tastatur und umgekehrt. Daher wurde eine möglichst praxisnahe Belegung gewählt, um alle Tasten abbilden zu können.

- Power-Taste:
 - o Power: Wurde auf Print Screen/System Request gelegt. Der Apple-Scancode für Print Screen/F13 steht in den Source Code Kommentaren, falls jemand diese Taste dort hin legen möchte und stattdessen die Power-Taste einer Multimedia-Tastatur verwenden möchte
- Weitere F-Tasten:
 - o Scroll/F14 = Scroll
 - o Pause/F15 = Pause
- Modifier-Tasten:
 - o Option-Links = Alt-Links
 - o Alt Option-Rechts = Alt-Rechts/AltGr
 - o Command Links/Rechts = Control Links/Rechts
 - o Apple Links/Rechts (gespiegelt auf der Apple-Tastatur) = Windows Links/Rechts
- 10-Key Pad
 - o -=Taste = Windows Kontextmenü (da kein Bedarf zum Spiegeln der Apple-Taste erforderlich ist)

6.2 Joysticks

Alle Joysticks haben zwei Achsen und zwei Tasten. Mehr als X/Y und Tasten 1 und 2 verwendet der digitale Joystick-Port für Amiga/Atari auch nicht. Beim PC zog irgendwann die Möglichkeit ein, die Achsen und Tasten von Joystick 2 für bis zu 4 Joystick-Achsen und Tasten zu verwenden. In den letzten Tagen des analogen Gameports gab es auch Codierungen von mehr als 4 Tasten. Jedoch habe ich diese nicht implementiert, da diese auch eher typisch für die Windows 95 Ära sind und in DOS-Spielen eher schwierig zu nutzen. Daher werden die unzähligen Tasten, die heutige Joysticks bieten, strategisch umgeleitet, um in jeder Handposition ergonomisch und logisch als Tasten 1-4 des Analog-Joysticks nutzen zu können.

Alle implementierten Joysticks befinden sich in meinem Besitz. Ich ziele darauf ab, die Sammlung künftig stetig zu erweitern und ebenfalls 8-Bit-Do Gamepads oder ältere USB-Joysticks wie Logitech Wingman oder Microsoft Sidewinder einzupflegen und nehme natürlich auch gerne Quellcode-Erweiterungen von anderen Community-Mitgliedern mit neuen Joysticks an.

Ob es beispielsweise sinnvoll ist, einen Microsoft Sidewinder, der mit 15-Pin Joystick-Port und USB betrieben werden kann, zu implementieren, darüber kann man streiten. Sollten wir aber an den Punkt der „Eine KVM für Alles“ Implementierung kommen und jemand möchte den Sidewinder über seinen USB KVM-Switch vom aktuellsten Gaming-PC bis zum ältesten Retro-Gerät an allem nutzen, dann brauchen wir natürlich auch eine Sidewinder-Implementierung, sollte diese vom Standard-Joystick abweichen. Im Zweifelsfall könnte man für das „Eine KVM für Alles“ Setup sogar darüber nachdenken, den Joystick-zu-USB Adapter, also den umgekehrten Weg, von Necroware ([GitHub - necroware/gameport-adapter: GamePort adapter to connect old DB15 joysticks to USB port](https://github.com/necroware/gameport-adapter)) mit zu implementieren, sofern dieser besondere Implementationen benötigt, um so beispielsweise einen alten Gravis-Joystick in seinem all-umfassenden Computer-Setup nutzen zu können.

6.2.1 Speedlink Competition Pro USB

Mapping „Straight Forward“: Die Tasten befinden sich in der traditionellen Haltung oberhalb von „Up“, die Tasten sind von links nach rechts große runde Tasten 1 und 2 und kleine, eckige Tasten 3 und 4.



Abbildung 2: Speedlink Competition Pro

6.2.2 Speedlink Phantom Hawk

Der Phantom Hawk ist ab Werk schlecht zentriert und muss kalibriert werden. Dies erfolgt automatisch für die X, Y und Twist Achse, wenn das Phantom Hawk Profil ausgewählt wurde.

Der Phantom Hawk hat seine zweite Taste als untere Trigger-Taste für den kleinen Finger unterhalb des Zeigefinger-Triggers Taste 1. Da ich dies im Retro-Bezug nicht intuitiv finde – Taste 2 an einem Flightstick hat mit dem Daumen bedient zu werden –, habe ich die Tasten umgelegt. Der Trigger für den kleinen Finger ist nun Taste 3 und der Daumen-Trigger auf der Stick-Rückseite Taste 2. Der Hat links oben wird ignoriert, der mittige 2-Wege-Schalter und Tasten in der Joystick-Basis wurden jeweils auf Tasten 1-4 gelegt, wobei eine Limitierung ist, dass sich die Standfuß-Taste oben links über USB ebenfalls als Taste 1 meldet und so einen Teil des Mappings vorgegeben hat.

X und Y sind bezogen auf X1 und Y1, Twist ist X2 und Throttle Y2.



Abbildung 3: Speedlink Phantom Hawk

6.2.3 Logitech Extreme 3D Pro

Der Logitech Extreme 3D Pro hat eine höhere Auflösung und braucht daher ein eigenes Profil.

Der Logitech Extreme 3D Pro ist recht direkt belegt. Der Hat wird ignoriert. Tasten in der Joystick-Basis wurden wieder auf 1-4 gelegt.

X und Y sind bezogen auf X1 und Y1, Twist ist X2 und Throttle Y2.



Abbildung 4: Logitech Extreme 3D Pro

6.2.4 Logitech F-Serie Gamepads (F310, F710) Direct Input

Beim Logitech F310 und F710 muss der Modus-Schalter auf D (Direct Input) stehen. Der Modus X (X-Input) funktioniert nicht.

Die Mode-Taste schaltet das Verhalten von D-Pad und linkem Analog-Stick um. Ist die Mode-LED aus, wird der linke Analog-Stick benutzt und das D-Pad ignoriert, ist die Mode-LED an, wird das D-Pad genutzt und der linke Analog-Stick ignoriert. Der rechte Analog-Stick arbeitet immer analog.

Die Tasten sind im Kreis angeordnet 1, 2, 3 und 4, angefangen bei X (Blau)=1, A (Grün)=2, B (Rot)=3 und Y (Gelb)=4. Zudem werden die Taste 1, 2, 3 und 4 durch die Zeigefinger/Mittelfinger-Trigger an der Stirnseite des Pads (nicht im Foto zu sehen) repliziert, mit den Tasten 1 und 3 links (1=Zeigefinger-Button, 3=Mittelfinger-Trigger) und 2 und 4 rechts (2=Zeigefinger-Button, 4=Mittelfinger-Trigger).



6.2.5 Thrustmaster TC Sidestick Airbus Edition (evtl. auch FCS Hotas)

Der Thrustmaster TC Sidestick hat eine höhere Auflösung und braucht daher ein eigenes Profil.

Der Thrustmaster TC Sidestick ist recht direkt belegt mit Tasten 1-3, wobei die Ausrichtung der Tasten 2 und 3 vom Links/Rechts-Schalter des Joysticks abhängt. Der Stick empfindet den Sidestick des Airbus Fly-By-Wire Systems nach, welcher beim Captain (Sitz links) links und beim First Officer (Sitz rechts) an der rechten Seite sitzt. Die seitlichen Tasten sind den Sitzpositionen angepasst, eine Tatsache, der dieser Joystick mit umschaltbarer Logik sowie austauschbaren Seitentasten Rechnung trägt. Der Hat wird ignoriert. Tasten in der Joystick-Basis wurden wieder auf 1-4 gelegt. Der vermutlich ebenfalls compatible FCS Hotas hat 4 Tasten am Stick, diese sollten korrekt durchverbunden sein (nicht getestet).

Der Airbus Sidestick aktiviert beim vollständigen Zurückziehen des Throttles zudem einen weiteren Button (für Flugsimulatoren: Thrust Reverser aktivieren), die aber nicht ausgewertet wird.

X und Y sind bezogen auf X1 und Y1, Twist ist X2 und Throttle Y2.

Ein weiteres Plus in Bezug auf Retro: Absolute Oldschool-Gamer, die noch aus der Ära vor den wilden Microsoft Sidewinder und Logitech Wingman Joysticks kommen, werden vielleicht der Twist-Achse nichts abgewinnen können. Diese lässt sich beim Airbus Sidestick mechanisch sperren, da ambitionierte Hobbypiloten wohl eher mit einem echten Schubhebel-Set und Pedalen fliegen, als den einfachen Throttle und Twist zu missbrauchen. Somit ist der Thrustmaster der einzige moderne USB-Flightstick, der noch klassisch ohne Twist-Achse nutzbar ist.



Abbildung 6: Thrustmaster TC Sidestick

6.2.6 Trio Linker Plus mit Playstation 2 Dual Analogue Controller

Beim Playstation 2 Controller hängt die Belegung der Achsen davon ab, ob der Analog-Modus aktiviert ist.

USB-RetroARDUInput

Open Source USB to Retro Computer Adapter

Bei abgeschaltetem Analog-Modus (LED aus) werden nur X1/Y1 gesteuert, und zwar über das Steuerkreuz mit Werten im Vollanschlag. Im Analog-Modus macht der linke Stick X1/Y1 und der rechte Stick X2/Y2.

Die Tasten 1 bis 4 liegen einmal auf den Feuertasten rechts, und zwar im Uhrzeigersinn von oben aus, mit 1 auf dem grünen Dreieck, 2 auf dem roten Kreis, 3 auf dem blauen X und 4 auf dem magentafarbenen Quadrat. Zudem liegen die Tasten noch auf den Zeigefinger-Triggern, auf den unteren größeren Tasten 1 (links) und 2 (rechts) und auf den oberen, kleineren Tasten 3 (links) und 4 (rechts).



Abbildung 7: Playstation 2 Dual Analogue / Trio Linker Plus

6.2.7 Trio Linker Plus mit Dreamcast Controller

Beim Dreamcast-Controller sind die Achsen X1/Y1 doppelt belegt und können entweder über den Analog-Stick analog oder über das D-Pad digital (Vollanschlag) gesteuert werden. X2 findet sich am linken Zeigefinger-Trigger, Y2 am rechten Zeigefinger-Trigger.

