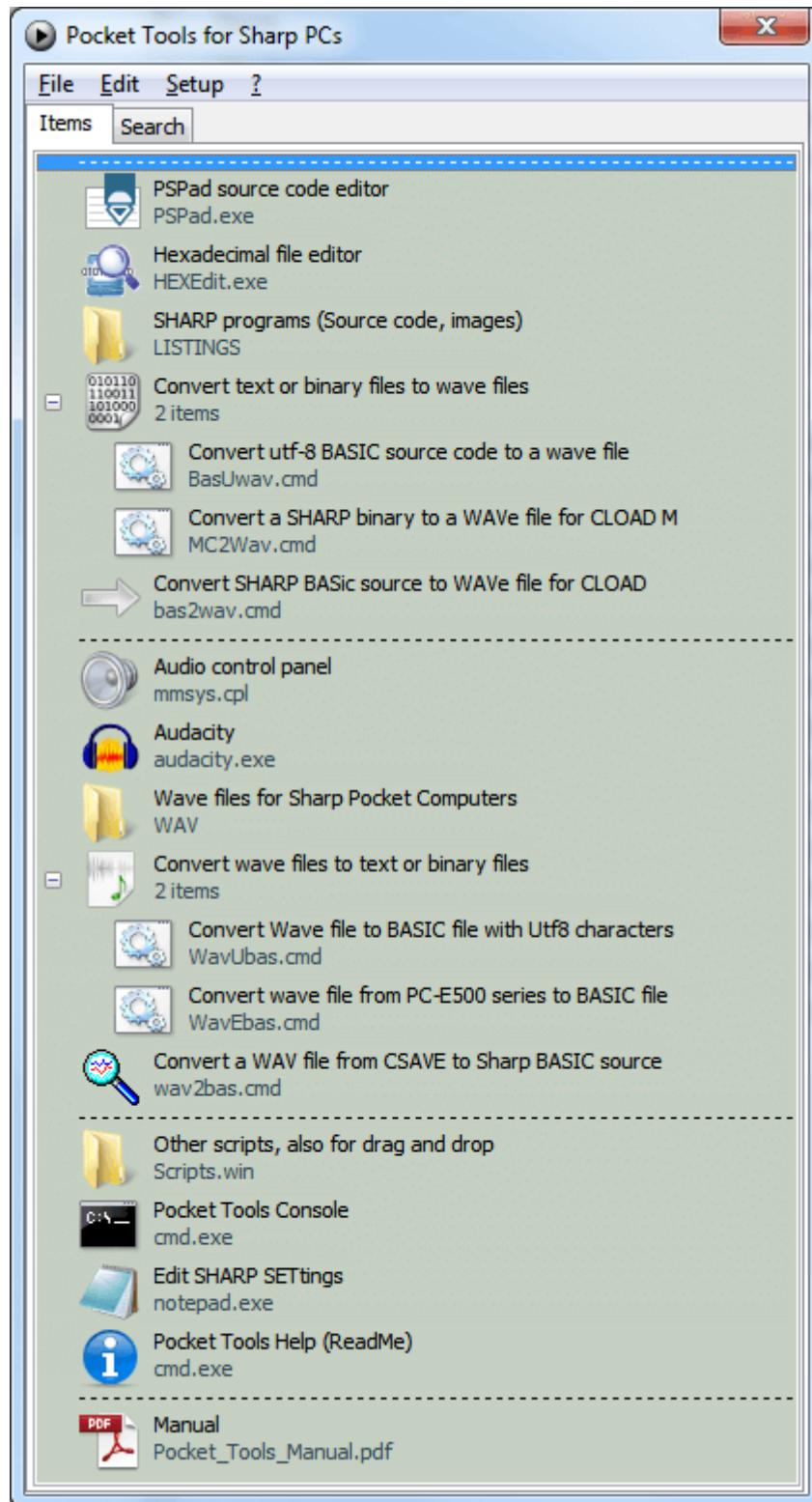


Pocket Tools 2.1 for Sharp Pocket Computer



User Manual Revision 08a4
for Pre-release 2.09c2c2

04.01.2020

1.	Introduction	3
1.1.	Why do you need the Pocket Tools or why not?	3
1.2.	What do I need so I can use the Pocket Tools?	4
1.2.1	Hardware	4
1.2.2	Software	4
2.	Basic Description	4
2.1.	Wav2bin 2.1	4
2.2.	Bas2img 6.1	4
2.3.	Bin2wav 2.1	4
3.	Recording and playback	5
3.1.	How to record the wave file from a Pocket Computer?	5
3.2.	Recording guide for Audacity	5
3.2.1	What should the recorded wave file look like?	6
3.2.2	What should the recorded wave file not look like?	7
3.3.	How to prevent processing errors with Wav2bin from a wave file?	8
3.3.1	Recommended procedure for digitizing tape recordings	9
3.4.	How to transfer programs to a Pocket Computer using the tape interface?	9
4.	Feature list of Pocket Tools	11
4.1.	Technical limitations or not implemented	11
4.2.	Limitations on reading and playing wave files	12
5.	Handled file types	13
6.	Command line options	17
6.1.	Bin2wav	17
6.1.1	Bin2wav generateable sample frequencies	18
6.2.	Wav2bin	19
6.3.	Bas2img	21
6.4.	Understanding the command-line options	22
6.4.1	Pc=Number	22
6.4.2	Type=Type	22
6.4.3	Device=Type	23
6.4.4	Command Line Options for text formatting and others	23
7.	Old Parameter format	24
8.	Error and other codes, returned to OS	25
9.	Supported SHARP formats, cassette files and commands	26
10.	Installation and use, start menu, scripts, operating systems	28

10.1.	Installation for Windows	28
10.2.	Installation for other operating systems	29
10.3.	Use with Windows	29
11.	Use of the Tools, chain and examples	30
12.	Default start addresses (for load) of binary code	33
13.	Supported and tested Sharp Pocket Computers	33
13.1.	Naming convention for Sharp Pocket Computers with special interface	34
14.	Description for some special characters and abbreviations	34
14.1.	Important abbreviations for the source code with Bas2bin	34
14.2.	Special characters, partially across generations	35
14.3.	Processing Flow of special Chars	35
15.	How To	36
15.1.	What type of conversion of special chars should be used with Wav2bin type BAS	36
15.2.	How should you type in the Basic source code?	36
15.3.	Use file names with spaces and special chars	37
15.4.	Best method to transfer BASIC source files to PC-E500 series	37
15.5.	Convert PC-E500 to BASIC source files with Wav2bin	37
15.6.	Transfer of assembler and C sources from the PC-G850V(S)	38
15.7.	Handle wave files from BASIC images, containing binary code	38
15.8.	Sample: Edit BASIC variables with "Transfile PC", transfer to emulator on Android	38
15.9.	Use the very fast SuperTape format for PC-1500	39
15.10.	Use the Quick-Tape format	39
15.11.	Steps to backup and archive a BASIC program	40
16.	Changes in version 2.1 compared to version 2.0	40

1. Introduction

The Pocket tools are a set of programs with which you can convert between audio records of SHARP Pocket computers, and binary files and text files of a personal computer.

You can edit these files on your PC with any editor and convert them back to audio files.

1.1. Why do you need the Pocket Tools or why not?

If you have an adapter cable for each of your SHARP pocket computer that you can connect to a port of your PC, you have the right driver for this interface installed on your operating system (and on the pocket computer, if necessary), and if the data transmission with the (cable-specific) interface settings and your software for the required file types with that operating system works, then you do not need the pocket tools. Especially for the series E500 and newer series a suitable serial cable should be the most comfortable solution.

