Kassetteninterface

Der AC1 benutzt in seiner originalen Ausbaustufe (FA, Monitor 3.1) ein eigenes Aufzeichnungsformat, dass an den ZETBUG angelehnt ist.

Von F.Ludwig (SCCH) stammt ein Turbo-Tape, dass das Aufzeichnungsverfahren des Poly-Computers und des Z1013 nutzt, aber mit abweichenden Zeitkonstanten.

AC1-Standard

Das Prinzip entspricht dem in [12] vorgestellten Kassetteninterface. Die dort angegebene Interfaceschaltung wurde dem Grundmodul des „AC1“ angepaßt. Die Software dazu wurde neu geschrieben, das Aufzeichnungsformat entspricht etwa den in [10] verwendeten.

Achtung: Das Kassettenaufzeichnungsformat des AC1 enthält nicht nur Speicherbereiche. Vielmehr ist es ein Containerformat mit verschiedenen Abschnitten (so gibt es z.B. auch Ladebilder). BASIC-Programme zum großen Basic V1.1 müssen als Audio-Datei geladen werden, ein Laden eines binären Dumps ist nicht direkt möglich.

Aufbau:

1. Vorton 512x 00-Bytes
2. Synchronisationsmuster E6h
3. 'U' [Kennung Name] 6(16) Zeichen Dateiname, mit Leerzeichen aufgefüllt
4. Vorton 256x 00-Bytes [Zeit für Namensdarstellung beim Einlesen]
5. Datenblöcke: '<' [Kennung Block], je 1 Byte Blocklänge + max. 256 Byte Daten + 1 Byte Prüfsumme, Blocklänge 0 == 256 Byte
6. 'x' [Kennung Startadr] 2 Byte startadr.

Die Startadr wird nach Einlesen nur angezeigt und in ARG1 abgelegt, das Programm startet nicht automatisch! Mit J: wird das Programm gestartet.

Datei:

<ditaa noedgesep> +——-+—-+——-+——-+———+———+———+ +———+——-+

vorton	E6	U Name	vorton	< Block	< Block	< Block	...	< Block	x sadr

+——-+—-+——-+——-+———+———+———+ +———+——-+ </ditaa>

Daten-Block '<':

max 256 Datenbytes. Obwohl jeder Block in der Anzahl variieren kann, hat nur der letzte Block < 256 Bytes