Die Tasten 1, 2, 3 und 4 finden sich auf dem Tasten-Pad rechts, beginnend oben auf der grünen Taste mit 1 und dann im Uhrzeigersinn laufend 2 auf der blauen Taste rechts, 3 auf der roten Taste unten und 4 auf der gelben Taste links.



Abbildung 8: Dreamcast Controller / Trio Linker Plus

6.2.8 Trio Linker Plus mit Gamecube Controller

Mangels Gamecube-Controller kann ich diesen nicht testen, gehe aber davon aus, dass die Mappings gut vergleichbar mit dem Playstation 2 Dual Analogue sind.

6.2.9 Default Joystick (sofern es sowas gibt)

Da der Phantom Hawk relativ gut mit dem Default-Template aus der Github-Library von felis funktionierte, dient dieser als Template für einen „Standard-Joystick“ für die Achsen (X1=X, Y1=Y, X2=Twist, Y2=Throttle, Button 1-4=Button 1-4), jedoch werden keine Extra-Buttons auf Button 1-4 erweitert und es findet keine Kalibrierung statt.

7 Materialliste

- Arduino Mega 2560, damit das Gehäuse passt, mit großer USB Typ B Buchse – Achtung, es kursieren verschiedene Mega 2560 Varianten. Die meisten folgen dem offiziellen arduino.cc Layout. Bei der „AZ Delivery“ Variante auf Amazon Deutschland sind sowohl die USB-Buchse wie auch die Power-Buchse leicht versetzt, sodass das linke Seitenteil nicht passt. Die Power-Buchse kann zudem selbst bei einem Redesign des linken Seitenteils mit größeren USB-Steckern für das USB Host Shield kollidieren. Achtet darauf, dass der Arduino dem Standard-Layout folgt, z.B. Arduino Official oder Elegoo.
- USB Host Shield 2.0 – Derzeit sind mir keine unterschiedlichen Versionen des Uno/Mega USB Host Shield 2.0 bekannt. Jedoch kann je nach Anbieter ein kurzer oder ein langer Reset-Schalter verbaut sein. Der Pin wurde für den kurzen Schalter entworfen und kann notfalls bei langen Reset-Schaltern mit einem Seitenschneider gekürzt werden.
- Three-Headed Monkey Shield PCB
 - PCB selbst:
 - Gerber-Dateiset auf Github runterladen: [RetroARDUInput/PCB/Gerber at main · RetroFuturisticEngineer/RetroARDUInput \(github.com\)](#)
 - Bestellen bei <https://www.pcbway.com/> oder <https://jlcpcb.com/> (Mindestbestellmenge 5 Stück)
 - 3x Sockel DIL-16 Rastermaß 2,54, sofern die ICs gesockelt werden sollen
 - DE: 3x [MPE 001-1-016-3: Präzisionsfassung, 16 polig, RM 2,54 bei reichelt elektronik](#)
 - International: 3x [115-47-316-41-003000 Mill-Max | Mouser Europe](#)
 - 1x Maxim MAX232 RS232-Transceiver, DIL-16
 - DE: 1x [MAX 232 DIP: RS232, 2 Treiber - 2 Empfänger, DIL-16 bei reichelt elektronik](#)
 - International: 1x [MAX232CPE+ Maxim Integrated | Mouser Europe](#)
 - 2x Maxim/Dallas Semiconductor DS1803-100 (DS18030-100+) 100kOhm Dual Digital-Potentiometer, 8 Bit Auflösung je Poti, DIL-16
 - DE: 2x [Digitales Potenziometer DS18030-100+, 10kΩ 256-Position Linear 2-Kanal DIP 16-Pin | RS Components \(rs-online.com\)](#) (nur mit Gewerbeschein – Vorsicht, Reichelt hat nur den DS18030-010, der nur 10kOhm hat und daher zu falschen Werten am Joystick-Port führt)
 - International: 2x [DS18030-100+ Maxim Integrated | Mouser Europe](#)
 - 5x Elektrolytkondensator Radial 10µF/50V
 - DE: 5x [FR-A 10U 50: Elko radial, 10 µF, 50 V, 105°, Low ESR, 5 x 11 mm, RM 2 bei reichelt elektronik](#)
 - International: 5x [EEU-FR1H100 Panasonic | Mouser Europe](#)
 - 3x D-SUB 9 Female Buchse für Print-Montage, kurze Bauform Gehäusetiefe 14,3mm
 - DE: 3x [W+P 107-09-2-1-2: D-SUB Buchse, 9-polig, gewinkelt bei reichelt elektronik](#)
 - International: [L77SDE09S1ACH4F Amphenol Commercial Products | Mouser Europe](#)
 - 1x D-SUB 15 Male Buchse Print-Montage, kurze Bauform Gehäusetiefe 12,3mm
 - DE: 1x [D-SUB ST 15US: D-SUB-Stecker, 15-polig, gewinkelt, RM 7,2 bei reichelt elektronik](#)

- International: 1x [ID15P33E4GV00LF Amphenol FCI | Mouser Europe](#)
- 1x DIN-8 Tastaturstecker
 - 270° Buchse für Print-Montage (Kycon KCDX-8S-N oder pinkompatibel)
 - DE: 1x [DIN-Buchse, 8-polig \(270°\), Printausführung - Restore-Store](#)
 - International: 1x [KCDX-8S-N Kycon | Mouser Europe](#), nicht lieferbar, Langer Vorlauf, Mindestbestellmenge 10000 Stück
 - oder DIN-5 180° Buchse mit gleichem Grund-Pinout bei Verzicht darauf, das Tandy 1000 Keyboard Reset Signal (derzeit ungenutzt) künftig nutzen zu können
 - DE: 1x [MABPM 5S: DIN-Buchse, 5-polig, halbrund 180°, Print, Metallpl. bei reichelt elektronik](#)
 - International: 1x [57PC5F Switchcraft | Mouser Europe](#)
- 1x Mini-DIN 6-Pin für Print-Montage
 - DE: 1x [EB-DIO M06: Mini-DIN-Printbuchse, 6-polig bei reichelt elektronik](#)
 - International: 1x [5749266-1 TE Connectivity | Mouser Europe](#)
- 1x Mini-DIN 4-Pin für Print-Montage
 - DE: 1x [EB-DIO M04: Mini-DIN-Printbuchse, 4-polig bei reichelt elektronik](#)
 - International: 1x [5749264-1 TE Connectivity / AMP | Mouser Europe](#)
- 15x Schiebeschalter 1-1 (Umschalter), Rastermaß 4,7mm
 - DE: 15x [SS 13ASP: Schiebeschalter 1x UM, stehend, Print, RM 4,7 bei reichelt elektronik](#)
 - International: 15x [CS12ANW03 NKK Switches | Mouser Europe](#)
- 2x Stiftleiste 1x6-Pin, Rastermaß 2,54mm, kurz
 - DE: 2x [MPE 087-1-006: Stiftleisten 2,54 mm, 1X06, gerade bei reichelt elektronik](#)
 - International: 2x [G800W302018EU Amphenol Commercial Products | Mouser Europe](#)
- 2x Stiftleiste 1x8-Pin, Rastermaß 2,54mm, kurz
 - DE: 2x [MPE 087-1-008: Stiftleisten 2,54 mm, 1X08, gerade bei reichelt elektronik](#)
 - International: 2x [10129378-908003BLF Amphenol FCI | Mouser Europe](#)
- 2x Stiftleiste 1x8-Pin, Rastermaß 2,54mm, lang oder Stiftleisten plus Buchsenleisten 8,5mm Bauhöhe plus Buchsenleisten flach (siehe Beispiel)
 - DE: 2x [MPE 087-1-008: Stiftleisten 2,54 mm, 1X08, gerade bei reichelt elektronik](#)
 + 2x [MPE 094-1-008: Buchsenleisten 2,54 mm, 1X08, gerade bei reichelt elektronik](#)
 + 1x [BKL 10120990: Präzisionsbuchsenleiste 2,54mm 1x36 bei reichelt elektronik](#) (kürzen auf 2x8)
 (Alternativ: Stapelleisten wie diese auf Länge trimmen [STAPELLEISTE 50G: Stapelleiste, 50-polig, einreihig, Höhe 38mm bei reichelt elektronik](#))
 - International: 2x [10129378-908003BLF Amphenol FCI | Mouser Europe](#)
 + 2x [8Fx1-254mm Gravitech | Mouser Europe](#)
 + 1x [310-87-136-41-001101 Preci-dip | Mouser Europe](#) (kürzen auf 2x8)
 (Alternativ: Stapelleisten wie diese auf Länge trimmen [8-146455-6 TE Connectivity | Mouser Europe](#))
- 1x Stiftleiste 2x18-Pin, lang oder oder Stiftleisten plus Buchsenleisten 7mm Bauhöhe plus Buchsenleisten flach (siehe Beispiel)

