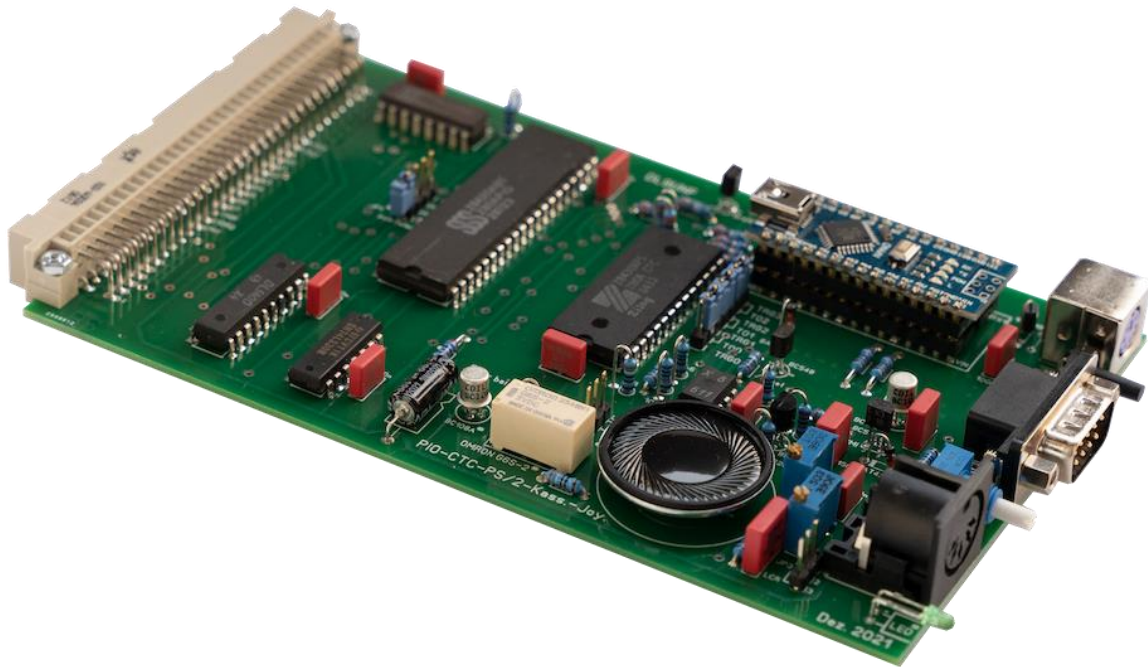


# Beschreibung AC1-2010 PIO-Karte mit PS/2-Tastatur und Joystick

Stand: 17.01.2022



## Motivation:

Für den Anschluss einer PS/2-Tastatur an den AC1-2010 gibt es bereits Lösungen aus den Jahren 2011 und 2017, bei denen die Umsetzung des seriellen Signals der PS/2-Tastatur in 8 Bit für den AC1 mit einem PIC16F877 realisiert wird. Leider steht der Quellcode nicht zur Verfügung, so dass nur die Funktionstasten nach eigenen Wünschen programmiert werden können.

Für die Arbeit in CP/M-Betriebssystem sind die AC1-Kursortastencodes nur bedingt geeignet, da die ↑-Taste den Wert 0x0B = ^K erzeugt. Dieser Code wird aber in den CP/M-Programmen Wordstar und Turbo-Pascal für zahlreiche andere Steueraufgaben eingesetzt. Daher ist eine Umschaltung der Steuertasten-Codes wünschenswert, wenn mit dem AC1-Monitor und dem Betriebssystem CP/M gearbeitet wird.

Zum Schluss war es eine nette Idee, einen Joystick-Anschluss zu integrieren. Auch wenn ich in den Anfangsjahren des AC1 viel programmiert und wenig gespielt habe, hat das Retro-Feeling schon etwas Besonderes...

Daraus entstand eine modifizierte Leiterplatte mit folgenden Änderungen:

- Austausch des PIC16F877 durch einen Arduino Nano,
- Ersatz der Klinkenbuchse durch einen Lautsprecher,
- Joystickbuchse für den Anschluss eines Atari-Joysticks mit 5 Tastern,
- Zusätzliche Stützkondensatoren zur Minimierung der HF-Anteile.