<ditaa noedgesep> +—+——+——-+—-+——-+——+

<	len	byte 0	...	byte255	cks

+—+——+——-+—-+——-+——+ </ditaa>

Beispielprogramm CLIST@ von der Musterkassette. Der Datenstrom besteht aus 4 Blöcken (Name + 256×00, 256 Datenbytes, 224 Datenbytes, Startadresse). Das ist Dump des Bytestreams, wie er auf Kassette abgelegt wird, und KEIN Binärdump des Programms im Speicher!

```
00000000: 55 43 4C 49 53 54 40 20|20 20 20 20 20 20 20 20 | UCLIST@
00000010: 20 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 |  ........
00000020: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 | ........
00000030: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 | ........
00000040: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 | ........
00000050: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 | ........
00000060: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 | ........
00000070: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 | ........
00000080: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 | ........
00000090: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 | ........
000000A0: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 | ........
000000B0: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 | ........
000000C0: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 | ........
000000D0: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 | ........
000000E0: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 | ........
000000F0: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 | ........
00000100: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 | ........
00000110: 00 3C 00 00 19 4A 19 3F|19 00 09 40 0D 11 C4 09 | .<...J.?...@..Ä.
00000120: 01 00 00 D9 DF 0C 8D C3|8B 1A 3E 19 ED 47 21 3A | ...ÙB..Ã.>.íG!:
00000130: 19 01 07 05 ED B3 AF D3|00 ED 5E FB DB 04 FE 83 | ....í³¯Ó.í^ûÛ.þ
00000140: 20 FA 3E 03 F3 CD 7B 1A|D3 00 D3 07 C3 FD 07 02 |  ú>.óÍ{.Ó.Ó.Ãý..
00000150: CF 80 B7 7F 3E 87 D3 00|3E 3C D3 00 FB ED 4D D9 | Ï€·.>‡Ó.><Ó.ûíMÙ
00000160: 08 3E 03 D3 00 DB 05 CB|7F 20 09 1B AF BA 20 07 | .>.Ó.Û.Ë. ..¯º .
00000170: 21 67 19 E5 11 C4 09 D9|08 FB ED 4D E1 3E 03 D3 | !g.å.Ä.Ù.ûíMá>.Ó
00000180: 07 D3 00 D9 03 C5 D9 E1|AF 57 06 10 29 8F 27 5F | .Ó.Ù.ÅÙá¯W..).'_
00000190: 7A 8F 27 57 7B 10 F5 EB|DF 20 A0 CD F1 07 DF 20 | z_'W{.õëß ÍÑ.ß
000001A0: A0 AF 32 D8 1A CD 3B 1A|FE E6 28 09 FE 19 20 F5 |  ¯2Ø.Í;.þæ(.þ. õ
000001B0: 3E FF 32 D8 1A CD 62 1A|FE 55 20 F9 06 06 2A 00 | >ÿ2Ø.Íb.þU ù..*.
000001C0: 18 CD 62 1A 77 2B 10 F9|2B 22 00 18 EB 21 D9 1A | .Íb.w+.ù+"..ë!Ù.
000001D0: 36 00 CD 62 1A FE 78 28|58 FE 3C 20 F5 CD 62 1A | 6.Íb.þx(Xþ< õÍb.
000001E0: 47 CD 06 1A 85 4F CD 62|1A F5 3A D9 1A A7 20 07 | GÍ..…OÍb.õ:Ù.§ .
000001F0: 3C 32 D9 1A 22 DA 1A F1|23 81 4F 10 E9 CD 62 1A | <2Ù."Ú.ñ#.O.éÍb.
00000200: B9 28 CF DF 20 20 43 48|45 43 4B 53 55 4D AF C3 | ¹(ÏB  CHECKSUM¯Ã
00000210: 8B 1A DF 20 20 76 3C E0|00 1A 20 A0 CD F1 07 C9 | ‹.ß  v<à..  ÍÑ.É
00000220: CD 62 1A 6F CD 62 1A 67|EB CB 4E 28 04 36 20 18 | Íb.oÍb.gëËN(.6 .
00000230: 02 36 5E EB 7C C9 2B 22|DC 1A CD 06 1A 22 DE 1A | .6^ë|É+"Ü.Í.."Þ.
00000240: 2A DA 1A CD FD 19 2A DC|1A CD FD 19 2A DE 1A CD | *Ú.Íý.*Ü.Íý.*Þ.Í
00000250: FD 19 C3 8B 1A C5 F5 DB|05 E6 80 4F DB 05 E6 80 | ý.Ã‹.ÅõÛ.æ€OÛ.æ€
00000260: B9 28 F9 4F 06 3F 10 FE|CD 58 1A F1 CB 11 CB 17 | ¹(ùO.?.þÍX.ñË.Ë.
00000270: C1 C9 DB 04 FE 83 C0 DF|8D C3 2D 19 C5 06 08 CD | ÁÉÛ.þƒÀß.Ã-.Å..Í
00000280: 3B 1A 10 FB 4F 3A D8 1A|A9 C1 C9 4F 58 37 32 59 | ;..ûO:Ø.©ÁÉOX72Y
00000290: 20 29 43 28 20 21 05 10|11 06 10 01 0A 00 D5 C5 |  )C( !........ÕÅ
000002A0: ED B0 C1 D1 C9 CD 7B 1A|21 71 1A DF 8D ED B0 11 | í°ÁÑÉÍ{.!q.ß.í°.
000002B0: FF 17 21 D6 1A 01 28 00|ED B8 06 27 3E 2D 21 BE | ÿ.!Ö..(.í¸.'>-!¾
000002C0: 17 77 2B 10 FC C3 15 19|20 54 52 41 54 53 20 20 | .w+.üÃ.. TRATS
000002D0: 20 20 45 44 4E 45 20 20|20 47 4E 41 46 4E 41 20 |   EDNE   GNAFNA
000002E0: 20 20 20 20 45 4D 41 4E|20 20 20 20 52 4E 20 20 |     EMAN    RN
000002F0: 20 20 FF 01 00 19 DF 1A|00 00 04 78 00 00 00 00 |   ÿ...ß....x....
```