- DE: 1x [MPE 087-2-040: Stiftleisten 2,54 mm, 2X20, gerade bei reichelt elektronik](#) (auf 2x18 kürzen)
 +1x [BL 2X18G7 2,54: Buchsenleiste - 18-pol, gerade, RM 2,54, H: 7,0 mm bei reichelt elektronik](#)
 +1x [BKL 10120990: Präzisionsbuchsenleiste 2,54mm 1x36 bei reichelt elektronik](#) (auf je 2x 18 Pin mittig trennen)
 (Alternativ: Stapelleisten wie diese auf Länge trimmen [STAPELLEISTE 50G: Stapelleiste, 50-polig, einreihig, Höhe 38mm bei reichelt elektronik](#))
- International: 1x [M20-9981845 Harwin | Mouser Europe](#)
 + [1-534206-8 TE Connectivity | Mouser Europe](#) (8mm)
 +1x [310-87-136-41-001101 Preci-dip | Mouser Europe](#) (auf 2x18 mittig trennen)
 (Alternativ: Stapelleisten wie diese auf Länge trimmen [8-146455-6 TE Connectivity | Mouser Europe](#))
- 3x Widerstand 10kOhm min. 1/4W (für Pulldown R1 bis R3)
 - DE: [VI MBA02040C1002: Dünnschichtwiderstand, axial, 0,4 W, 10 kOhm, 1% bei reichelt elektronik](#)
 - International: [MBA02040C1002FC100 Vishay / Beyschlag | Mouser Europe](#)
- Optional: 6x LED 5mm, Farbe nach Belieben, und 1x Vorwiderstand (um die 10 Ohm Minimum), nach Helligkeitspräferenz, für die Beleuchtung des Gehäuses, falls mit transparenten Materialien gearbeitet wird
 - DE:
 - Vorwiderstand: [VIS PR0100010102: Widerstand, Metallschicht, 10 Ohm, axial, 1 W, 5% bei reichelt elektronik](#)
 - LED 8000mcd 30° 4V 30mA Grün: [LED 5-08000 GN: LED, 5 mm, bedrahtet, grün, 8000 mcd, 30° bei reichelt elektronik](#)
 - International:
 - Vorwiderstand: [CPF110R000GKRE6 Vishay / Dale | Mouser Europe](#)
 - LED [WP7113LGD Kingbright | Mouser Europe](#)
- Gehäuse (Schrauben sind zu kaufen, Rest ist 3D-Druck und auszudrucken)
 - 21x Schrauben 2,9mm x 6,5mm selbstschneidend PH1 Linsenkopf, davon
 - 4x Fixierung Arduino
 - 4x Fixierung 3-Headed Monkey
 - 4x Fixierung Deckel
 - 6x (3x je 2x) Fixierung Platten Front und Links/Rechts
 - 3x Fixierung hintere Platte
 - DE: [SBL 29065-100: Blechschrauben, PAN Head, PZD, 2,9 x 6,5 mm, 100 Stück bei reichelt elektronik](#) oder im lokalen Baumarkt
 - International: Im lokalen Baumarkt
 - Basisplatte
 - Gehäusedeckel
 - Gehäuse-Seitenwand links (Arduino Ports)
 - Gehäuse-Seitenwand hinten (1x D-SUB 15, 3x D-SUB 9)
 - Gehäuse-Seitenwand rechts (1x DIN, 2x Mini-DIN, versteckter Debug-Schalter)
 - Gehäuse-Seitenwand vorne (Schriftzug)
 - 1x Führung für Reset-Schalter
 - 1x Stift Reset-Schalter
 - 1x Schalterkappe 5 Schalter übergreifend (Tandy/Andere Tastaturumschalter)
 - 2x Schalterkappe 2 Schalter übergreifend (Joystick-Potentiometer Open/Ground)

- 5x Schalterkappe 1 Schalter übergreifend (diverse Modus-Schalter)

Gehäuse im Auftrag drucken lassen: STLs auf Github runterladen ([RetroARDUInput/Case/STL at main · RetroFuturisticEngineer/RetroARDUInput \(github.com\)](https://github.com/RetroARDUInput/Case/STL)) und bestellen bei <https://www.pcbway.com/>, <https://jlcpcb.com/> (beide SLA-Druck) oder einem deutschen Anbieter (einfach suchen – haben evtl. auch FDM-Druck) - Limitierungen wie „nur ein Teil pro STL“ (nicht das STL mit allen Schaltern nehmen, sondern 5x den 1-fach Schalter, 2x den 2-fach Schalter und 1x den 5-fach Schalter bestellen), Warnungen wegen zu dünner Wände (bitte „ich akzeptiere das Risiko“ auf den Prüfbericht zurückschreiben) und separater Versand von den PCBs (leider zwei Bestellungen nötig) können bei PCBWay und JLCPCB auftreten.

8 Zusammenbau

Das Kapitel behandelt alle Themen von Einzelteilen und Dateien bis zum fertigen Gerät. Das USB Host Shield bedarf etwas Vorbereitung. Das Three-Headed Monkey Shield wird beidseitig bestückt. Einige Teile kommen auf die Oberseite, u.A. natürlich die Schalter, die von oben zugänglich sein müssen, aber andere Teile wurden zwecks Reduzierung der Gehäusehöhe nach unten verlagert, wie die Anschlüsse für die Retro-Computer, aber auch die Kondensatoren für den MAX232, die ansonsten am Gehäusedeckel anstoßen könnten, oder die Verbinder zum Arduino, der unter dem Shield sitzt. Alle Positionierungen sollten auf dem Silk Screen markiert sein, man kann eigentlich kaum etwas falsch machen. Wer das Gehäuse mit seinem 3D-Drucker (hier werden konkret FDM-Drucker behandelt) selbst drucken möchte, bekommt noch einige Tipps für Einstellungen, aber auch Filamentwechsel für Mehrfarbdruck, und dann wird das Gehäuse zusammengebaut. Zum Schluss wird der Arduino programmiert.

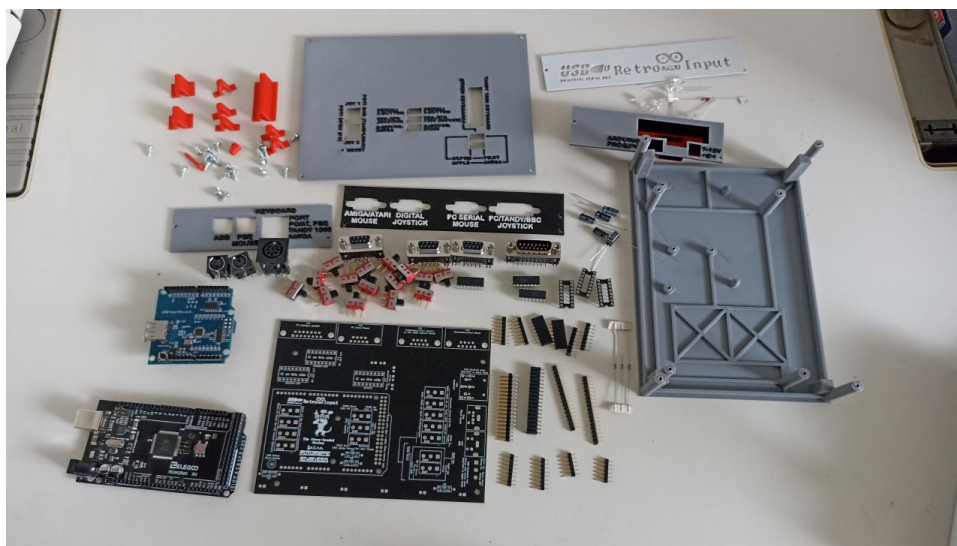


Abbildung 9: Das Projekt in Einzelteilen

8.1 Vorbereitung des USB Host Shields (Lötbrücken setzen)

Das USB Host Shield ist universell für den Betrieb auf 3,3V und 5V vorbereitet und muss erst durch Setzen von Lötbrücken auf den jeweiligen Modus konfiguriert werden. Sind diese Lötbrücken nicht gesetzt, stürzt der Arduino sofort ab und lässt sich nicht mal mehr über das USB-Kabel programmieren.

Wir arbeiten im 5V-Modus und müssen deshalb die Lötbrücken wie in der Abbildung unten setzen:

- Am Header 3.3V und 5V
- Am USB-Port nur 5V

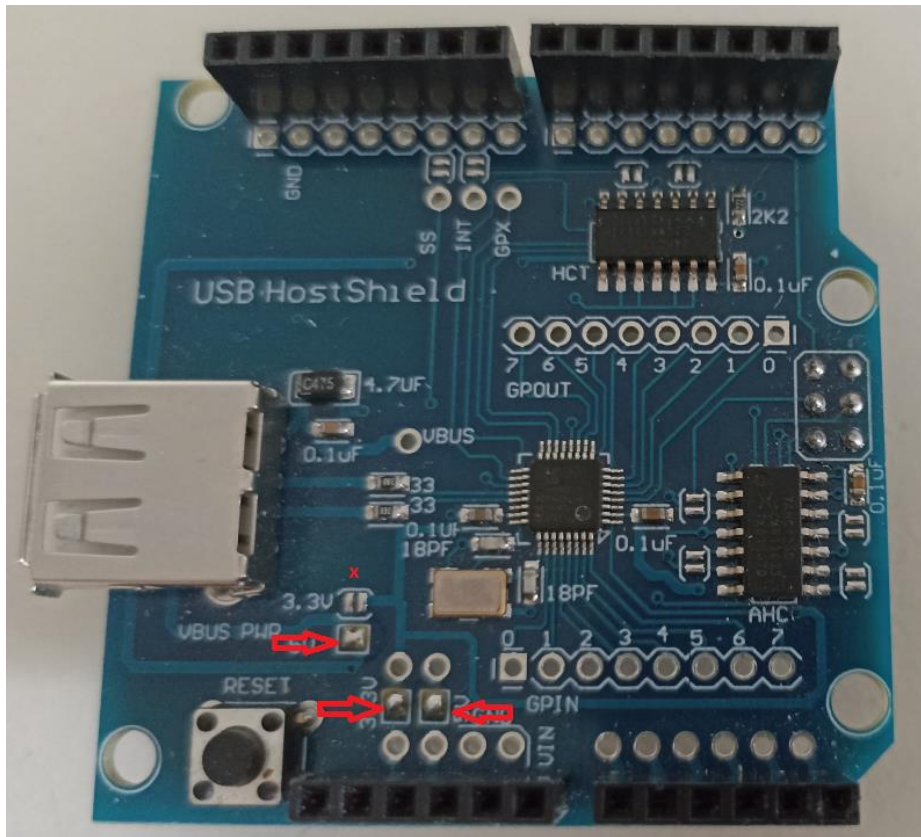


Abbildung 10: Lötbrücken zum Jumpen auf 5V-Modus

8.2 Bauteile auf der Oberseite des Three-Headed Monkey Shields

Beim Zusammenbau fangen wir auf der Oberseite an. Diese enthält nur die IC-Sockel/ICs und die Schalter, abgesehen vom Debug-Message Schalter SW11, die Pulldown-Widerstände R1 bis R3 sowie optional der Vorwiderstand R4 für die LEDs. Ich empfehle, die ICs zu sockeln.