If you are not sure whether everything still works now, or want the data transmission using your SHARP pocket computer still to work for many years, then we recommend that you learn about Pocket Tools.

1.2. What do I need so I can use the Pocket Tools?

1.2.1 Hardware

You need any sound adaptor or sound card with a microphone input and headphone output.

The adaptor of your personal computer should support microphone pre-amplification, and the cables to the audio terminal should not run next to lines with high current. For insensitive microphone inputs an additional preamp is needed.

For the pocket computer you need an audio interface, usually a cassette interface, that suits your pocket computer. In general interfaces work also without batteries or low batteries. All cables, interfaces or printers that contain batteries must be charged several times a year to avoid damage!

PC-1500: CE-150 or CE-162E

PC-1600: CE-1600P (for mode 1 see PC-1500)

PC-121x: CE-121 or CE-122

All other pocket computers:

CE-124 (except G850 series), CE-126P or another suitable 11-pin commercial cassette interface, also a self-built interface (for example according to the circuit diagram by M. Nosswitz or a PC-G850VS cassette interface)

To backup a BASIC program from PC-1245 up to PC-1475 (referred to here as PC-1234) you need only a microphone and a very quiet room, but this is tricky and not recommended.

If you use an emulator for a SHARP pocket computer, then you need no interface.

1.2.2 Software

You need a recording software or an audio editor with recording function of wave files for input.

Audacity is recommended. You may use the media player of your system for output.

Note both the playback volume of the system (master) and the volume control of the player.

No driver is needed.

You need a separate text editor and possibly a hex editor. PSPad is recommended.

2. Basic Description

2.1. Wav2bin 2.1

With Wav2bin you can convert the content of a wav file containing the sound of a SHARP pocket computer program to a binary file, to be used on your Personal Computer or a source text file.

The resulting source texts can be edited with a text editor and binary files with a hex editor.

Please see chapter 5 for more supported file types.

2.2. Bas2img 6.1

With Bas2img you can convert a source text file containing SHARP BASIC language to

- A) a binary image file (IMG) with intermediate code,
- B) a binary image (TXT) without intermediate code but with binary line numbers or
- C) an ASCII file (ASC) with pocket-specific format.

2.3. Bin2wav 2.1

With Bin2wav you can convert the content of a pocket-specific binary image file of your Personal Computer utilizing the internal format of a SHARP pocket computer to a wav audio file, which you can play back with any media player.

Initially, set both the system master volume and the media player volume to 75%.

Then modify both values simultaneously to find the limits of the range within which the pocket computer can do a CLOAD of the program from the sound card correctly.

For both volume controllers, select a value slightly above the mean of this range, and note down each combination of settings for this PC and audio hardware, see also 3.4.

3. Recording and playback

3.1. How to record the wave file from a Pocket Computer?

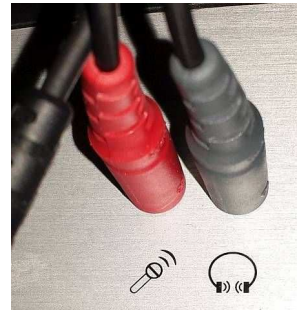
Record PC-1500 with a minimum sample rate of 11025 Hz, others with 22050 Hz.
Record PC-1500 Quick-Tape with a sample rate of 48000 Hz.

For sound from modified Pocket Computers with a hardware speedup switched on: use 48000 Hz or more (and later use the wav2bin -cspeed parameter +-5% accurate).

You can record high levels and clear signals with 8-bit, otherwise: Record 16-bit!
Record monophonic. If the software does not support this, then record stereo.

In general, the microphone input (pink) should be used. Pre-amplification should be switched on for the microphone in the sound card, especially for PC-1500, PC-1600 and PC-121x.

Where there is a choice between Mic and Line In, the LINE INput (blue) should only be used for direct recordings from cassette recorder. Unplug and replug the microphone plug (usually red) in the sound card to select the correct port. Disable all mixers and filters. Especially with extension cables, check that the plugs have not been confused. It could still record a signal, but the quality is not sufficient.



Record at a higher level, but not above 95%. An input level of 5% or less is too weak. You should then use a mic preamp, but you can still try to record borderline signals at a sample rate of 48kHz.

First start CSAVE or the cassette recorder, wait up to one second for a stable audible sound, then record. Sound from PC-1600 and later models starts after a silent time. The preceding silent period can be omitted, however the intermediate periods and tenth of a second after the end of transfer must be recorded. Some data blocks are split by more silent periods. Don't stop recording too fast! If the silence continues more than 9 sec, then it is safe to stop.

In earlier versions of Wav2bin you had to convert to 8-bit, mono and low frequency.
With this version all this is no longer required. Normalizing is recommended but not mandatory.

3.2. Recording guide for Audacity

The following guide of "Edgar Pühringer" and "Norbert Roll" for "Audacity" is adapted to the current tools version, new supported SHARP formats and software version now.

1. System settings

On a Mac, you may have to set the sound input to "Line In" using the utility "LineIn". Normally set the recording device to the microphone input.

On Windows right click the loudspeaker symbol in the task bar for the selection of the device.

Switch microphone pre-amplification on in the preferences of your microphone-device (because of the quality a bit less than the highest gain, if you can adjust it).

2. Start Audacity. Set Edit->Preferences as follows:

Devices (I/O) -> Channels = Mono (1)

Quality-> Standard Sample Frequency for most recordings = 22.050 Hz

PC-1500 standard format : = 11.025 Hz or better

Quick-Tape or Hardware Speedup used : = 48.000 Hz or better

SuperTape (not supported by Wav2bin) : = 44.100 Hz or better

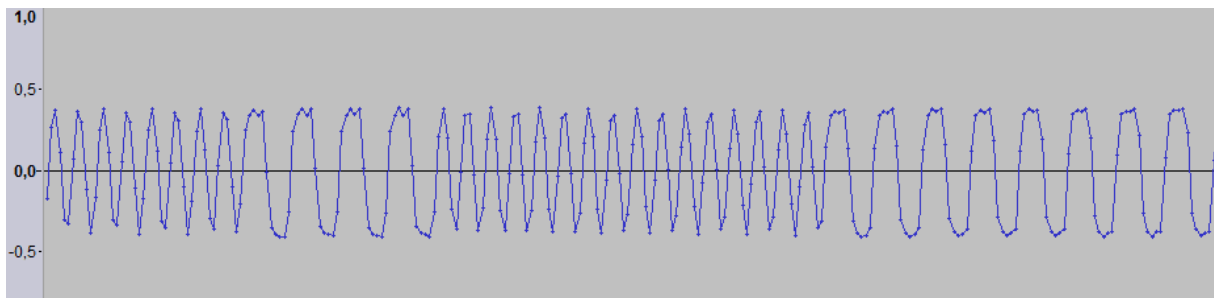
Quality-> Standard Sample Format = 16-bit (or better)

3. Set the Input Volume from 85% (older series) to 95% (PC-1500 and newest series) for most sound hardware first. If the resulting recording will be overdriven (between +/-100%), then the level should best be reduced to +/-70%.
4. Start the SHARP pocket (C)SAVE / PRINT# and wait for it to produce sound. E2- and G-series may not produce any sound. For the other: Wait up to 1 second and start a recording in Audacity (avoid preceding silence).
5. After the Sharp has stopped producing sound, stop the recording in Audacity. However about 10 milliseconds (minimum) should remain after the last sound. PC-16/E/G-series include one or more silent blocks between and after the data: PC-1600 up to 9 sec and PC-E/G-series up to 6 sec silence. If in doubt, wait 10 seconds after the last sound was sent. Normally it should not be trimmed and never resampled! It is best if you normalize the recording: Effect->Normalize (remove DC,-1.0 dB).
6. File->Export the sound recording as WAV 16-bit PCM.