Für die Programmierung des Arduino standen folgende Anforderungen:

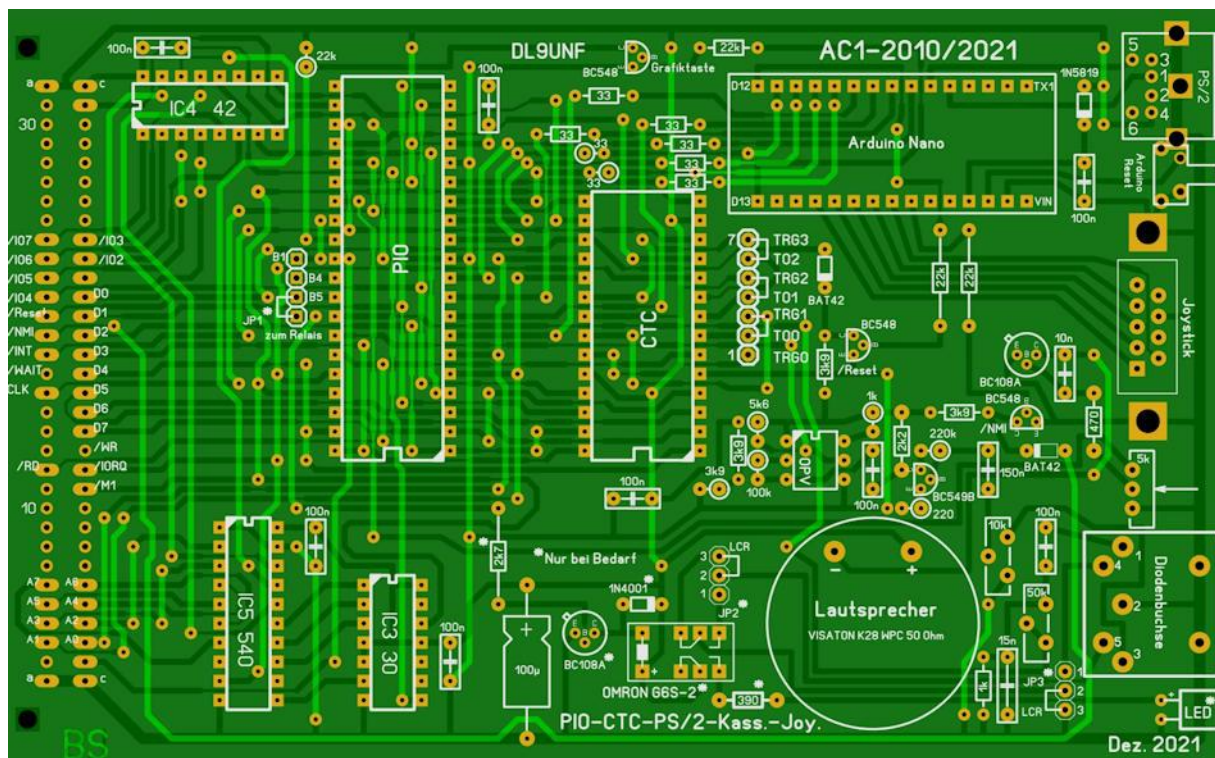
- Kompatibilität der Tastenbelegung zur PIC16F877-Lösung,
- Nutzung von NUM-Lock als SCCH-Grafiktaste,
- Unterstützung von AC1-Reset und AC1-NMI,
- Umschaltung auf CP/M-Tastencodes mit der Rollen-Taste,
- Joystick-Abfrage und Erzeugung der Codes,
- Konfigurationsdatei für alle wichtigen Einstellungen.

### Schaltplan:

Die Vorlage für den Schaltplan war die PIO-Karte für den AC1-2010 aus dem Jahr 2011. Die durchgeführten Änderungen wurden entsprechend eingepflegt. Der Schaltplan ist am Ende des Textes angefügt.