Das **8K-Basic 1.1** verwendet einen eigenen Dateiaufbau !!!

1. 512 x 00-Byte Vorton
2. E6 Syncbyte
3. 6 Byte Name

4. D3 Kennung Basic
5. 2 Byte Laenge (hi, lo)
6. xxx Bytes

<ditaa noedgesep> +——-+—-+——+—-+——-+———-+

vorton	E6	Name	D3	laenge	Programm

+——-+—-+——+—-+——-+———-+ </ditaa>

Beispielprogramm LOCATO von der Musterkassette, Länge Programmblock 0F92h Bytes. Das ist Dump des Bytestreams, wie er auf Kassette abgelegt wird, und KEIN Binärdump des Programms im Speicher!

```
00000000: 4C 4F 43 41 54 4F D3 0F|92 04 61 05 00 8E 55 4C  | LOCATOÓ.'.a..■UL
00000010: 4F 43 41 54 4F 00 0E 61|0A 00 99 3A 9E 3A 9E 00  | OCATO..a..■:■:■.
00000020: 38 61 14 00 9E 22 20 20|20 20 20 50 52 4F 47 52  | 8a..■"     PROGR
00000030: 41 4D 4D 20 46 55 45 52|20 44 45 4E 20 55 4B 57  | AMM FUER DEN UKW
00000040: 2D 41 4D 41 54 45 55 52|22 00 62 61 1E 00 9E 3A  | -AMATEUR".ba..■:
00000050: 9E 22 28 43 29 20 20 59|32 33 4C 41 20 2F 20 59  | ■"(C)  Y23LA / Y
00000060: 32 2D 31 38 37 35 35 2F|41 34 32 20 20 20 31 39  | 2-18755/A42   19
00000070: 38 35 22 00 89 61 28 00|9E 22 20 20 20 20 20 45  | 85".■a(.■"     E
00000080: 52 57 45 49 54 45 52 55|4E 47 45 4E 3A 20 59 32  | RWEITERUNGEN: Y2
```

Beschreibung aus dem FA:

Die Aufzeichnungsgeschwindigkeit beträgt 1500 bit/s. Das bedeutet, daß bei einer Bandgeschwindigkeit von 4,76 cm/s auf 1 mm Bandmaterial etwa 4 Byte (genau 31,5 Bit) Platz finden. Anders gesagt, auf einer zweiseitig bespielten C-60-Kassette kann man etwa 640 KByte an Daten bzw. Programm unterbringen. Wenn man außerdem in Betracht zieht, daß bei diesem Verfahren ein 16 KByte langes Programm in nur knapp 90 s geladen wird, so wird doch deutlich, daß dies ein ökonomisches und auch einigermaßen schnelles Mittel zur Programmspeicherung ist.

Das Prinzip des hier angewandten Aufzeichnungsverfahrens ist in Bild 32 am Beispiel des zur Synchronisation verwendeten Bitmusters dargestellt. Für die Aufzeichnung eines Bit wird eine Periode einer 1,5-kHz-Rechteckschwingung abgelegt. Mit dem hier verwendeten Bitmuster 0E6H bzw. dessen Negation 19H werden dann die Aufzeichnungen beim Einlesen wieder auf den Byteanfang synchronisiert. Das negierte Synchronisationsmuster muß mit ausgewertet werden, da man nicht einfach voraussetzen kann, daß das aufgezeichnete Signal in der gleichen Phasenlage vom Band kommt, in der es abgespeichert wurde. Schon gar nicht, wenn man auch noch von einem anderen Gerät einlesen will, mit dem die Aufnahme nicht gemacht wurde. Und genau das dürfte ja beim Programmaustausch der Regelfall sein.