Wav2bin applies its own preamp with a filtering rule and then converts the PCM format to 8-bit internally.

3.2.1 What should the recorded wave file look like?

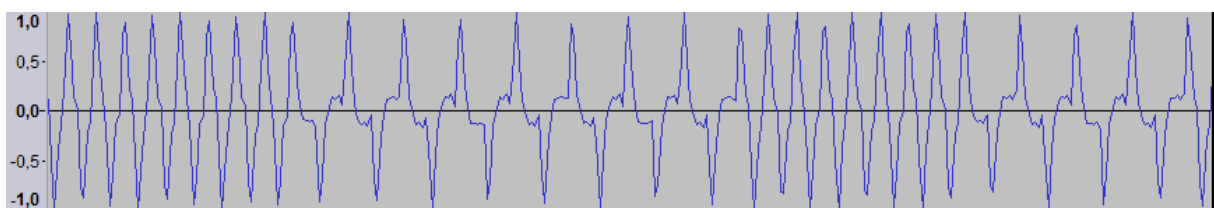
Part of a wave file (PCs from PC-1210 to PC-1475)



The number of wave cycles must be a multiple of 8 for the 4.0 kHz signal (for Bit 1, sync bits or stop bit(s)) and a multiple of 4 for the 2.0 kHz signal (Bit 0 or the start bit of a quater).

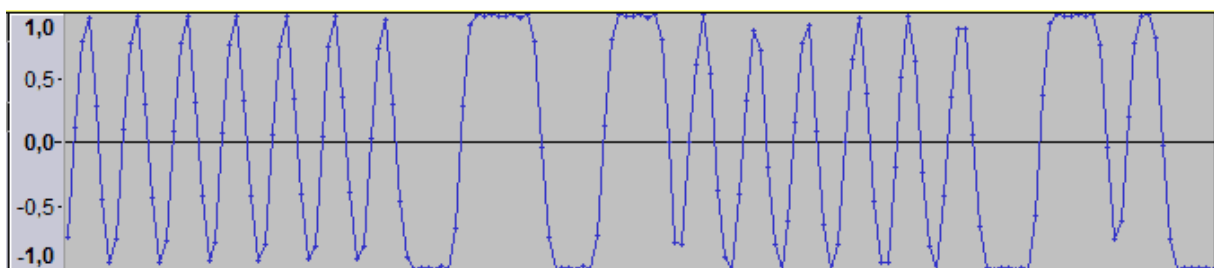
For more information, please see Heise c't 5/88, page 116 "Draht zum großen Bruder" and "Ergänzungen und Berichtigungen" in c't 5/88, page 116 or see the article by Norbert Unterberg "SHARP <--> PC" chapter 5.2.

Part of a normalized wave file of the PC-1500, CE-150/162E directly connected



The number of wave cycles must be a multiple of 8 for the 2.5 kHz signal (for Bit 1, sync bits or stop bits) a multiple of 4 for the 1.25 kHz signal (Bit 0 or the start bit of a quater). For more information, please see the Technical Reference Manual of PC-1600 ch 3.11.2, page 122-124.

Part of a wave file of the PC-1600 with CE-1600P or PC-E- and -G-series



There are only allowed waves of 3.0 kHz signal (for Bit 0, sync bits) and waves of the 1.2 kHz signal (Bit 1 or the start bit of a byte, less sync bits). For more information, please see the Technical Reference Manual (TRM) of PC-1600 chapter 3.11.1, page 117-121 (IOCS) and the TRM of PC-E500 chapter 3 no.04 cas: page 64-66.

Avoid any other activities at your computer while recording!

Your sound system must always provide a clear signal with no dropouts, clicks or hum during the entire recording.

If you cannot resolve this, then you have to use another sound adapter or computer for the recordings. Otherwise Wav2bin can not fully convert and reports the errors of your wave file.

3.2.2 What should the recorded wave file not look like?

Coarser errors (e.g. gaps) cannot be resolved by wav2bin.

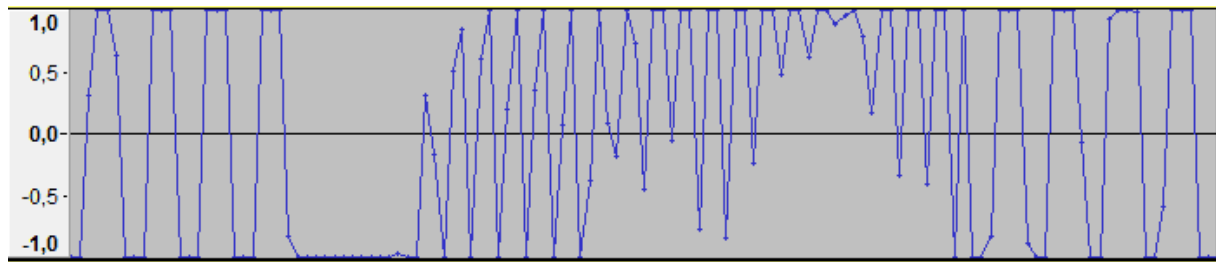
Unfortunately, it sometimes happens that through system activities e.g. 10 ms of the recording have been cut out or replaced by dummy data (last picture). Do not waste too much time debugging, but try a new recording first.

It is often more difficult to find the wrong spots than to correct them. Wav2bin can help you to find it.

Shown here are some typical errors.

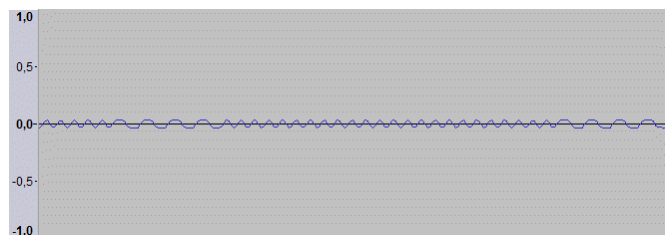
The first picture is a tape recording. Here you have to pull up the missing 4 waves in the gap in the front part with an audio editor and pull down at least one amplitude on the right.

Dropouts (PC-1245-1475)



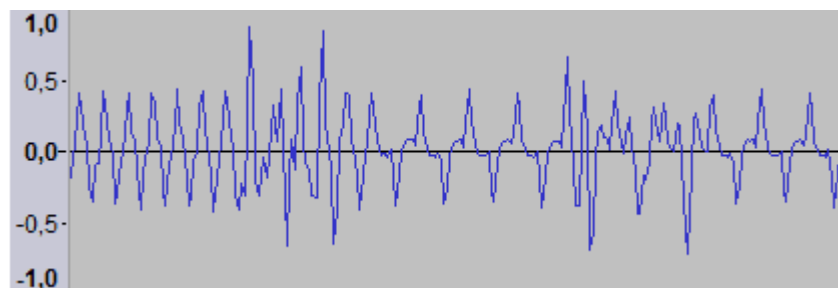
Cause: Tape dropouts and less DC compensation of the interface with the tape recorder output

Low Level (PC-1500): The signal is very sensitive to disturbances.