### Layout für die Leiterplatte:

Die Leiterplatte wurde mit Sprintlayout erstellt. Als Vorstück diente die Leiterplatte der PIO-Karte für den AC1-2010 aus dem Jahr 2011. Die Positionen und Verschaltung der PIO und CTC sind weitgehend unverändert. Nachfolgend ist das Layout abgebildet:



### Software für den Arduino:

Die Software wurde von Gerd Staffen und Frieder Tonn entwickelt. Die aktuelle Version steht auf [github.com](https://github.com/friedertonn/AC1-Tastatur): <https://github.com/friedertonn/AC1-Tastatur>

Für die Programmentwicklung wurde die Arduino-IDE V1.8.8 verwendet. Sie bietet die Möglichkeit, Status- und Fehlermeldungen über einen Seriellen Monitor auszugeben. Zum Programmieren des Arduino müssen das Board „Arduino Nano“, der Prozessor „ATmega328P“ (bei China-Importen häufig „ATmega328P (Old Bootloader)“) sowie der COM-Port eingestellt werden.

Ein wesentliches Element des Programms ist die Bibliothek "PS2KeyAdvanced" von Paul Carpenter. Sie verfügt über umfangreiche Funktionen zur Tastatursteuerung inklusive verschiedener Betriebsarten, dem Abfragen und Ansteuern der LEDs sowie einem Tastatur-Reset. Der Quellcode und eine Dokumentation stehen hier: <https://github.com/techpaul/PS2KeyAdvanced>

Die Datei „AC1-Tastatur.ino“ initialisiert die Tastatur und wandelt die empfangenen 16bit-Codes in einen 7bit-Code für den AC1. In der Datei „config.h“ können Anpassungen nach persönlichen Erfordernissen vorgenommen werden. Sie ist weitgehend selbsterklärend.

### Schreibmaschinenblock:

Die Tasten im Schreibmaschinenblock sind mehrfach belegt. Die Umschaltung erfolgt mit Shift, Ctrl, Alt und AltGr. Im Gegensatz zur Tastatur mit dem PIC16F877 lassen sich die Codes @, [, \, ], {, |, } und ~ auch mit AltGr aufrufen. Je nach eingestelltem Zeichensatz werden die oben aufgeführten Zeichen oder die deutschen Umlaute auf dem Bildschirm ausgegeben.

### Funktionstasten:

Die Funktionstasten werden in der Datei „config.h“ definiert. Insgesamt stehen 4 Ebenen (ohne Shift, mit Shift, mit Ctrl und mit Alt) für jeweils AC1- und CP/M-Zeichenkodierung zur Verfügung. Das sind insgesamt 96 mögliche Funktionstasten-Befehle. Bei Bedarf ließe sich das noch weiter ausbauen (z. B. Shift+Ctrl+F1, Windows+F1, Menü+F1), wurde aber als nicht notwendig erachtet. Für die Funktionstaste F8 sind in der Datei „config.h“ exemplarisch Zeichenketten eingetragen, die bei Tastendruck ausgegeben werden.

Die Zeichenketten können auch Steuerzeichen enthalten, diese müssen dann als Oktal-Code angegeben werden. Der Code \015 steht für die Entertaste und \007 gibt auf dem SCCH-Monitor ein Tonsignal aus. Für die Umrechnung zwischen Dezimal-, Hexadezimal- und Oktal-Code leistet der Windows-Taschenrechner im Programmierer-Modus gute Dienste.

Der Code \037 nimmt eine Sonderstellung ein: \037xy (wobei x und y Ziffern zwischen 0 und 9 sind) erzeugt eine Verzögerung von x,y Sekunden. Die maximale Verzögerung beträgt also 9,9 Sekunden. Damit können häufig wiederkehrende Abläufe realisiert werden, z. B. Programmpaket X starten – kurz warten – ein Programm laden – kurz warten – das Programm starten. Die Zeichenkette darf insgesamt nicht länger als 63 Zeichen sein, wobei der Oktal-Code jeweils als 1 Zeichen zählt.

Mit dem Code \037C1 kann von der AC1- in die CP/M-Zeichenkodierung gewechselt werden. Der Code \037C0 schaltet wieder in die AC1-Kodierung zurück.