Reihenfolge (kleinstes Bauteil zuerst):

- Widerstände einlöten.
- IC-Sockel einlöten und auf Lötbrücken prüfen.
- Schalter (außer SW11) einlöten (die Pins sind so weit auseinander, dass keine Brücken entstehen sollten). Die Schalter haben einen dünnen und einen dicken Teil der Pins. Der dicke Teil geht nicht durch die Löt-Pads und das ist auch nicht vorgesehen. Die Schalter sitzen auf der Verdickung gerade auf, können aber etwas wackeln. Daher erst mal mit einem Pin anlöten, das Lötzinn noch einmal erhitzen und den Schalter von oben festdrücken, sodass er gerade sitzt, damit nachher die Schalterkappen sauber sitzen).
- Ob man jetzt schon die ICs in die Sockel einsetzt oder dies erst am Ende, nach Prüfung auf Lötbrücken für die Komponenten auf der Unterseite, macht, ist Geschmackssache.

8.3 Bauteile auf der Unterseite des Three-Headed Monkey Shields

Auf der Unterseite sitzen der Schalter SW11, die Kondensatoren für den MAX232, die Anschlüsse für die emulierten Eingabegeräte und die Pins zur Verbindung des Arduinos, sowie optional die LEDs.

Reihenfolge:

USB-RetroARDUInput
Open Source USB to Retro Computer Adapter

- LEDs einlöten, falls mit Beleuchtung gearbeitet wird.
- Anschlüsse, in der Reihenfolge „klein nach groß“: ADB und PS/2, dann alle D-SUB, dann der DIN-Tastaturanschluss.
- Kondensatoren (Markierung auf PCB Silk Screen und Kondensator beachten, gepolte Elkos). Damit diese halten, ggf. nach dem Einsetzen die Pins leicht aufbiegen.
- Schalter SW11 einlöten.
- Kontaktierung zum Arduino. Empfehlung: In den Stack aus Arduino und USB Host Shield die Pinreihen einsetzen, die kurzen Reihen in das USB Shield und die langen Reihen (bzw. Stapel zur Verlängerung) in den Arduino, das Host Shield aufsetzen und verlöten. So werden die Pinreihen schön gerade.
 - Tipp zu dem Steckleistenstapel für die langen Pins: Die Stiftleisten (Pins oben und unten) kommen mit der langen Seite in den Arduino. Auf diese kommen die hohen Buchsenleisten. Diese werden dann in die niedrigen Buchsenleisten gesteckt, welche durch die Löcher der 3-Headed Monkey Platine kommen. Die hohe Buchsenleiste für den 36-Pin Anschluss ist etwas niedriger, als die 8-Pin Varianten, allerdings passt es trotzdem problemlos, da die Summe der Pins lang genug ist.

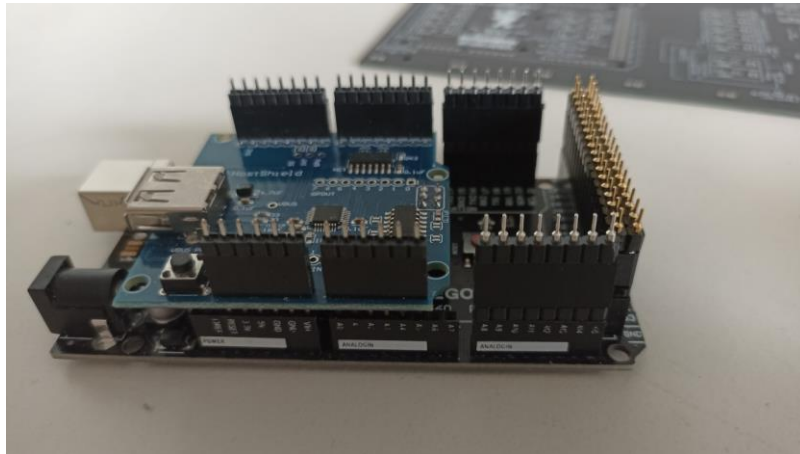


Abbildung 11: Einsetzen und anlöten der Pinreihen

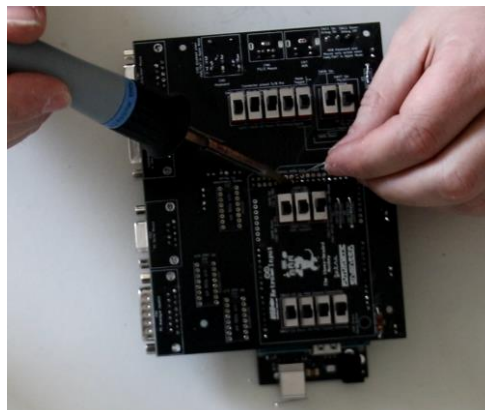


Abbildung 12: Bestückte Platine beim Einlöten der Verbinder zum Arduino

8.4 Slicing und Druck des 3D-Druck Gehäuses

Wer sich dieses Gehäuse selbst drucken möchte, hier ein paar Tipps zum Slicing:

Zu Temperatur und Geschwindigkeit müsst ihr euch selbst die besten Einstellungen suchen. Generell kann ich euch aber den „Monotone Top Layer“ von Cura ans Herz legen (Ausnahme: Linke Seite, führt zu ganz kruden Bewegungsmustern bei der USB-Host Mulde und ist für die Stirnfläche hier nicht nötig), da es ansonsten zu vermehrten Neuansätzen kommen kann, die zu Streifenbildung führen. Zwar setzt auch der Monotone Top Layer neu an, aber, auf Kosten von minimal mehr Druckzeit, deutlich intelligenter mit weniger bis keiner Streifenbildung. Ironing führte bei mir zu einer Verschlechterung des Ergebnisses, wenn ihr dort bessere Erfahrungen gemacht habt, könnt ihr es gerne nutzen.

Generell drucke ich alle Teile mit 100% Infill, da ein reduzierter Infill konstruktionsbedingt kaum zum Einsatz kommt und nur minimalste Zeitersparnis bringt und aufgrund der Verschraubung mit selbstschneidenden Schrauben durch mehr Material auch mehr Stabilität bedeutet.

Mehrfarbigen Druck muss der Drucker beherrschend. Probiert das erst mal mit einem kleinen Teil und macht euch über eventuelle Firmware-Updates schlau, falls euer Drucker das mit der Stock-Firmware nicht beherrscht. So ignoriert z.B. der Ender 3 V2 mit seiner Marlin-basierten Stock-Firmware den M600-Befehl für den Filamentwechsel und druckt weiter. Zudem empfehle ich, dass euer Drucker (ab Werk oder nachgerüstet) über eine beidseitige Z-Führung, einen Bed-Level-Sensor und ein hochwertiges Druckbett (z.B. PEI) verfügt, da der Filamentwechsel mehr Präzision erfordert. Ich empfehle an dieser Stelle den kostenlosen alternativen Marlin-Fork Jyers UI, den es als Quellcode und fertig compiliertes Paket (manuelles Leveln oder BL-Touch/CR-Touch) gibt und der auch ein paar weitere sinnvolle Zusätze mitbringt, sich ansonsten aber wie die Stock-Firmware anfühlt. Bei einem Update guckt nach, welche Board-Version ihr habt (beim Ender 3 V2 ein Silent-Board Version 4.2.2 oder 4.2.7), eine Firmware für die falsche Board-Version führt zu Fehlfunktionen, lässt sich aber durch erneutes Aufführen problemlos beheben.

Im Punkt Post Processing können die Filamentwechsel und weitere Änderungen (ChangeAtZ) eingefügt werden. Bei mehrfarbigen Drucken (Layerhöhe 0.2, keine Rafts oder andere Anhaftungshilfen) handhabe ich das wie folgt:

- Gehäuse und Schalter: Keine Filamentwechsel, alles einfarbig, Gehäuse in Mittelgrau, Schalter in Rot
- Top-Cover (Ausrichtung bleibt, Stützstrukturen notwendig):
 - o Start mit Mittelgrau
 - o Filamentwechsel auf sowie (müsst ihr ausprobieren) ggf. Geschwindigkeitsreduzierung (ChangeAtZ, Geschwindigkeit in %) für die Beschriftung bei Layer 23
- Rechtes Cover (um 90° drehen, sodass es mit der Schrift nach oben liegt):
 - o Start mit Mittelgrau
 - o Filamentwechsel auf Schwarz sowie ggf. Geschwindigkeitsreduzierung (ChangeAtZ, Geschwindigkeit in %) für die Beschriftung bei Layer 11
- Hinteres Cover (um 90° drehen, sodass es mit der Schrift nach oben liegt):
 - o Start mit Schwarz
 - o Filamentwechsel auf Hellgrau sowie ggf. Geschwindigkeitsreduzierung (ChangeAtZ, Geschwindigkeit in %) für die Beschriftung bei Layer 11
- Linkes Cover (um 90° drehen, sodass es mit der Schrift nach oben / USB-Mulde nach unten liegt, Stützstrukturen notwendig):

- Start mit Rot
- Filamentwechsel auf Schwarz sowie ggf. Geschwindigkeitsreduzierung (ChangeAtZ, Geschwindigkeit in %) für die Beschriftung bei Layer 6
- Geschwindigkeit, falls bei Layer 6 reduziert, kann bei Layer 11 wieder erhöht werden (ChangeAtZ)
- Filamentwechsel auf Mittelgrau für die Deckschicht bei Layer 88 – hier wurden noch 5 Layer der Deckschicht in schwarz gedruckt, da es keine gute Idee ist, direkt über Stützstrukturen einen Filamentwechsel durchzuführen
- Filamentwechsel auf Schwarz sowie ggf. Geschwindigkeitsreduzierung (ChangeAtZ, Geschwindigkeit in %) für die Beschriftung bei Layer 94
- Front Cover „Gravierte Schrift“:
 - Start mit Transparent oder Schwarz nach Belieben
 - Filamentwechsel auf Hellgrau für die Deckpflicht bei Layer 6
- Front Cover „Erhabene Schrift“:
 - Start mit Hellgrau
 - Filamentwechsel auf Schwarz für die Beschriftung bei Layer 11

Wem die Schrift beim Filamentwechsel im Mehrfarbdruck Probleme macht, der kann unter Umständen einen Purge Tower drucken. Cura unterstützt standardmäßig allerdings nur dann Purge Tower, wenn ein Dual Extruder verbaut ist. Mein Workaround dafür ist, einen schmalwandigen Hohlzylinder (Tube) zu bauen, z.B. 5mm Außenradius, 4,5mm Innenradius, dessen Höhe so gewählt ist, dass der letzte gedruckte Layer der ist, an dem man die Farbe wechselt. Ein Seitenteil (außer links) ist beispielsweise 2mm stark, hat also bei 0,2mm Layerstärke 10 Layer. Beim 11. Layer wird das Filament gewechselt und die Schrift gedruckt. Der Purge Tower muss also 2,2mm hoch sein und so positioniert werden, dass er vor dem Hauptteil gedruckt wird (Preview für den entsprechenden Layer und dann den Druckkopf-Pfad auf Anfang drehen). Dann wird zuerst auf dem Purge Tower gedruckt, mit allen Problemen, die man nach einem Filamentwechsel so erwarten kann (zu viel oder zu wenig Filament), der Druckkopf ist danach korrekt gefüllt, es erfolgt ein Retract und der Druckkopf wechselt kontrolliert zum Schriftbereich und schreibt dort sauber.