Cause: The microphone input is not sensitive enough or the gain of the preamplifier is too low.

Disturbances normalized, PC-1500 (PC-1211) Normalizing amplifies disturbances also.

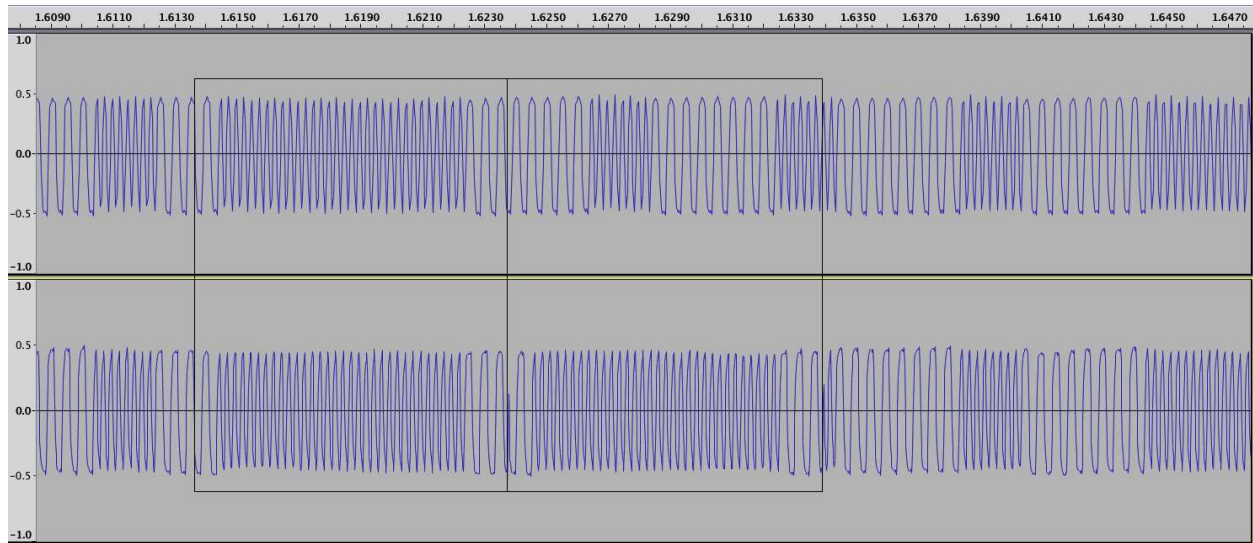


Cause: Induction from other cable to the audio cable (internal power supply cable)

Some of the following problems are caused by energy-saving modes of the computer.
The screensaver should not be active during long audio recordings.

Dropped waves can be detected by spikes in the curve or incorrect wavenumbers.

Repeated samples – a copy of a part of the waveform overwrites a later part (PC-1403)



Upper trace: correct recording

Lower trace: disturbed recording

Cause: Sound recording software, operating system activities or energy saving states, sound hardware

3.3. How to prevent processing errors with Wav2bin from a wave file?

If the level from mic input is only less than 5% during recording, then you must use a good mic preamps. Behringer MIC100, connected with ProCab CAB714(S) to SHARP CE, has been successfully tested. At the output of the amplifier is still an adapter DAP XGA-13 (or equivalent) or a suitable cable needed.

In case of errors you should use batteries and avoid the use of an old power supply (mains hum) for CE-126P or other interface. Try to unplug any power supply first, and then connect the CE. Avoid a long audio cable. In case of side effects, connect only the microphone plug, not the headphone. Signal monitoring, cables and low levels may generate uncorrectable bit errors.

When digitizing from cassette, use a cassette deck with a flutter < 3%.

Preceding speech or other sound can preclude the detection of base frequency (no sync found). The same problem arises if the preceding silent period of PC-16/E/G-series is overlaid with a disturbing frequency. In this case you have to cut it, but normally don't cut the lead in! If this is necessary, then cut directly on a sample point to avoid resampling.

Before you normalize very low sound, cut louder noise following the transfer 10 ms or more after the signal. Normalization and wave editing was tested with SoX and Audacity.

Do not convert the sample rate! If you have to do that for other reasons, then use a recording with a sample rate of 48 kHz or higher.

Try to use the device specific filter rules:

1. For sound via system from emulator also set `-device=EMU` and `cspeed`. For sound recorded via microphone from air use a silent room, avoid near (about 5 cm) reflections (for example a display), set `-device=AIR` and `--level`.
2. For recordings direct from cassette recorder (only with fluctuating DC bias)
`-device=CAS` or `-device=CS` may be helpful.

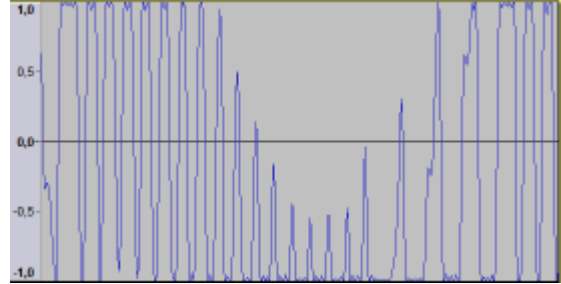
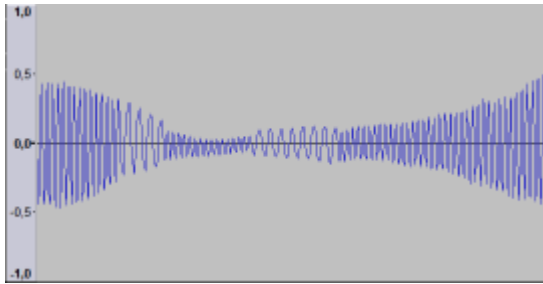
3.3.1 Recommended procedure for digitizing tape recordings

If tape material is faulty, digitizing can be time consuming or it can become impossible to completely recover the data. Always use Wav2bin with the **pc parameter** and the parameters described below. If an error occurs, go to the next step. The following order is recommended.

1. Use the pc parameter but not any device-specific device parameters
2. Record the file again at the standard frequency of your system, often 48 kHz and repeat (1), that is, with the default settings for the selected pc (without -d).
3. In the audio editor, check the places where the transmission was aborted.
Depending on the error, use:

for left picture: --device=CS (symmetric)

for right: --device=CAS (asymmetrical)



without specifying a level (subsequent number), i. with type-specific default values.

4. Use the parameters =CS0 to CS4 or =CAS0 to CAS4 one after the other
5. Use a hi-fi tape deck instead of a tape recorder as the player
6. If you get constant errors from a tape, use --level=0xC0, then 0x400 to get the wave time of the error, and fix the amplitudes with an audio editor. With 0x800 you convert it with checksum errors. Or you create an unchecked tap file.

If Wav2bin still can't find any synchronisation, then open the wav file in the audio editor, select Analyze->Frequency analysis (FFT), Spectrum and navigate to the highest peak.

This is the base frequency, typically for computers with original clock frequency

PC-1500	2500 Hz,	SuperTape PC-1500: 3600/1250 Hz (not supported by wav2bin)
PC-1600/E/G series	3000 Hz,	Quicktape PC-1500: about 5000 Hz
All others PCs	4000 Hz	

If the frequency deviates by more than 5% of it, then something is wrong and the corresponding factor must be passed to Wav2bin with the parameter "cspeed" = (measured/ typically base frequency).