Code:	Befehl:
\"	Anführungszeichen z. B. bei Ladebefehlen von USB bzw. GIDE
\007	im SCCH-Monitor wird ein Tonsignal ausgegeben
\015	Enter
\032	^Z – Zeichensatzumschaltung ACC/SCCH
\037xy	Pause x,y Sekunden
\037C1	schaltet auf Tastenbelegung CP/M, z. B. beim Start von CP/M
\037C0	schaltet auf Tastenbelegung AC1, z. B. beim Verlassen von CP/M

Die nachfolgende Tabelle zeigt die Belegung der Funktionstasten, wie sie in der Datei „config.h“ voreingestellt ist. Wenn die BWS-Karte mit CPLD vorhanden ist, kann mit Shift+F4 der CPU-Takt auf 4 MHz umgeschaltet werden. Die Tastenkombination Shift+F2 schaltet auf 2 MHz zurück.

	AC1-Kodierung				CP/M-Kodierung			
	ohne Shift	mit Shift	mit Ctrl	mit Alt	ohne Shift	mit Shift	mit Ctrl	mit Alt
F1	?↓				FC↓			
F2		WFO 0↓			DIR↓			
F3								
F4	CLS↓	WFO 1↓						
F5	CLOAD							
F6	CSAVE							
F7								
F8	Teststring	Teststring	Teststring	Teststring	Teststring	Teststring	Teststring	Teststring
F9	STOP↓							
F10	RUN↓							
F11	LIST↓							
F12	CONT↓			Version	EXIT↓			

### Steuerblock auf der Tastatur:

Der Steuerblock auf der PS2-Tastatur erzeugt die nachfolgend angegebenen Tastencodes. Die Rollen-Taste dient der Umschaltung zwischen dem AC1- und CP/M-Modus. Dies wird durch die Tastatur-LED signalisiert. In der Konfigurationsdatei „config.h“ kann festgelegt werden, welcher Modus nach einem Arduino-Reset aktiv ist. Die CP/M-Tastencodes sind kompatibel zum CP/A-Standard, so dass viele CP/M-Programme von Robotron-Rechnern ohne Änderungen verwendet werden können.

Steuertasten: AC1-Code

Drucken	Rollen	Pause
^X		^S

Einfg	Pos1	Bild ↑
^E	^A	^U
Entf	Ende	Bild ↓
^D	^C	^Q

	↑	
←	↓	→
^H	^J	^I

Steuertasten: CP/M-Code

Drucken	Rollen	Pause
^P		^S

Einfg	Pos1	Bild ↑
^V	^QS	^R
Entf	Ende	Bild ↓
^G	^QD	^C

	↑	
←	↓	→
^H	^X	^D

Die Rollen-Taste schaltet zwischen der AC1- und der CP/M-Codierung um

### Ziffernblock:

Der Ziffernblock verhält sich wie erwartet, d. h. mit NUM-Lock kann zwischen Kursortasten und Ziffern umgeschaltet werden. Zu beachten ist, dass die NUM-Lock-Taste bei diesem Projekt die Zweitfunktion als SCCH-Grafiktaste erfüllt. Wenn im GBASIC Grafikzeichen programmiert werden sollen, muss NUM-Lock aktiviert sein. Umgekehrt ist es so, dass beim Schreiben von Texten Kleinbuchstaben nur bei deaktiviertem NUM-Lock möglich sind. Ich hoffe, damit kommt niemand durcheinander...

### Spezial-Tastenkombinationen:

Die Spezial-Tastenkombinationen dienen dazu, verschiedene Kommandos auszuführen. Sie sind in der nachfolgenden Tabelle aufgelistet.