8.5 Zusammenbau der 3D-Druck-Teile für das Gehäuse

8.5.1 Tipp vorab – Zerlegen und Neueindrehen von Schrauben in Kunststoffgehäusen

Solltet ihr einmal Schrauben aus dem Gehäuse entfernen, so gilt ein Tipp, der sich auch auf andere Kunststoffgehäuse, z.B. Spritzgussgehäuse aus ABS wie sie unsere heißgeliebten Retro-Computer haben, anwenden lässt:

Selbstschneidende Schrauben schneiden sich beim ersten Eindrehen ein Gewinde. Dreht man die Schraube aus und wieder ein und erwischt dabei nicht das ursprünglich geschnittene Gewinde, schneidet sich ein neues Gewinde durch das alte Gewinde durch und die Schraubenaufnahme nutzt sich ab und bietet bei zu häufiger Wiederholung keinerlei Halt mehr.

Daher gilt, bei jedem Kunststoffgehäuse mit selbstschneidenden Schrauben, bei dem etwas empfindlicheren 3D-Druck aber noch mehr: Beim ersten Eindrehen in ein jungfräuliches Gehäuse einfach rein mit der Schraube. Beim Aus- und Eindrehen aber folgende Technik anwenden: Die Schraube ansetzen und erst mal „auf“ (gegen den Uhrzeigersinn) drehen, bis man einen leichten Ruck spürt und hört. Dann ist die Schraube in ihr bereits geschnittenes Gewinde gefallen und kann dann eingedreht (im Uhrzeigersinn) werden.

8.5.2 Jetzt aber – Gehäuse zusammenbauen

Beim Zusammenbau wird wie folgt vorgegangen:

Schritt 1: Den Arduino in das Gehäuse einsetzen. Zwei der Abstandshalter haben Pins. Diese helfen, den Arduino auszurichten, sind aber auch der Tatsache geschuldet, dass Schraubenköpfe hier keinen Platz hätten. Den Arduino mit insgesamt 4 Schrauben festdrehen

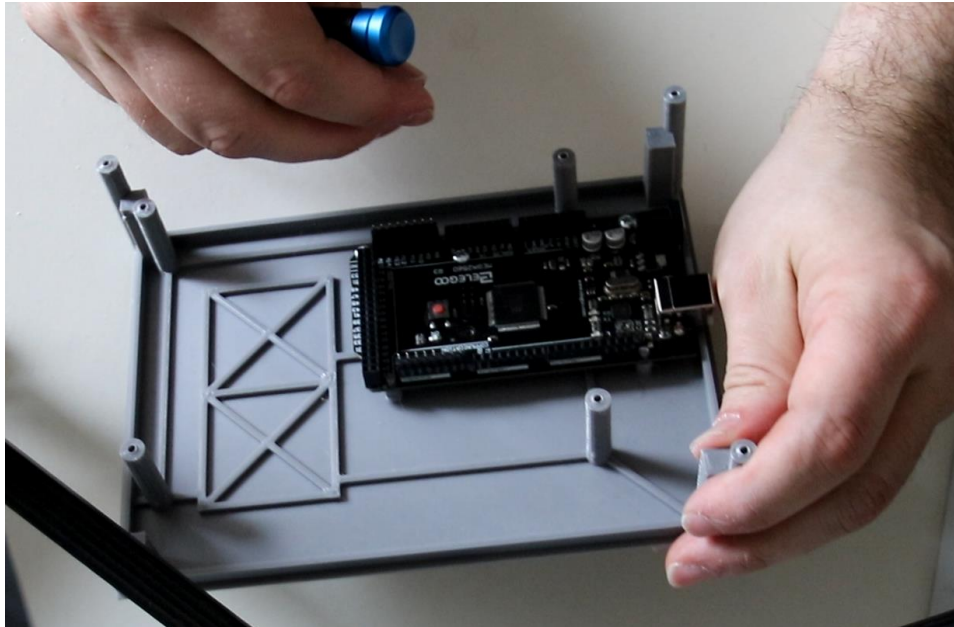


Abbildung 13: Verschraubung Arduino

Schritt 2: USB Host Shield aufsetzen

Schritt 3: 3-Headed Monkey Shield aufsetzen und mit insgesamt 4 Schrauben verschrauben. Vorne links, vorne rechts und hinten links die Schrauben wieder etwas lösen und den 3-Headed Monkey leicht anheben. Die Führungshülse für den Reset Pin auf den Reset-Schalter des USB Host Shields aufsetzen (das Loch passt genau um den Taster herum) und die Hülse zwischen Reset-Schalter und 3-Headed Monkey einklemmen. Den 3-Headed Monkey wieder festdrücken und fest verschrauben.

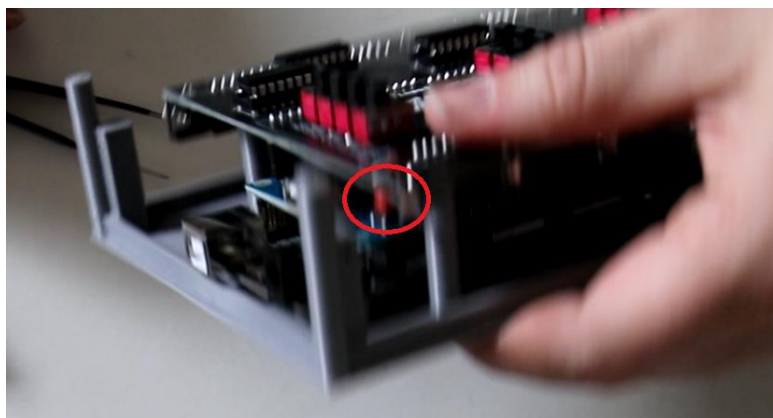


Abbildung 14: Befestigung 3-Headed Monkey und Reset-Führungshülse

Schritt 4: Schalterkappen auf die Schalter aufsetzen. Die 5-Fach-Kappe deckt die 5 Schalter (Modus und Pinout-Änderung) für den Tandy Tastaturmodus ab. Die beiden 2-Fach-Kappen decken die jeweils 2 Schalter ab, die die Analog-Joystick-Potis offen oder gegen Masse gezogen halten (PCs offen, BBC Micro, Tandy und Tano/Dragon auf Masse). Die 1-Fach-Kappen kommen auf die restlichen Schalter. Den Reset-Pin durch das Loch im 3-Headed Monkey Shield stecken, sodass er auf dem Reset-Schalter des USB-Shields aufliegt. Dann vorsichtig den Deckel aufsetzen und die Schalter durch die jeweiligen Öffnungen führen, den Reset-Pin vorsichtig in die Führungshülse im Deckel einführen. Sobald der Deckel inklusive aller Schalter sauber auf dem Gehäuse aufliegt, den Deckel mit 4 Schrauben verschrauben.

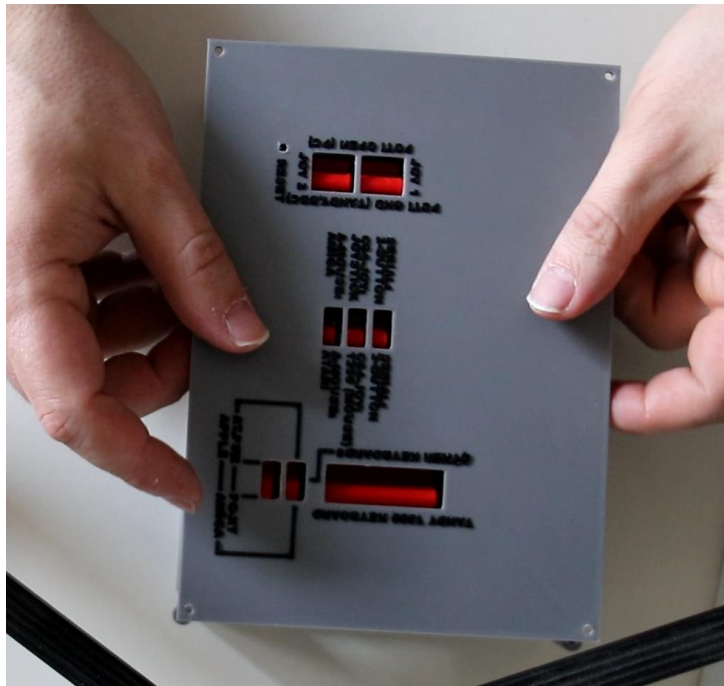


Abbildung 15: Deckel aufsetzen

Schritt 5: Die Seitenteile mit jeweils 2 (Rückseite 3) Schrauben verschrauben



Abbildung 16: Fertig ist das Gehäuse

9 Programmieren des Arduinos

Der Arduino benötigt nun natürlich noch seine Firmware. Ladet hierzu das Paket mit dem .ino Sketch sowie einigen .cpp und .h Dateien für den RetroARDUInput von meiner Github-Seite runter:

[RetroARDUInput/RetroArduInput_FW at main · RetroFuturisticEngineer/RetroARDUInput \(github.com\)](https://github.com/RetroARDUInput/RetroArduInput_FW_at_main)

Zudem wird die USB Host Shield Library 2.0 von Circuits@Home benötigt, die ihr im Bibliotheken-Verwaltung (Strg+Shift+I) herunterladen könnt. Die Version ist erst mal zweitrangig, ich empfehle aber die aktuelle 1.6.1. Auf dem Github von felis (Link in den Quellen) gibt neben der Library (ich empfehle aber die Bibliotheken-Verwaltung, da diese simpel die aktuellste Library global installieren kann, man kann sie aber auch dort runterladen und manuell einbinden) auch einige Beispiele, auf denen ich dann auch aufgesetzt habe.