If there are fluctuations and drop-outs in the lead-in that prevent synchronization, it will be necessary to remove them to a sample point (with very high accuracy), with approx. 0.5 s remaining.

3.4. How to transfer programs to a Pocket Computer using the tape interface?

If the WAV file created by Bin2wav was zipped, unzip it (for example, use 7-zip).

Do not connect the original small black plug (Remote) to the sound card!

1. Connect the EAR input of your SHARP cassette interface or with the grey plug (usually) preferably to the HEADPHONE output of your computer, or alternatively to the LINE OUTput (green) of the sound card.
2. Enter the appropriate (C)LOAD / INPUT# command on your pocket computer and wait 2 seconds (for some of PC-E/G series, until remote relays is switched on). For the first test use a very short, e.g. the one-line program: 10 PRINT „HELLO, WORLD“: END
3. With the following settings, play back the WAV file with a media player on the PC.
Turn off other sound sources such as system, browser and mic input in your sound system.
For testing or problems, disconnect the microphone plug.

You can start your tests with a volume of 75% (**Master and Player**), and adjust both

simultaneously in 5% increments upwards or, if necessary, downwards.

Especially PC-E500S and PC-G850 series may need a higher volume: Both controls 85% to 95%. In general, it is not recommended to set the master player to 100% and the media player to 50% alternatively, because this can limit the error-free range.

4. For pocket computers with built-in beeper, it must sound when CLOAD / CSAVE is active. Except for the oldest series, the display changes after a file is detected.
5. After performing a short transfer in the previous step, optimize the settings by transferring a longer BASIC program. Make note of the setting values of both volume controls after successful transmission.
6. Some sound hardware requires other sample rates for an error-free playback. If you were unsuccessful, test everything with the options -l2 or -l1 (lowercase "L", scripts with the parameters: ? -l2). With the parameter -l2 for PC-1500 a sample frequency of 44.1 kHz and otherwise 48 kHz is used for the generated WAV file.

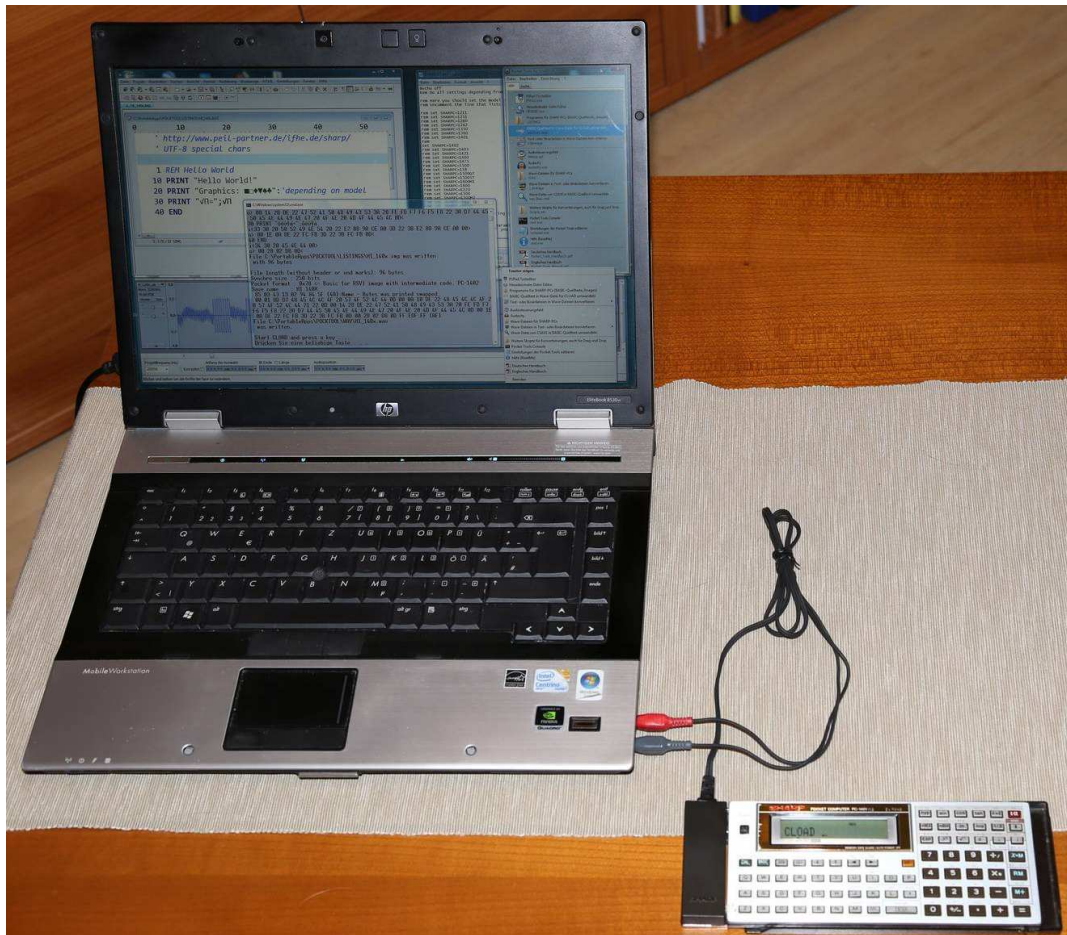
The tested interfaces gave the best result with a waveform between triangle and rectangle, for example sine or trapezoidal. The sound chips deliver the best quality if the sample rate matches their own frequency. You can find the best compromise for your system.

If you use the supplied scripts, you can permanently save the option in the BIN2OPT variable.

7. In the PC-1600 but also other models, especially all open and newer models, the parameters for the cassette transmission can be so adjusted after system errors that CLOAD and in most cases also CSAVE (better recognizable here) no longer delivers meaningful results. It is then recommended to write the existing data on a battery-backed memory card if possible and then to carry out the total reset: [ON] + Reset.

Application example for SHARP PC-1401/02 with CE-124

1. Set for transfer in the file Sharpset.bat: set SHARPC=1402
2. Edit the BASIC source text with PSPad
3. Convert it with the PStart menu: bas2wav.cmd, for transfer first enter on the Pocket Computer CLOAD and then press any key on the personal computer in the terminal window



4. Feature list of Pocket Tools

1. All SHARP pocket computers with a commercial cassette interface are supported:
From PC-1210 up to PC-G850VS and also PC-1100 series
 2. Reads and writes all standard cassette file types of all SHARP pocket computer generations:
BASIC source file or Image, ReSeRve key data, Def Key data, Binary data or machine code,
all multiple variable data, ASCII Data, ASCII Sources, Text Mode and
Assembler and C source code
 3. Reads and writes for PC-1245 (PC-1211) up to PC-1475 all binary files for "Transfile PC plus "
(SHC), also arrays of variable and the BASIC sources (SHA)
 4. Reads and writes merged basic source (line number 99999) and images
 5. Supports inline comments and offline comments (offline ' not transferred, except with --auto) and
different line number formats (aligned, without and with ':')
 6. Supports shortcuts to many commands for faster typing , such as: 'C.'
 7. In addition to the standard BASIC statements, you can also use up to three hardware and
software extension token keyword sets and special character codepoint files.
 8. Conversion of special chars between SHARP generations and operation systems conversion to
upper case letters (if necessary), loadable keywords for extensions,
Processing of undocumented signs and tokens within a generation
 9. Limited conversion of numeric (no double) variable data and text between SHARP generations
 10. Supports all usual wave audio files without any conversion or finishing, writes different wave forms
(apply it hardware dependent)
 11. Supports wave- and tap-files from emulators and wave files from pocket computers with hardware
speedup
 12. Read Quick-Tape formats and writes Quick-Tape 4 format,
writes SuperTape wave files (but does no read directly)
 13. Automatic correction of many disturbances in wave audio files and named filters for recording
from cassette recorder and microphone, all checksums are checked, easy and reliable
transmission
- Note: CE-1600P, E-Series and G-Series is a clear signal quality required for the data contained
with working DC-compensation For data directly from a cassette recorder, you can use one of the
--device=CS or CAS parameters, especially for the older series if required.
14. All Operating Systems with ANSI C Compiler are supported, GCC compiler recommended
 15. Provides return codes (error levels) for inclusion in scripts,
other exit codes (type, ident) and exit after header can be selected with Wav2bin
 16. Wav2bin can write parameters, header values and the real speed to a configuration file, usable
with Bin2wav
 17. Parameters in standard format and old format supported
 18. Continuous debug traces and more special options exist, see the help screens