Ebenfalls zur Anpassung an die Gegebenheiten der Aufzeichnungsgeräte ist eine zweite Ergänzung notwendig, die mit dem eigentlichen Aufzeichnungsverfahren nichts zu tun hat. Geräten mit einer Aussteuerungsautomatik muß man nämlich etwas Zeit geben, damit diese sich einpegeln können. Zu diesem Zweck werden dem Synchronisationsbyte 512 Nullbytes vorangefügt. Das heißt, für etwa 3 s ist ein 1,5-kHz-Ton zu hören, und dann beginnt erst die eigentliche Information. Dieser charakteristische Ton erleichtert dann gleichzeitig das Auffinden der einzelnen Programmanfänge. Beim Einlesen wird am besten innerhalb dieses Kenntones das Ladeprogramm mit der CR-Taste aktiviert.

Nun muß auch noch etwas für ein Kriterium getan werden, das angibt, ob die eingelesenen Daten richtig sind, da natürlich Störimpulse oder Fehler im Band Fehlauswirkungen zur Folge haben können.

Dazu gelangt die Information in Blöcken zu 256 Byte mit einer Prüfsumme je Block auf das Band. Während des Einlesens wird diese Prüfsumme dann wieder aus den gelesenen Bytes berechnet und mit der am Blockende aufgezeichneten verglichen. Sollten beide nicht übereinstimmen, so erfolgen eine Fehleranzeige und Abbruch des Ladeprogramms. Es muß dann erneut ausgeführt werden.

Zur Kontrolle der ordnungsgemäßen Funktion des Einlesens blinkt auf dem Bildschirm hinter dem gelesenen Programmnamen im Blockrhythmus ein Stern. Dieser bis hier beschriebene Aufzeichnungs- bzw. Leseprozeß wird mit Software realisiert. An Hardware ist nur noch ein minimaler Aufwand erforderlich, der die Impulse vom Rechner in Pegel und Form anpaßt, und der die Information vom Band wieder zu einer Rechteckfolge regeneriert. Die dazu erforderliche Schaltung ist im Bild 31 dargestellt. Sie findet auf einer 40 x 70 mm großen Leiterplatte Platz. Die Leiterzugführung und der Bestückungsplan sind in den Bildern 33 und 34 zu sehen (diese Bilder werden im Teil 10 veröffentlicht). Der Aufbau dieser Platine ist unproblematisch. Der OPV im Wiedergabeteil arbeitet als Nulldurchgangskomparator. D1 und R5 schützen den PIO vor negativen Überspannungen, bzw. bei versehentlicher falscher Programmierung den entsprechenden Ausgangstreiber vor der Zerstörung. Bei der Verwendung des B 176 D ist R3 mit zu bestücken. Dieser Schaltungsteil wurde mit beiden angegebenen OPV-Typen erprobt. Die Funktion der Schaltung läßt sich leicht überprüfen, wenn man z.B. den Kennton vom Band an den Eingang anlegt. Das Eingangssignal sollte man dem Lautsprecherausgang des Geräts entnehmen, um eine einwandfreie Begrenzung des Signals zu ermöglichen, da der Pegel des Diodenausgangs bei den meisten Geräten zu gering ist. Das Netzwerk im Aufnahmezweig dient zur Reduzierung des Pegels, um das Kassettenbandgerät nicht zu übersteuern, und zur Unterdrückung der Oberwellen, die mit der Löschfrequenz Interferenzen bilden könnten. Mit R7 stellt man den Ausgangspegel so ein, daß die Aussteuerungsautomatik noch nicht begrenzt. Das sind meist 10 bis 20 mV.