Eine Neuprogrammierung ist derzeit aufgrund fehlender Joystick-Erkennung auch für den Wechsel auf einen anderen Joystick nötig (Variable JoyType in globals.cpp mit einem der per #define definierten Werte aus der globas.h anpassen), daher sollte sich jeder einmal mit der Arduino

Entwicklungsumgebung beschäftigen. Aber wer einen DOS-PC am Leben halten kann, für den sollte das wohl auch die geringste Hürde darstellen.

Das Verzeichnis für den Sketch muss den gleichen Namen tragen, wie die .ino Datei, nämlich RetroArduInput_FW. Dann kann der Sketch geladen werden und sollte, mit mehreren weiteren automatisch geöffneten Tabs für die .h und .cpp Includes in der IDE angezeigt werden. Wurde die Host Shield Library korrekt als globale Library eingebunden, braucht man eigentlich nur noch den Programmier-Button zu klicken und schon wird der Sketch compiliert und hochgeladen.

10 Informationen zu den Interfaces

10.1 CN1: Analog-Joystick

Das PC-Joystick-Interface wurde ursprünglich ausgelegt für 2 Joysticks mit je 2 Achsen und 2 Buttons, die sich über ein Y-Kabel anschließen ließen. Da die USB-Libraries für das Arduino USB Host Shields keine zwei gleichzeitigen Joysticks unterstützen, wurde der Ansatz modernerer Joysticks gewählt, die die Pins für Joystick 2 für zusätzliche Achsen (z.B. Throttle beim CH Products Flight Stick) und Buttons nutzten. Da die meisten USB-Joysticks ebenfalls über 3 oder 4 Achsen (meist X, Y, Rotation und Throttle) und 4 oder mehr Knöpfe beherrschen, lassen sich so 4-Achsen/4-Tasten-Joysticks bereitstellen.

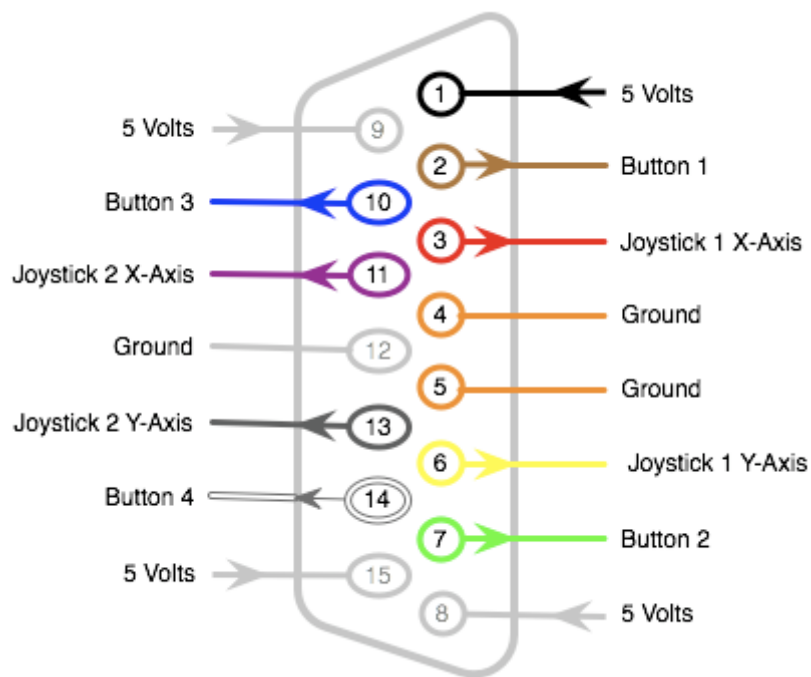


Abbildung 17: Pinout Analoger Joystick-Port

Der 15-Polige Analog-Joystick-Port wird nativ für PC-Joysticks verwendet. Das 15-polige Joystick-Interface stellt hierfür eine Referenzspannung bereit (5V bei ISA-Karten, 3,3V bei PCI-Karten). Diese werden pro Achse bereitgestellt und durch je ein 100kOhm Poti geschickt. PC-Soundkarten haben an den Eingängen je einen 10kOhm Widerstand gegen Masse. Über diesen wird der Spannungsabfall gemessen. Das Poti, in einem echten Joystick in lineares Poti oder mechanisch übersetztes Dreh-Poti, im Falle des RetroARDUInputs ein Digital-Poti, liefert zwischen Eingang und Schleifer einen Widerstand von 0-100kOhm, was zusammen mit dem 10kOhm Poti auf der Soundkarte gemäß Spannungsteilerregel die jeweiligen Messspannungen im Bereich von 5V (bzw. 3,3V) bei $R_1=0\Omega$ und 0,45V (bzw. 0,3V) bei $R_1=100\Omega$ ergibt.

$$U = U_{ref} * \frac{R2}{R1 + R2}$$

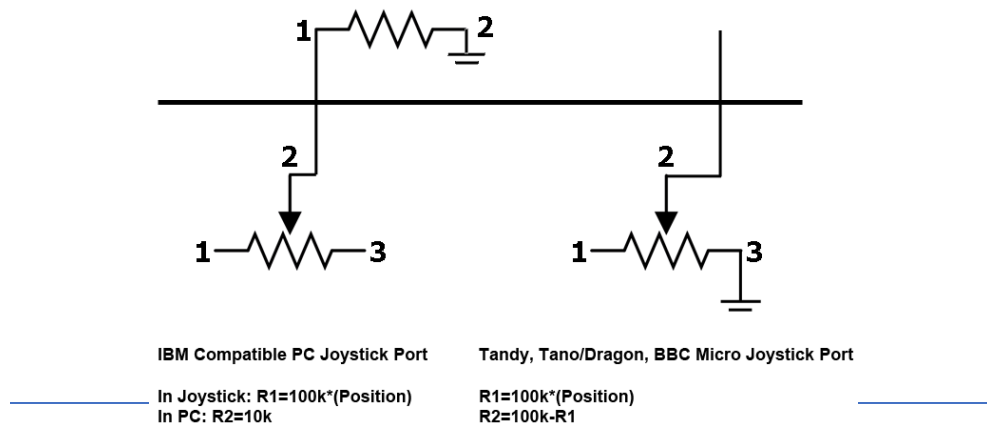


Abbildung 18: Joystick-Spannungsteiler in verschiedenen Computer-Typen

Da einige Computer mit analogen Joystick-Ports diesen Messwiderstand nicht haben, muss hierfür das Poti gegen Masse gezogen werden. Das erledigen die Schalter SW12 bis SW15. Betroffene Computer sind (keine Anspruch auf Richtigkeit) der BBC Micro (ebenfalls ein 15-Pin D-SUB, jedoch mit einem anderen Pinout), der Tandy TRS-80 (und somit wohl auch der Tandy 1000, der Ports für TRS-80 Joysticks besitzt) und der Tano/Dragon. Bei diesen Geräten gilt, dass R1 der Weg zwischen U_{ref} und Schleifer (0-100kOhm) und R2 der Weg zwischen Schleifer und Masse ($R2 - R1 = 100-0kOhm$) ist. Durch Umschalten der Poti-Ausgänge auf Masse lässt sich der Joystick-Port des RetroARDUInput mit Adapterkabeln auch für diese Computer nutzen.

Die Buttons werden durch Ziehen der Button-Pins gegen Masse geschaltet (Button gedrückt=Verbindung zu Masse)

10.2 CN2: PC Serial Mouse

Die serielle Maus nutzt die RS232-Schnittstelle, über die die Daten seriell als verpackte Datenpakete übermittelt werden. Da diese Schnittstelle einen Spannungsbereich von +12 bis -12V nutzt, ist ein MAX232 als Umsetzer zur 5V-TTL-Spannung des Arduino nötig.

Wir nutzen hier den Serial2 UART des Mega2560, der auf die von Microsoft (2-Button) und Logitech (3-Button) genutzte Bitrate und Handshake 1200-7N1 gestellt wird.

Zudem wird der RTS-Pin über eine Receive-Leitung des MAX232 auf einen I/O Pin gegeben. In diesem Fall nehmen wir einen mit Interrupt. Der PC versorgt normalerweise über CTS und RTS die Maus mit Strom, lässt RTS jedoch für einen kurzen Moment fallen, wenn der Maustreiber den Typ der verbundenen Maus abfragen möchte. Zurückgegeben wird „M“ für Maus (2-Button Microsoft) oder „M3“ für den Logitech 3-Button Modus.

Übermittelt werden:

Bit	6	5	4	3	2	1	0
Paket							
1	1	L-Button	R-Button	Y7	Y6	X7	X6
2	0	X5	X4	X3	X2	X1	X0

3	0	Y5	Y4	Y3	Y2	Y1	Y0
(4)	0	M-Button	0	0	0	0	0

Wobei das vierte Datenpaket nur im Logitech 3-Button Protokoll verwendet wird, und selbst da nicht immer.

Es handelt sich um ein Signed Byte, bei dem das achte Bit das Vorzeichen darstellt und dementsprechend Werte von +/- 0 bis 127 erreicht werden können.

Die Logitech-Maus überträgt im normalen Zustand nur die Pakete 1 bis 3 und verhält sich damit wie eine Microsoft-Maus. Wird die mittlere Taste gedrückt, wird das vierte Paket mit dem Wert 0x20 (Bit 5 = 1) übertragen. Wird die mittlere Taste losgelassen, muss das vierte Paket mit dem Wert 0x00 einmalig übertragen werden, falls die Maus im Stillstand ist. In Bewegung reicht auch ein Set mit X/Y Daten das nur aus 3 Datenpaketen besteht, damit der Maustreiber die mittlere Taste als gelöst ansieht.