4.1. Technical limitations or not implemented

1. Bas2img does not support E500 series intermediate code completely.
The numerical constants in the code remain in ASCII code and are not converted into the binary
code. Only the line numbers of known token are converted to binary.
The transferred source code has to be converted to an executable state on PCs of the E500
series by entering "TEXT" and then "BASIC", see "HowTo" Chap. 15.4.

During the run of a BASIC program the PC-E500 series does change the jump targets in the

source file image from line numbers to absolute positions.

Read "HowTo" Chap. 15.5 to get a complete BAS file from a Wave file (or use WavEbas.cmd).

2. Support for MZ series (700) is experimentally with Bin2wav SuperTape only. Bas2img does not support MZ series intermediate code completely. Numbers inside the lines remain ASCII coded. Keyword files are mandatory.
3. Quick-Tape Variable data are supported by Wav2bin, but Qsave-specific information is lost. Use the standard format of PC-1500 with DIM and INPUT # for the retransfer with Bin2wav. QSAVE D and QLOAD D can be used alternatively.
4. SuperTape is supported by Bin2wav, but not by Wav2bin directly. "SuperTape for Win32" is recommended to convert a Wave file to a binary ST file. Wav2bin can convert an image or ST file to a BAS file, see "HowTo".
5. Other Fast/Turbo Tape formats are not supported: for example not TRAMsoft Tool2, FSL, TOS.
6. No stream processing: wav2bin needs the complete wave file.
7. For the pocket computer a conversion by Bin2wav is completely reversible compared to a faultless with Wav2bin, however the result of a conversion by Bas2img may differ from the original template converted with Wav2bin and must be reworked in individual cases - see chapter 6.4.1. "Pc=". This is also to be considered in the backup concept.
8. No graphical user interface, only portable start menu and scripts (with drag and drop) and a dialog box for file selection (command line options, CFG file and return values for an external user interface, however, exist)
9. Japanese characters for PC-1600K and PC-1480/ E500j or higher are supported only with bas2img -l0x100 or codepoint files. Japanese characters for PC-1100-1255 are not supported.

Note: In older versions of the tools, most special characters were only supported in the form [HH] for PC-E220 and the G-series, as well as PC-1500+CE-156/ 1600/ E500.

This is now solved by using codepoint files.

10. There is no hex to bin conversion built in. Additional software can be used to convert an ihx or hex file into a raw BINary file. But Bin2wav reads the addresses from the screen output of Hex2bin 2.5 appended to a cfg file.

4.2. Limitations on reading and playing wave files

1. You must use PC and audio hardware, which is neither crack nor crackling or rustling. The recording level must be high enough to mask minor interferences, otherwise you will need a mic preamp - see chapter 3 "Recording".

Not every inexpensive onboard or USB hardware meets the quality requirements for a data transfer. In this case, you must use different hardware for the Pocket Tools.

2. By default, uncorrectable errors cause the conversion to abort. Corrected errors lead to warnings (text or !) when the debug levels are switched on. A single error can generate multiple warnings if it is detected on different layers of processing.
3. In the bit detection of Wav2bin only "one-dimensional" methods are used. In addition to the absolute level, a differential gain is also applied. With PC-1210 to PC-1500 after the application of amplifiers and filters 1. the number of transitions and 2. the number of amplitudes are evaluated. For the PC-1600, G-Series, E-Series, Quick-Tape and SuperTape format, only the distance between the zero-crossings is evaluated. This format is much more sensitive.

There is no pattern recognition. If the wave file is not readable, then please note the time stamp of the error. Use the debug option -l 0x400. You can clearly see a lot of mistakes and manually correct quickly with an audio editor. "Chopped" wave files, where only a few milliseconds (inside or just after the end) are missing, can not be read.

5. Handled file types

- WAV file** RIFF Wave File, mono or up to four channels (first is used)
sample rate 5 kHz (PC-1500) /8 kHz (PC-1211 to PC-1475) up to 192 kHz
A) from the audio interface of a SHARP pocket computer
B) from a cassette tape --device=CAS
C) from the emulated cassette tape device of "Pockemul"
D) from the system sound of an emulator --device=EMU (PC-1500, 1600 and later)
E) via microphone from the buzzer --device=AIR (PC-1245-1475)
The wave file should be recorded using an audio editor with a sample frequency about 22050 Hz and 16-bit. PC-1500 can be recorded with 11025 Hz. If more editing is required, and for Fast-/Quick-/SuperTape please use 48000 Hz (or 44100 Hz).
The wave files made by Bin2wav are aligned and therefore the frequency can be lower.
A wav file could be normalized and minor errors corrected with an audio editor. Normalizing is recommended but nothing else.
If you want to convert the wav file to 8-bit (unnecessary), then you have to normalize it first! Conversions can be done with SoX. Audacity is the recommended editor.
The wave file must be a "public readable type" (with less chunks).
- tap file** Raw binary format of quaters (PC-1500) or bytes (all other PCs), represents the content of the wave file
This format was introduced by Olivier De Smet for his emulators running on Android devices for SHARP pocket computers.
The ID, the name, all headers, end marks and all checksums are included in this files type.
You should not change the file with a hex editor, because you also have to correct the checksums. But you can try to correct errors from a corrupt wav file after the wav file was converted to a tap file: wav2bin --type=tap
All single bits, synchronisations and spaces are ignored!
It is possible to create a tap file from a wav file but impossible to make a wave file from a tap file.
- SHC file** Binary byte format of the Software "Transfile PC plus" or an OEM version, Copyright Yellow Computing or licensed partners. The very last version 5.55 runs on faster PCs, in Dosbox also, but without transfer, because this needs a Single-Tasking OS and an old style parallel port.
All formats of PC-1245-1475 are supported with --type=shc . The PC-1211 is mapped to the format of the PC-1251 as it is not directly supported by Transfile.
The ID, the name, all headers and the end marks are included in this files type, but no checksums. A shc file can contain images, binary data or variable data.
Transfile also includes a text editor for variable data. You could edit it with a hex editor, but note that most headers and some data bytes consist of swapped quaters.
- IMG file** Binary byte format of Wav2bin --type=img, Bin2wav and Bas2img that represents the internal format of a pocket computer series, the intermediate code of a BASIC program, usually generated with CSAVE.
All formats of all SHARP pocket computers are supported, but using the computers of PC-E series with Bas2img you have to switch to the TEXT Modus and back to BASIC for complete translation after the transfer.
No ID, no name, no file headers, and no checksums are included in this file type. You can save this information in a CFG file. In older versions, sometimes an end marker was

inserted, which is now automatically removed unless the "--endmark" switch is set.