Dieses Kassetteninterface wurde mit mehreren „AC1“ und mit verschiedenen Kassetten- und Magnetbandgeräten unter Verwendung normalen, handelsüblichen Bandmaterials getestet. Der Aufbau der Interfaceschaltung war dabei immer problemlos. Als Kassettenbandgeräte wurden z.B. „Mira“ und „Sonett“ verwendet, auch der Programmaustausch mit verschiedenen Geräten wurde simuliert. Einlesefehler traten so gut wie gar nicht auf, bzw. waren auf fehlerhafte Geräteeinstellung oder zu große Abweichungen in der Tonkopfsteilung zurückzuführen. Bei Geräten mit Tonblende bzw. Höhenregler muß dieser voll aufgedreht werden. Bei Fehlern infolge zu großer Tonkopfabweichungen genügt eine Einstellung des Tonkopfes auf helle Wiedergabe nach Gehör. Selbst relativ extreme Geschwindigkeitsschwankungen, die z.B. die Wiedergabe von Musikaufzeichnungen zur Qual machen würden, haben nicht zu Einlesefehlern geführt. Auch leichte Pegelschwankungen infolge des Bandmaterials konnten durch die Begrenzung des OPV ausgeglichen werden. Gegen vollständige „Drop out's“ (magnetische Fehlstellen) ist natürlich auch dieses Verfahren empfindlich, wobei diese in der Testphase bis jetzt nicht auftraten. Um einigermaßen sicher zu gehen, empfiehlt es sich, jede Aufzeichnung zweimal zu machen oder die gemachte Aufzeichnung mit einem Kontrolleseprogramm zu überprüfen. Ein solches Programm wird für den Monitor des „AC1“ noch erstellt werden.

Zusammenfassend kann gesagt werden, daß dieses Verfahren mit der gewählten Geschwindigkeit wohl einen guten Kompromiß zwischen Schnelligkeit und sicherem Programmaustausch bei minimalen Anforderungen an das Aufzeichnungsgerät darstellt. Bei der Verwendung hochwertiger Geräte sind sicher noch weitaus höhere Bitraten mit der gleichen Sicherheit möglich. Auch ist eine komfortablere Software denkbar, die dann z.B. Namenerkennung und wählbare Geschwindigkeiten gestattet, aber dafür auch weitaus mehr Speicherplatz benötigt und somit erst bei einer Speichererweiterung sinnvoll ist.

[10] Krake, H.: ZETBUG - ein komfortabler Z-80-Monitor, Funkschau 52 (1980), H.11

[11] Klein, R. D.: Basic Interpreter, Franzis Verlag, München
[12] Selenko, G.; Panov, W.; Popov, S.: Mikroprozessoren und Mikrorechner für den Funkamateur, RADIO, Heft 9/1983, Seite 32 bis 35

Turbo-Tape

Von F.Ludwig (SCCH) stammt ein Turbo-Tape, dass das Aufzeichnungsverfahren des Poly-Computers und des Z1013 nutzt, aber mit abweichenden Zeitkonstanten.

Im SCCH-Monitor V8 ist Turbo-Tape neben dem originalen Aufzeichnungsverfahren vorhanden; im SCCH-Monitor 1088 findet sich nur noch Turbo-Tape.

Die Übertragungsrate beträgt ca. 4000 Bit/s.

Im wesentlichen gelten die Ausführungen von Z1013-Kassettenformate. Es gibt einen Kopfblock (Header), der nicht zum Z1013-Headersave kompatibel ist, sondern einen eigenen Aufbau besitzt. Dieser Kopfblock und der erste nachfolgende Datenblock haben einen langen Vorton, die anderen Datenblöcke einen kurzen Vorton.