10.3 CN3: C64/Atari Joystick und 1350 Joystick Mouse

Anmerkung: Hier wurde nicht die 1351 Proportional Mouse implementiert. Diese würde tatsächlich über die Paddle-Register des SID unterschiedliche Positionen, die sich abhängig von der Bewegungsgeschwindigkeit ändern können, vorgeben. Eine Implementierung schien mir aber gemessen an Aufwand und Nutzen erst einmal nicht erstrebenswert. Vielleicht kommt diese in einer späteren Revision.

Die 1350 Joystick Mouse nutzt die Richtungspins des Digitaljoysticks und schließt diese, genauso wie der Mikroschalter im Joystick, gegen Ground, wenn die Maus in eine bestimmte Richtung bewegt wird. Sie kann daher auch nur „bewegen“ und „nicht bewegen“. Der Anschluss CN3 reagiert in diesem Fall also auf die USB-Maus und nicht auf den USB-Joystick. Dies ist mehr als nette Spielerei zu betrachten und war schnell implementiert.

Für die Joystick-Implementierung wird auf den USB-Joystick gehört. Digitale Joysticks wie das D-Pad eines USB-Gamepads oder die Schalter eines Competition Pro USB erzeugen Vollausschlag. Analoge Joysticks hingegen liefern verschiedene Werte und sind je nach Modell in einem runden Bewegungsfeld gefangen, können also in Diagonalen keinen Vollausschlag melden. Daher wird die Reaktion, ob eine Richtung getriggert wird oder nicht, an einer Dead Zone (Default +/- 40 um den Mittelpunkt von 128 bei 8 Bit Auflösung (0-255)), die in der globals.h auch verändert werden kann, vorgegeben. Innerhalb der Dead Zone, also einem Radius um die Zentrierung, gilt der Pin als offen. Verlässt der Wert die Dead Zone, wird der Pin geschlossen. Somit löst ein Analog-Joystick die Digital-Joystick-Emulation nicht schon bei der kleinsten Bewegung, aber auch nicht erst im Endanschlag aus.

Die Funktionsweise des Digital-Joysticks ist denkbar einfach. Jede Richtung und jeder Schalter wird durch einen Pin repräsentiert, der entweder offen ist (keine Aktion) oder gegen Masse geschlossen wird (Bewegung, Feuerknopf gedrückt). Das Pinout ist dabei bei allen Systemen (Atari, C64, Amiga) gleich und wie folgt:

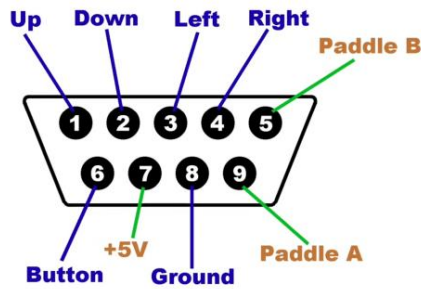


Abbildung 19: Commodore/Atari Digital Joystick Port

Anmerkung: Paddle bei C64, Pin 9 auch Button 2 möglich

10.4 CN4: Commodore/Atari Mouse

Commodore und Atari verwenden beide die Rohdaten des Quadratsensors, einem Drehgeberrad mit zwei versetzten Lichtschranken, mit denen die Maus die Richtung und Geschwindigkeit erkennen kann. Dieses Verfahren benutzen alle Kugelmäuse, jedoch werden bei Interfaces wie RS232, PS/2 und ADB die Daten ausgewertet und die Bewegungsschritte in ein Protokoll enkapsuliert.

Da Steg und Schlitz im Drehgeberrad breit genug sind, um beide Lichtschranken zu öffnen oder zu schließen, ergeben sich somit folgende Kombinationsmöglichkeiten:

- 0-0 – Beide Lichtschranken verdeckt
- 0-1 – Lichtschranke 1 verdeckt, Lichtschranke 2 offen
- 1-1 – Beide Lichtschranken offen
- 1-0 – Lichtschranke 1 offen, Lichtschranke 2 verdeckt

Der Zustandsübergang gibt die Geschwindigkeit vor, der Verlauf von Öffnung und Schließung der beiden Lichtschranken (00-01-11-10 oder 00-10-11-01) die Richtung.

Commodore (Amiga, PC 1, PC 10-III, PC-20-III, COLT) und Atari (ST-Serie) verwenden die gleichen Pins, jedoch ordnet Atari die Pins für die Quadraturimpulse etwas anders an. Diesem Umstand kann damit begegnet werden, dass der Arduino einfach andere Ausgabe-Pins für die Werte benutzt, konfiguriert durch den Modus-Schalter SW10.

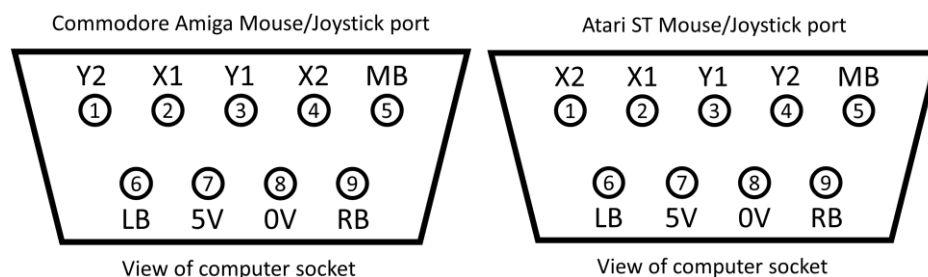


Abbildung 20: Commodore und Atari Mouse Port

Die Buttons werden durch Ziehen der Button-Pins gegen Masse geschaltet (Button gedrückt=Verbindung zu Masse).

10.5 CN5: Keyboard

Gemeinsam haben alle Computer mit DIN-Tastaturanschluss (PC-XT, PC-AT, Desktop-Amigas, Tandy 1000) die ersten 5 Pins des DIN-Steckers.

Bis auf den Tandy 1000 verwenden alle Systeme den gleichen 5-Pin 180° DIN-Stecker mit dem gleichen Pinout. Der Tandy 1000 verwendet einen 8-Pin 270° DIN-Stecker mit abweichendem Pinout. Da das 5-Pin 180° DIN-Kabel auch in die 8-Pin 270° Buchse passt, wurde hier auf eine 8-Pin 270° Buchse gesetzt, um alle Pins des Tandy abdecken zu können.

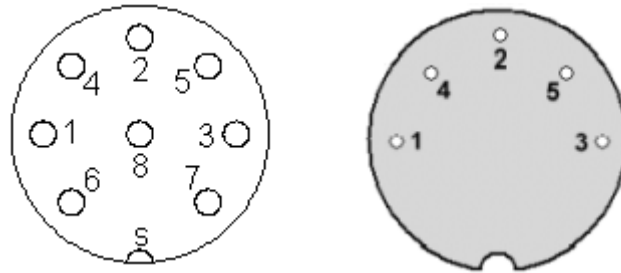


Abbildung 21: 5-Pin 180° (PC, Amiga) und 8-Pin 270° (Tandy 1000) Tastatur-Pinout Buchsenseite

Tandy1000 8-Pin	PC-XT/PC-AT/Amiga 5-Pin
1: Data	1: Clock
2: Busy	2: Data
3: Ground	3: Reset
4: Clock	4: GND
5: VCC +5V	5: VCC +5V
6: Reset	
7: NC	
8: NC	

Tatsächlich würde auch eine 5-Pin Buchse reichen, da einzig Pin 6, der Keyboard Reset des Tandy 1000, auf einem nicht in der 5-Pin Buchse belegten Pin liegt. Die Pins Keyboard Reset und Busy werden eigentlich nicht benötigt und wurden in der Entwicklung nicht berücksichtigt. Da sie nun aber da sind und auf dem Arduino wie auch dem Tastaturstecker belegt wurden, steht künftigen Entwicklungen, die diese Pins benutzen, nichts im Wege.

Unterschiedlich sind bei allen Systemen die Timings, Ein-/Zwei-Wege-Kommunikation sowie die Scancodes und der Umgang mit den sogenannten Modifier Keys (Strg, Alt, Shift). Diese wurden in der jeweiligen Programmierung individuell für jedes System umgesetzt.

Die Timings sehen meist so aus, dass erst die Daten gesetzt werden und dann leicht zeitverzögert die Taktflanke folgt. Den Takt gibt immer die Tastatur vor, auch bei der Host-to-Device Kommunikation, die eine Besonderheit von AT bzw. PS/2 ist. Hier initiiert der Host einen Handshake über die Clock-Leitung (zieht auf LOW), worauf die Tastatur den Takt vorgibt, auf dem der Host seine Kommunikation schreibt.

Die Timings zum Schreiben auf die Tastatur sind (Quelle: www.kbdbabel.org Keyboard Signalling Collection):