Tastenkombination:	Befehl:
Ctrl+ESC	AC1-NMI
Alt+ESC	AC1-Reset und PS/2-Tastatur-Reset
Ctrl+Alt+Entf	AC1-Reset
NUM-Taste	NUM-Lock + SCCH-Grafiktaste ein/aus
Rollen-Taste	Umschaltung Tastencodes: AC1 - CP/M
AltGr+0	40ms-Impulse auf PA7, siehe nächster Abschnitt
AltGr+1	PA7 ist aktiv, bis Taste losgelassen wird

### Parallelausgabe an den AC1:

Für die Parallelausgabe an den PIO-Port A am AC1 stehen 8 Signalleitungen zur Verfügung. Die Signalleitungen an PA0 bis PA6 übermitteln eine 7bit-Codierung 0x00 bis 0x7F. Die 8. Signalleitung an PA7 zeigt die Gültigkeit des Codes an. Bei den ersten Tastatur-Bauanleitungen für den AC1 hatte PA7 High-Pegel, so lange die Taste gedrückt war. Wenn man jedoch mit dem Programm EDAS\*4 und 4 MHz CPU-Takt arbeitet, funktioniert das nicht mehr, da die im EDAS\*4 integrierte Tastaturabfrage

nach sehr kurzer Zeit in den Autorepeat-Modus wechselt. Als Workaround muss dann bei jedem Tastendruck an PA7 ein High-Impuls mit 40 Millisekunden Länge erzeugt werden. Dann funktionieren jedoch einige Spiele nicht mehr zuverlässig. Um beide Möglichkeiten abzudecken, gibt es (wie bereits beim PIC16F877) zwei Methoden der Signalerzeugung für PA7:

- Tastenkombination AltGr+1: "Standard-Modus" – PA7 ist aktiv, bis die Taste losgelassen wird
- Tastenkombination AltGr+0: "Ralph's Spezial-Modus" – 40ms-Impuls auf PA7

In der Datei „config.h“ kann die Initial-Einstellung festgelegt werden. Der „Spezial-Modus“ ist für Schnellschreiber besser geeignet, da sich die Tastatur hier „flüssiger“ anfühlt. Das Retro-Feeling gibt es im „Standard-Modus“.

### Verhalten der Software beim Programmstart und nach Alt+ESC:

Die Software kann nach jedem Arduino-Reset bzw. der Tastenkombination Alt+ESC eine Zeichenfolge ausgeben, z. B. "WF0 1\015", um den CPU-Takt auf 4 MHz umzuschalten. Die Definition dazu steht in der Datei „config.h“. Weitere denkbare Szenarien sind AC1-Monitor-Initialisierungen wie V.24, Color, Druckertreiber, usw. (ähnlich AUTOEXEC.BAT). Bei der Tastenkombination Ctrl+Alt+Entf (Affengriff) wird die Zeichenfolge nicht aufgerufen.

### Joystick-Modi:

Für die Abfrage eines Atari-Joysticks mit 5 Tastern gibt es im SCCH-Monitor die Funktion 0x0EB4. Die Leitung B1 von der AC1-PIO wird hier für die Joystick-Abfrage nicht benötigt. Im AC1-BASIC6 ist das JOY-Kommando implementiert, das ebenfalls diese Routine verwendet. Alternativ kann der Joystick Tastendrucke emulieren. Mit der Tastenkombination AltGr+Funktionstaste wird zwischen den Modi umgeschaltet. Insgesamt sind 12 Modi möglich. Der Initial-Modus und die Tasterbelegungen können in der Datei „config.h“ festgelegt werden. Die nachfolgende Tabelle gibt einen Überblick über die bisher definierten Codes:

Tastencode:	Funktion:	hoch	runter	links	rechts	Feuer
AltGr+F1	SCCH-Monitor Joystickabfrage 0x0EB4	0x01	0x02	0x04	0x08	0x10
AltGr+F2	Kursortasten + Enter, z. B. Ghosttown	^K	^J	^H	^I	0x0D
AltGr+F3	Kursortasten + Space, z. B. ACBALL	^K	^J	^H	^I	0x20
AltGr+F4	Belegung für ACTRIS	0x20	0x0D	^H	^I	0x20
AltGr+F5	Ziffernblock + Space, z. B. WURMI	'5'	'2'	'1'	'3'	0x20
AltGr+F6	Belegung für PACMAN	'T'	'V'	'F'	'G'	0x0D
AltGr+F7	frei					
⋮						
AltGr+F12	frei					

### Aufbauhinweise:

Am Dateiende ist eine Bauteilliste angegeben. Bei einigen Komponenten (z. B. DIN-Buchse) ist die Bauteilnummer eines Elektronik-Versenders angegeben. Die Bohrungen auf der Leiterplatte sind passend zu diesen Teilen.