Files of the PC-1600 (subtype **i16**) may contain a file header.

img file
(type **txt**)

Binary byte format of Wav2bin, Bin2wav and Bas2img that represents the internal format of a pocket computer, if it is switched to text mode, usually generated with CSAVE.

There does not exist any intermediate code inline of the BASIC program, but line numbers and the envelope of every line are binary coded. The content of the line is released in the ASCII format.

This format is supported with PC-1360, 1475, all older pocket computers with a serial interface and the PC-E500 series, but it is not a recommended format.

No ID, no name, no file headers and no checksums are included in this files type.

rsv file
(or in **img**)

Binary byte format of Wav2bin --type=rsv and Bin2wav, that represents the internal format of ReSerVe key data of pocket computers, if it is switched to RSV mode, usually generated with CSAVE.

No ID, no name, no file headers, no file end and no checksums are included in this files type.

This format is interchangeable between PC-1500 and PC-1600 or between similar pocket computers of other series. It is embedded in the IMG format for the latter.

def file

Def key image, PC-1500 (used in software, for example in PC-BASIC 84)

dat file

Binary byte format of Wav2bin and Bin2wav --type=dat, that represents the special internal format of data variables of the pocket computers, usually generated with PRINT #.

No ID, no name, no file headers, no file end and no checksums are included in this file type.

The block headers of all data variable blocks and the end mark of any standard variable data block are included in this file type. Multiple data blocks are supported.

All formats of all SHARP pocket computers PC-1211- PC-1600 are supported. Data of PC-1211 are saved in the standard variable format of PC-1251.

The formats of all the different PC generations are limited interchangeable, if possible. Sometime you have to specify the type of conversion with an -l option.

To edit the special data, we recommend not to use a hex editor, but to use the shc format and the third-party software (Transfile PC plus, no double precision).

bin file

Binary byte format of Wav2bin and Bin2wav --type=bin, that represents any internal format of any data of a pocket computer, usually binary code, generated with CSAVE M.

No ID, no name, no file headers, no file end and no checksums are included in this files type. Very limited interchangeable! We recommend to save the addresses in an associated CFG file with the parameter Wav2Bin -m.

If you use this format for the machine code, you must write down the start address of the code (and also the entry address, if supported by your pocket computer), save it in a CFG, or use the Shc format instead.

For PC-E500 note the following regarding the switch -dINV.

For ihx files use Hex2bin to convert into a bin file first.

ihx file
hex file

ASCII data formats for the storage and transmission of binary images, mostly in hexadecimal notation. These formats are not supported directly. Additional software is needed to convert these files into "raw" BIN files as well as onward.

asc file

ASCII data file format of Wav2bin and Bin2wav, that represents the data format of the pocket computer, if the commands OPEN "CAS:data" with PRINT #n was used

This format is supported for PC-E200/PC-G Series and also for PC-1600/E500-series. For the PC-E500-series with level inverting interfaces the signal must be mirrored by

Bin2wav with -dINV, same for some self made interfaces.

Also some of PC-G series can write but not read this format from CAS: correctly.
For use with INPUT #n this is not recommended for G850 series

No ID, no name, no file headers, no fill bytes, no end mark, and no checksums are included in this file type.

The format is similar to bas, but there are differences in the header.

.

BAS file

ASCII source file format of Bas2img, Bin2wav and Wav2bin, that represents the source format of the pocket computer, if it is switched to the Text Editor Cmt menu or the command

SAVE "CAS:filename"(,A with PC-1600) was used.

This format is supported with Bin2wav for PC-E200/PC-G Series and also for the PC-1600/E500-series. For PC-E500-series some interfaces require --device=INV and other not understand it.

The user manual of PC-1600 points to a similar problem at page 6-24 (4).

No ID, no name, no file headers, no fill bytes, no end mark and no checksums are included in this file type. The format is the same as for serial transmission.

This format is editable with a text editor, however we recommend to always use bas2img -t asc before bin2wav (also for PC-1600-G850S, because of the differences between the ASCII formats of last PC series).

ASM file C file

ASCII file format for program source texts in Assembler or C of Bas2img, Bin2wav and Wav2bin, which represents the program format of the Pocket Computer (PC-E200 to G850S), if these are used with the Text editor Cmt menu.

For the PC-E500, 1500, 1600, G850V, G850VS and others the transmission via BASIC is necessary because this menu is not available. G850V (S): see "HowTo" chap. 15.6

A text mode image is used to send to this Pocket Computer. Tokenization is suppressed. For editing under BASIC and possibly the retransmission, the text lines must be provided with a leading apostrophe (asm5) or comment mark (asm6), depending on the PC type and assembler used. Asm8-type files contain marks at the beginning of the line that end with a colon and also mnemonics that must be preceded by a space. For CASL and PIC assembler only the file type "asm" should be used.

The line numbers are not necessary for this file type at the Personal Computer, should be automatically generated for the Pocket Computer by Bas2img and can be removed with Wav2bin.

SHA file

ASCII Text format of the Software "Transfile PC plus" or an OEM version, Copyright Yellow Computing or licensed partners

SHA files can contain various formats. Only the Basic source format (first line with ".BAS") is supported by Bas2img and Wav2bin --type=sha directly, other formats have to be saved with the software as binary shc file for use with Bin2wav.

The type, the name and all header bytes are included in this files first text line of a block, but no checksums. Possibly ASCII end marks are included.

The format is very similar to bas files.

Special chars are encoded for DOS with CP437 only.

ST file Binary image format of the Software "SuperTape" from Heise (c't).
The name, addresses and more informations of the SuperTape header are included in this file. The image follows the header (without end marks or checksums).

This binary file can be written by Supertape for Windows (32 bit) only.
It can be converted to a BAS file or IMG with Wav2bin --img=26.

This file can be converted into a SuperTape wave file with Bin2wav --type=st --pc=ST.

keyword files (CFG) ASCII file format of Bas2img and Wav2bin with a list of keywords and token codes. Each entry must be submitted at a separate line in the following format:

Keyword=HHHH (=16bit hex number)

See also: ReadMe.cfg

This file contains a list of tokens, which supplement or replace the built-in token. If the same token code occurs more than once, then the codes are used in the order of loaded files and only then the built in codes.

This files are required for the commands of some hardware extensions of PC-1500.

This files was introduced by Eric Millescamp, pc-1500.info.

codepoint files (CFG) UTF8 file format of Bas2img and Wav2bin with a list of graphic characters and their codes. Each entry must be entered in a separate line in the following format:

Character=HH (=8bit hex number, Group New also 16bit)

optional: Character=[tag] (only for Bas2img, ignored by Wav2bin)

This file contains a list of UTF8 special characters and their code in the Pocket Computer. If the same token code occurs more than once, the codes are used in the order of the loaded files and then the built-in codes. Variable tags must be placed before fixed codes if variable tags are used and the same file is also used for Wav2bin.

The graphic characters should be sorted in descending order according to their number of bytes.

parameters file (CFG) ASCII file format written from Wav2bin and read by Bin2wav with a list of parameters and values. Each entry must be submitted at a separate line in the following format:

parameter=string (or number)

Parameters written by wav2bin and read by bin2wav:

name, pc (given), pcid (internal used), basefreq & realfreq (calculated cspeed), type (for bin2wav), startaddr (addr), entryaddr (addr), headerflags, endmark

The parameters of the file can be overwritten with command line parameters.