10.5.1 Timings XT

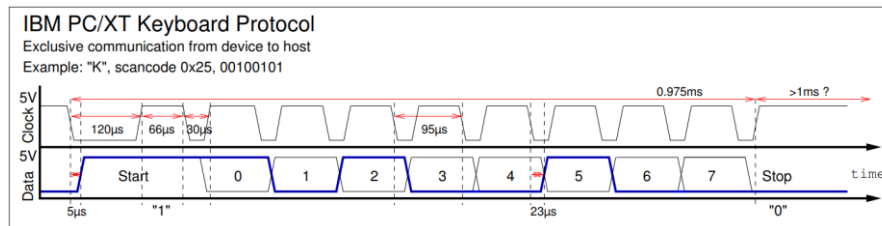


Abbildung 22: Timings XT

10.5.2 Timings AT und PS/2

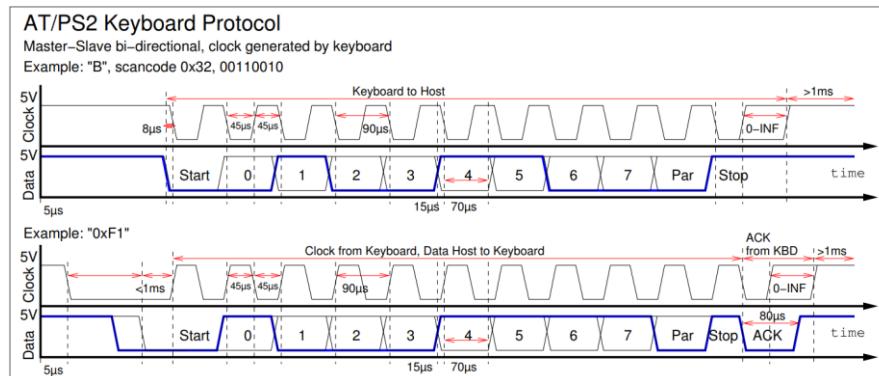


Abbildung 23: Timings AT und PS/2

10.5.3 Timings und Besonderheiten Amiga

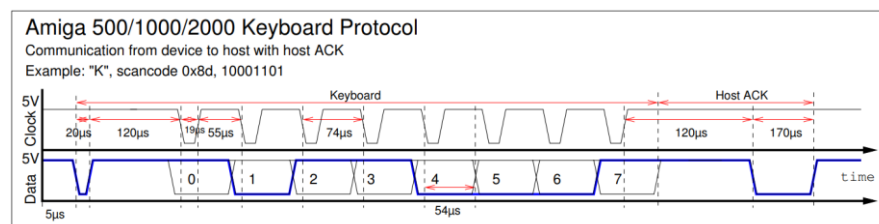


Abbildung 24: Timings Amiga

Zu beachten ist, dass hier das Timing-Diagramm (original und unverändert übernommen von kdbabel.org) die falsche Bitfolge angegeben wurde, der Amiga übermittelt die Bits rückwärts, aber erst ab Bit 6 bis 0 (7 Bit Tastenscancode höchstes Bit zuerst) und erst zuletzt Bit 7 (0=Make, 1=Break), also in der Reihenfolge 6-5-4-3-2-1-0-7. Zudem ist zu beachten, dass der Amiga invertierte Logik verwendet, 1=LOW und 0=HIGH.

Zudem löst die Amiga-Tastatur für den Amiga 2000 ein Reset-Warnung aus, wenn Ctrl+Amiga+Amiga gedrückt werden (spezieller Scancode) und wartet bis zu 10 Sekunden auf die Bestätigung des

Amigas, dass alle Notfall-Shutdown-Befehle abgeschlossen wurden und neugestartet werden kann. Der Amiga 500 ignoriert diese. Danach wird beim Amiga 500 die Keyboard Reset Leitung auf 0 gezogen bzw. beim Amiga 2000 das Clock-Signal für min. 500ms, wodurch ein Zählerchip den Reset auslöst. Es wurden hier beide Methoden implementiert, so sollte die Emulation für Amiga 500 und 2000 kompatibel sein, sollte die Keyboard Reset Leitung auf dem Amiga 2000 (habe leider keinen zum Testen) zu Problemen führen, muss deren digitalWrite auskommentiert werden.

10.5.4 Timings Tandy 1000

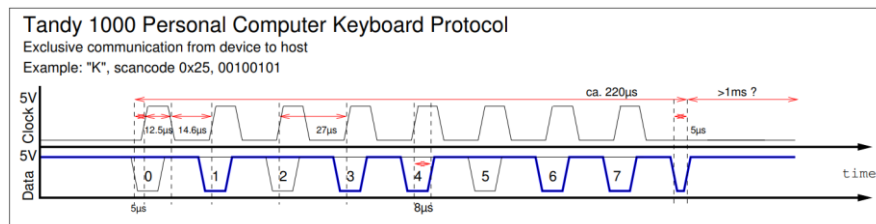


Abbildung 25: Timings Tandy 1000

10.6 CN6: PS/2 Mouse

Die PS/2 Maus verwendet die gleiche Kommunikation wie die AT- und PS/2 Tastatur. Daher lassen sich gewisse Routinen relativ problemlos recyceln.

Die Kommunikation ist, wie bei der Tastatur, bidirektional. Der Host kann so die Maus resetten/abfragen, aber auch steuern. Parameter wie Auflösung und Skalierungsfaktor werden erst einmal ignoriert, jedoch gespeichert, da der Host auch diese gegebenenfalls abfragt. Hier soll natürlich korrekt mit den zuvor gesetzten Werten geantwortet werden. Durch eine gewisse Folge von Auflösungsänderungen werden Mäuse mit Scrollrad in den Microsoft Intellimouse Modus versetzt und antworten mit Typ 3 statt Typ 0. Wir ignorieren das, da die USB Host Shield Library keine Scrollräder verarbeiten kann. Wir teilen dem PC mit dem wiederholten Ausgeben von Typ 0 mit: Hier ist eine normale 3-Tasten-Maus. Eine Intellimouse oder andere Scrollrad-Maus würde hier irgendwann mit 3 antworten und dem PC mitteilen, dass das Scrollrad vorhanden ist.

Mit Hilfe der Antworten erkennen BIOS (Stichwort Aktivierung des PS/2 Ports während des Selbsttests und Reservierung von IRQ 12) und Maustreiber, dass eine PS/2 Maus angeschlossen ist.

Der Standardmodus, in den die Maus vom Treiber versetzt werden sollte, ist der Stream-Mode. Hier werden mit jeder Bewegung folgende drei Datenpakete übertragen:

Bit Paket	7	6	5	4	3	2	1	0
1	X-Overflow	Y-Overflow	X-Sign	Y-Sign	1	M-Button	R-Button	L-Button
2	X7	X6	X5	X4	X3	X2	X1	X0
3	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

Die Overflows nutzen wir nicht, das Sign gibt die Richtung an, somit kann die Maus sich in einer Geschwindigkeit von +/- 0 bis 255 bewegen.

10.7 CN7: ADB

ADB ist ein Bus-System welches über Adressen und bidirektionaler Kommunikation arbeitet. Die Default-Adressen sind 2 für die Tastatur und 3 für die Maus. Der Host sendet wiederholt ADB-Befehle, welche Reset (Leitungs-Timing), Flush (Kommando 0 – in den meisten Implementierungen,

wie auch dieser, ignoriert), Listen (Kommando 2 – Host sendet Daten oder Befehle) oder Talk (Kommando 3 – das Device antwortet nun mit Datenpaketen wie gedrückten Tasten oder Mausbewegungen) sind.

Das ganze findet bidirektional über nur eine Datenleitung statt.

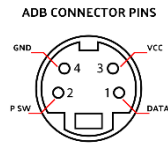


Abbildung 26: ADB Pinout

Der Power Switch Pin dient dazu, den Computer einzuschalten. Gerade bei älteren Macs ist am Computer selbst gar kein Power Schalter vorhanden, sondern wird, wenn man die entsprechende Taste auf der Tastatur drückt, durch Ziehen des Power Switch Pins auf Ground eingeschaltet.

11 Code-Quellen

- USB Host Shield Library – Beispiel-Codes für die Circuits@Home USB Host Shield Library 2.0 sowie die Library selbst von felis, [GitHub - felis/USB_Host_Shield_2.0: Revision 2.0 of USB Host Library for Arduino.](https://github.com/felis/USB_Host_Shield_2.0) / https://github.com/felis/USB_Host_Shield_2.0
- Ansteuerung der Tandy 1000 Tastatur: PS/2 to Tandy Converter, Adrian Black (Adrian's Digital Basement), angepasst auf die Belange der USB-Tastatur-Scancodes. Daraus hergeleitet mit anderen Timings und weiteren nötigen Anpassungen alle weiteren Tastatur-Ausgabetreiber für den Keyboard Port, [GitHub - misterblack1/ps2-to-tandy1000-keyboard: PS2/AT Keyboard to Tandy 1000 Keyboard Adapter](https://github.com/misterblack1/ps2-to-tandy1000-keyboard) / <https://github.com/misterblack1/ps2-to-tandy1000-keyboard>
- Kleinere Teile für AT/PS/2 Maus/Tastatur von [ps2dev/src at master · Harvie/ps2dev · GitHub](https://github.com/Harvie/ps2dev/tree/master/src) / <https://github.com/Harvie/ps2dev/tree/master/src>
- Kleinere Teile von Apple PS/2 to ADB von schenapan, [GitHub - schenapan/PS2TOADB: simple ps2 mouse/keyboard arduino adapter to adb for old macintosh](https://github.com/schenapan/PS2TOADB) / <https://github.com/schenapan/PS2TOADB>

12 Abbildungsverzeichnis

Abbildung 1: Die Vision – Eine KVM für alles!	- 1 -
Abbildung 2: Speedlink Competition Pro	- 13 -
Abbildung 3: Speedlink Phantom Hawk	- 14 -
Abbildung 4: Logitech Extreme 3D Pro.....	- 15 -
Abbildung 5: Logitech F710	- 16 -
Abbildung 6: Thrustmaster TC Sidestick.....	- 16 -
Abbildung 7: Playstation 2 Dual Analogue / Trio Linker Plus	- 17 -
Abbildung 8: Dreamcast Controller / Trio Linker Plus	- 18 -
Abbildung 9: Das Projekt in Einzelteilen.....	- 23 -
Abbildung 10: Lötbrücken zum Jumpen auf 5V-Modus.....	- 24 -
Abbildung 11: Einsetzen und anlöten der Pinreihen.....	- 25 -
Abbildung 12: Bestückte Platine beim Einlöten der Verbinder zum Arduino	- 25 -
Abbildung 13: Verschraubung Arduino	- 28 -
Abbildung 14: Befestigung 3-Headed Monkey und Reset-Führungshülse.....	- 28 -
Abbildung 15: Deckel aufsetzen	- 29 -
Abbildung 16: Fertig ist das Gehäuse	- 30 -
Abbildung 17: Pinout Analog Joystick-Port.....	- 32 -
Abbildung 18: Joystick-Spannungsteiler in verschiedenen Computer-Typen.....	- 33 -
Abbildung 19: Commodore/Atari Digital Joystick Port	- 35 -
Abbildung 20: Commodore und Atari Mouse Port	- 35 -
Abbildung 21: 5-Pin 180° (PC, Amiga) und 8-Pin 270° (Tandy 1000) Tastatur-Pinout Buchsenseite	- 36 -
Abbildung 22: Timings XT	- 37 -
Abbildung 23: Timings AT und PS/2	- 37 -
Abbildung 24: Timings Amiga	- 37 -
Abbildung 25: Timings Tandy 1000	- 38 -
Abbildung 26: ADB Pinout	- 39 -