Für die Montage des Arduino werden IC-Steckadapter empfohlen, dann kann man den Arduino in eine Präzisions-IC-Fassung stecken. Die mitgelieferten Kontaktkämme sind dafür zu dick. Da es keine 30-pol. Stecker/Fassungen gibt, benötigt man auch eine Säge. (Ganz Mutige können den Arduino natürlich direkt auf die Leiterplatte löten, das Programmieren kann im laufenden Betrieb erfolgen – sozusagen „Operation am offenen Herzen“. Man sollte vorher nur testen, ob bei den 33 Ohm Widerständen genug Platz für den Mini-USB-Stecker ist.) Bei der Auswahl des Arduino Nano Boards ist es ratsam, eine Platine ohne aufgelöteten 3x2 Pfostenstecker zu verwenden, da es ansonsten zu Kollisionen der Kontakte mit der benachbarten Leiterplatte kommen kann. Die Kontakte des 3x2



Pfostensteckers werden in diesem Projekt nicht benötigt. Die Programmierung des Arduino kann über den Mini-USB-Anschluss erfolgen.

Die DIN-Buchse sollte vor dem Einbau auf der Oberseite um 1,5 mm gekürzt werden. Ansonsten kann es Schwierigkeiten mit der benachbarten Leiterplatte geben.

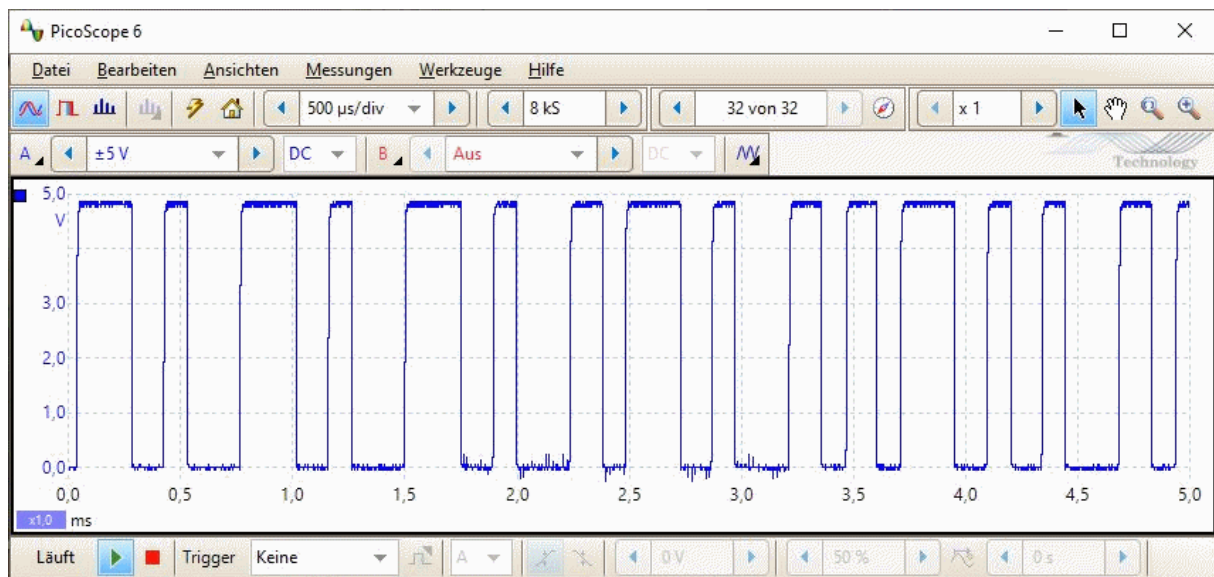
Der Lautsprecher ist mit dicken Drähten und einem Abstand von ca. 1mm „freischwebend“ auf die Leiterplatte gelötet.