Datei:

<ditaa noedgesep> +——-+——-+——-+——-+——-+——-+ +——-+

vorton	Header	vorton	Block	Block	Block	...	Block

+——-+——-+——-+——-+——-+——-+ +——-+ </ditaa>

Block:

<ditaa noedgesep> +——+——-+—-+——-+—-+

0000	word0	...	word15	cks

+——+——-+—-+——-+—-+ </ditaa>

Kopfblock:

Im Kopfblock ist Dateiname (16 Zeichen), Anfangs- und End-Adresse und der Dateityp (1 Zeichen) enthalten. Eine Startadresse gibt es nicht!

<ditaa noedgesep> +—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
'N'	'N'	'N'	'N'	'N'	'N'	'N'	'N'	Dateiname, mit Leerzeichen ..							

+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+

10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
.. aufgefüllt (16 Zeichen)	':'	' '	typ	eadr	aadr	'L'									

+————————————————————————————-+ </ditaa>

Aufbau des Kopfblocks:

Byte	**Bedeutung**
0-7	Turbo-Kennung (8x D'N')
8-23	Dateiname, mit Leerzeichen aufgefüllt
24,25	': '
26	Dateityp (z.B. „P“: Ausführbares Maschinencodeprogramm)
27,28	Endadresse
29,30	Endadresse
31	'L' (für Ludwig ?)

Es werden 4 Dateitypen unterschieden:

- P Lauffähiges Maschinenprogramm
- D Maschinendatei
- B Basicprogramm
- F Datenfeld (BASIC)

Die Filetypen B und F können nur mit BASIC - Interpreter V.3.2 geladen werden. Im Monitor erfolgt lediglich Anzeige des Namens, Filetyp, Anfangs- und Endadresse.

Im Kopfblock gibt es keine Startadresse. Nach dem Laden überprüft der Computer aber das Programm auf eine **Autostart-Sequenz**. Ist diese vorhanden, erfolgt der Autostart auf entsprechender Adresse. Die Autostartsequenz muss an der Anfangsadresse stehen (5 Byte):

```
53 43 48 xx xx ; xx xx = Startadresse.
```

Autostart-Tape

Speichern eines Programms auf Kassette (mit optionalem Startbild) und selbstständigem Start nach dem Einlesen, B.Nickel, 1988

Es wird nicht nur der reine Speicherbereich auf Kassette gespeichert. Das Aufzeichnungsformat ist ein sog. Container-Format, wo man ein Startbild einbinden und einen Autostart aktivieren kann. Der Autostart war ein effektiver Kassetten-Kopierschutz (nach 5x umkopieren ging nichts mehr) und wurde auch so genutzt.

!TODO!

Super-Tape

E.Ludwig

!TODO!

USB .Z80-Dateien

Von Ralph Hänsel stammt Software zur Nutzung eines VDIP-Moduls + USB-Stick. Hier werden Dateien im Z1013-.Z80-Format abgelegt. Auch der JKCEMU nutzt dieses Format zum Speichern und Laden von AC1-Programmen.

Es gibt einen Kopfblock (Header), der zum Z1013-Headersave kompatibel ist.

<ditaa noedgesep> +—–+———-+

Header	Programm

+—–+———-+ </ditaa>

Kopfblock:

Im Kopfblock ist Dateiname (16 Zeichen), Anfangs- und End-Adresse und der Dateityp (1 Zeichen) enthalten.

<ditaa noedgesep> +—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
aadr	eadr	sadr	frei/Autor/CRC	Typ	D3	D3	D3								

+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+—-+

10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
Programmname, mit Leerzeichen aufgefuellt															

+———————————————————————-+ </ditaa>

der freie Bereich ist mit „AC1RWTP“ „MO11.0“, „DVU21 “ o.ä. gefüllt.

Es werden 4 Dateitypen unterschieden:

- P Lauffähiges Maschinenprogramm (z1013: C !!)
- D Maschinendatei
- B Basicprogramm
- F Datenfeld (BASIC)

From:
https://hc-ddr.hucki.net/wiki/ - **Homecomputer DDR**

Permanent link:
https://hc-ddr.hucki.net/wiki/doku.php/homecomputer/ac1/kassetteninterface?rev=1675322630

Last update: **2023/02/02 07:23**