### **Abgleich des Kassetteninterfaces:**

Nachfolgend wird der Abgleich mit einem Datenrekorder LCR-C beschrieben. Bei anderen Rekordern dürfte es keine großen Unterschiede geben.

Kassetten-Aufnahme: Den 50 KOhm-Einstellwiderstand auf 22 KOhm einstellen, fertig.

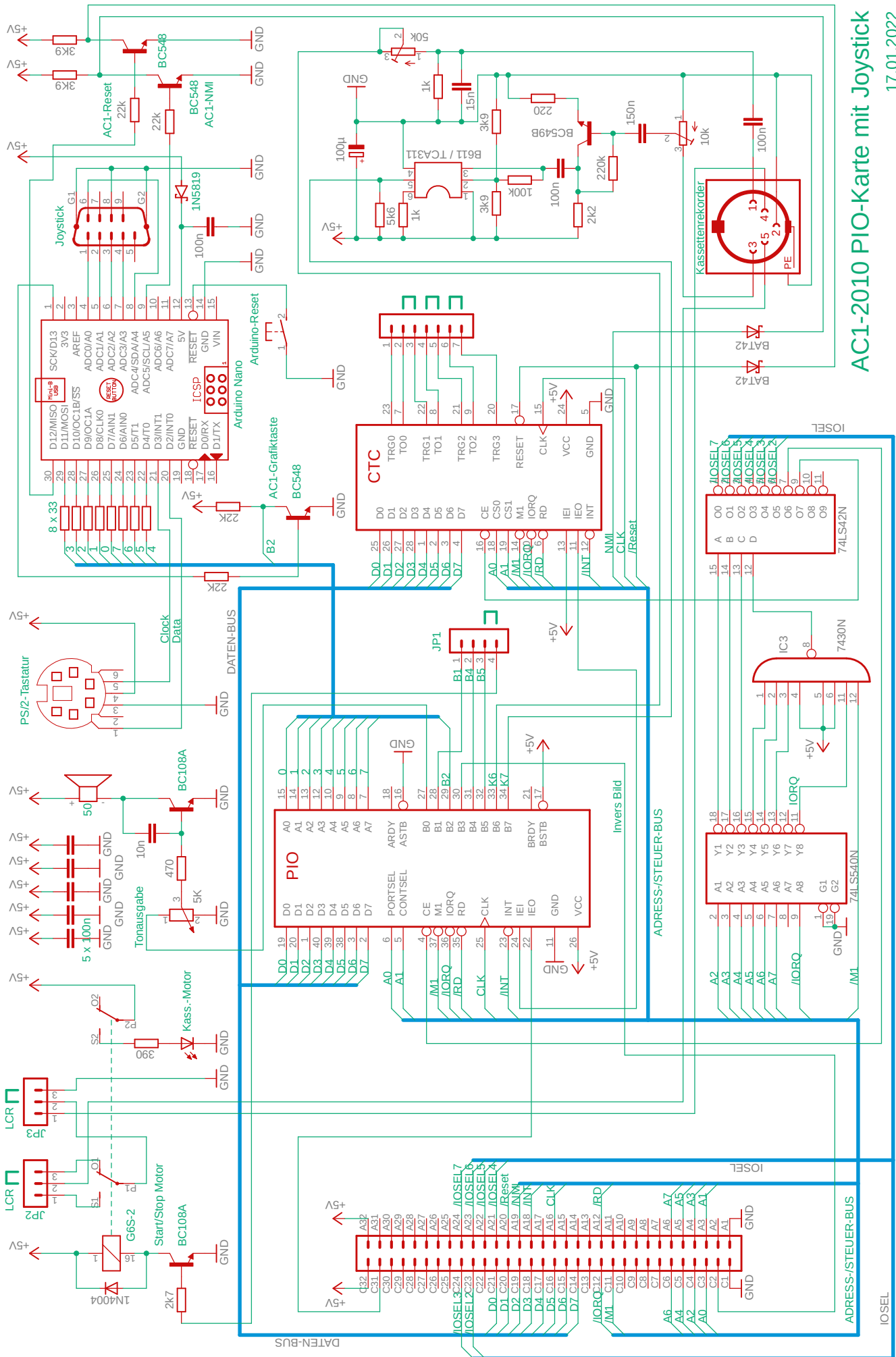
Kassetten-Wiedergabe: Den Schleifer am 10 KOhm-Einstellwiderstand auf Masse einstellen. Eine Kassette mit SCCH-Musik in den Kassettenrekorder einlegen und dudeln lassen. An Pin 5 vom Operationsverstärker ein Oszilloskop anschließen. Jetzt den Einstellwiderstand langsam aufdrehen, bis ein sauberes Rechtecksignal zu sehen ist. Bei guter Kassettenwiedergabe dürfen in den Pausen zwischen den Programmen höchstens einzelne Peaks zu sehen sein.



Zum Schluss ganz herzlichen Dank an das Team der Prototyp-Leiterplatten-Tester Dietmar Badtke, Gerd Staffen, René Vetter und Andreas Zeinert für die vielen Hinweise und Verbesserungsvorschläge.

Viel Erfolg beim Aufbau und der Inbetriebnahme wünscht  
Frieder Tonn

Für Fragen und Hinweise: [Frieder.Tonn@gmx.de](mailto:Frieder.Tonn@gmx.de) und [mail@gerd-staffen.de](mailto:mail@gerd-staffen.de)



AC1-2010 PIO-Karte mit Joystick  
17.01.2022

Stückliste AC1-2010 PIO-Joystick-Karte Stand: 06.01.2022		
	Anzahl	Bemerkung
Schaltkreise:		
74LS30	1	alternativ DL030
74LS42	1	
74LS540	1	alternativ DL540
Z80-PIO	1	das Rastermaß ist 2,54 mm, U855 wird schwierig
Z80-CTC	1	das Rastermaß ist 2,54 mm, U857 wird schwierig
Arduino Nano	1	funktioniert auch mit Arduino-Klones
B611	1	alternativ B621, TCA311
IC-Fassungen:		
6 pol.	1	
14 pol.	1	
16 pol.	1	
20 pol.	1	
28 pol., breit	1	
30 pol., breit	1	für den Arduino, längere IC-Fassung kürzen, Säge erforderlich
40 pol., breit	1	
IC-Steckadapter 24 pol.	2	zum Anlöten an den Arduino, Reichelt: AR 24, Säge erforderlich
Transistoren:		
BC548, npn	3	alternativ Schalttransistoren SS216, SS218
BC108A, npn 500mW	2	
BC549B, npn	1	



Stückliste AC1-2010 PIO-Joystick-Karte Stand: 06.01.2022		
	Anzahl	Bemerkung
Kleinteile:		
33R	8	
220R	1	
390R	1	
470R	1	
1k	2	
2k2	1	
2k7	1	
3k9	4	
5k6	1	
22k	4	
100k	1	
220k	1	
5k, Potentiometer	1	Reichelt: BOU 3310C001502L
10k, Einstellregler	1	Reichelt: 64Y-10K
50k, Einstellregler	1	Reichelt: 64Y-50K
10n RM5	1	
15n RM7.5	1	
100n RM5	8	
150n RM5	1	
100µ	1	Reichelt: COS 105 100/10
1N4001	1	
1N5819	1	Schottky-Diode
BAT42	2	
LED	1	
Relais	1	Conrad: OMRON G6S-2
Mini-DIN-Buchse	1	Reichelt: EB-DIOS M06V
Diodenbuchse	1	Reichelt: MABP 5S, 1,5mm oben absägen
D-Sub-Stecker 9 pol.	1	Reichelt: BKL 10120037
Stiftleiste, 3 pol.	2	
Stiftleiste, 4 pol.	1	
Stiftleiste, 7 pol.	1	
Messerleiste, 64 pol.	1	Reichelt: ML-C 64W
Lautsprecher 50 Ohm	1	Reichelt: VIS 2810
Reset-Taste	1	Reichelt: TASTER 3305D