Parameters written by wav2bin only:

ident, typew (wav2bin written), datasize, total, error (last), errors (count), warnings (count)

6. Command line options

It is recommended to use pre-configured scripts with environment variables. Please read ReadMe.txt. or an overlaying software. If you are using graphical software for Windows (32/64-bit) that uses an older version of the tools, you can simply replace the existing executable files with this new version. Otherwise you have to open a **console window** and run the tools in order to pass all parameters.

6.1. Bin2wav

Usage: bin2wav [Options] SrcFile(.typ) [DstFile(.wav/.tap)]

SrcFile	Binary image file (usually created by BAS2IMG or WAV2BIN)																								
DstFile	WAVE file (default: SrcFile.wav) or tap file																								
-t, --type=TYPE	Source file type: <table> <tr><td>img</td><td>BASIC-program binary image (default)</td></tr> <tr><td>bin</td><td>Binary mc of assembly program or data</td></tr> <tr><td>dat</td><td>Data variable blocks (binary data)</td></tr> <tr><td>dim</td><td>Block of all DIMensioned data (Quick tape)</td></tr> <tr><td>rsv</td><td>ReSerVe data (binary image with token)</td></tr> <tr><td>def</td><td>Def key image (PC-1500, used in software)</td></tr> <tr><td>txt</td><td>Text modus (binary image of a program without token, asm)</td></tr> <tr><td>shc</td><td>"Transfile PC" general binary file with header</td></tr> <tr><td>st</td><td>"SuperTape" for Windows binary file with header</td></tr> <tr><td>asc</td><td>ASCII data file, for example, variable data</td></tr> <tr><td>bas</td><td>ASCII source, for example, a BASIC</td></tr> <tr><td>asm</td><td>(asm5, asm6): Assembler or C source text</td></tr> </table>	img	BASIC-program binary image (default)	bin	Binary mc of assembly program or data	dat	Data variable blocks (binary data)	dim	Block of all DIMensioned data (Quick tape)	rsv	ReSerVe data (binary image with token)	def	Def key image (PC-1500, used in software)	txt	Text modus (binary image of a program without token, asm)	shc	"Transfile PC" general binary file with header	st	"SuperTape" for Windows binary file with header	asc	ASCII data file, for example, variable data	bas	ASCII source, for example, a BASIC	asm	(asm5, asm6): Assembler or C source text
img	BASIC-program binary image (default)																								
bin	Binary mc of assembly program or data																								
dat	Data variable blocks (binary data)																								
dim	Block of all DIMensioned data (Quick tape)																								
rsv	ReSerVe data (binary image with token)																								
def	Def key image (PC-1500, used in software)																								
txt	Text modus (binary image of a program without token, asm)																								
shc	"Transfile PC" general binary file with header																								
st	"SuperTape" for Windows binary file with header																								
asc	ASCII data file, for example, variable data																								
bas	ASCII source, for example, a BASIC																								
asm	(asm5, asm6): Assembler or C source text																								
-p, --pc=NUMBER	Model number of SHARP pocket computer, currently available: 1211, 1245, 1251, 1261, 1280, 1350, 1360, 1401, 1402, 1403, 1421, 1450, 1460, 1475, 1600, E500, E220, G850, 1500ST (SuperTape), 1500QT (Quick-Tape) and more (default: 1500)																								
-c, --cspeed=VALUE	Ratio of CPU frequency to original (use it with a modified Pocket Computer with speedup switched on, 0.25 to 2.7)																								
-a, --addr=VALUE	1.use: Start address, needed for BIN type (default: see this manual) 2.use: Entry address 0 to 65535 or 0xFFFF, E500:0xFFFFF (default: no autostart)																								
-s, --sync=VALUE	1. use Synchronisation duration, expressed in seconds, 0.5 to 9 (default: 0.5 or minimum for the PC and waveform) 2. use of this parameter: for spaces (silence) of PC-1600 and later																								
-nNAME, --name=	Sharp file name (7 characters max, 16 for the PC-1500, E:8) transcription for special chars [HH] and for older series _ are replaced - see also "HowTo" (default: DstFile without extension, nor path)																								
--parameters=CfgFile	Read header parameters from CFG file, -m(default: SrcFile.CFG)																								
-q, --quiet	Quiet mode (minimal display output)																								
--tap	Destination file: Emulator tap byte format (not wave file)																								
--version	Display version information																								
--help	Display this information																								
--help=l	show level option screen																								
-d, --device=TYPE	INV interface with inverting level converter (mirrored to zero line)																								
-e, --endmark	Ignore last byte of Img/Txt, Bas/ Asc, if BAS/ASC-EOF mark included																								

Usage: bin2wav [Options] SrcFile(.typ) [DstFile(.wav/.tap)]

-l, --level=VALUE Option bits and Print debug traces (2-times usable)
a (hexadecimal) integer (0x____) or sum of it:

Waveform and frequency (default sample rate is 4-times of base frequency):

- 1 Force low sample rate with triangle waveform for base frequency (old compact format)
- 2 Force wave with 48 kHz (PC-1500 formats: 44.1), near rectangle for sound chips that process other frequencies incompatible
- 3 Force sample rate of 16 kHz - for emulator, 72 kHz for SuperTape

Convert Data variables between series:

- 0x04 Convert PC-1500/1600 numeric data to other PC standard variable, otherwise to numeric array
- 0x08 Data for PC-1500/1600 of length 8 are numeric from other PC
- 0x10 Convert string data between ASCII code and old Basic code
- 0x1000 Use tape format of PC-1475 (slow) for images of the E500 serie, use CLOAD@ for old images
- 0x4000 Write no file header, have to merge data blocks manually
- 0x8000 Data variable block is from Wav2Bin 1.5 or version before
- 0x800 Write also, if checksum bug will be activated (not readable)
- 0x400 Write long synchronisation like the original measure
- 0x200 Write long synchronisation like Tech Ref Man.
- 0x80 Print some global infos more
- 0x40 Print all bytes and (Sum_calculated) - see also Wav2bin
- 0x20 Position and byte list, for data only

For more options - see the source code in PrintHelp.

6.1.1 Bin2wav generateable sample frequencies

Option: level=	1	(0)	2	3
PC-1210 – PC-1475:	8 kHz,	16 kHz ,	48 kHz	
PC-1600, E/G series:	6 kHz,	12 kHz ,	48 kHz,	16 kHz
PC-1500:	5 kHz,	10 kHz ,	44,1 kHz,	16 kHz
Quick-Tape	10 kHz,	20 kHz ,	44,1 kHz,	16 kHz
SuperTape	22.050 Hz,	36 kHz ,	72 kHz	

Your sound system (player software, operating system, driver and SampleRateConverter of the DSP) must be able to generate and reproduce the audio signal for the selected sample frequency without errors.

For systems that are less flexible with regard to the sample frequency and weaker signals, it is recommended to set the option –L2. The wave files generated with this option are larger, but easier to process by the sound system of the personal computer. Unfortunately, due to steeper signal edges, especially at high volumes, they lead to more errors in the audio interface of the pocket computer.

With 44.1 kHz and frequencies derived from it, the original signals cannot be reproduced exactly. However, the signals generated are within the permissible tolerances for the cassette recording.

6.2. Wav2bin

Usage: wav2bin [Options] SrcFile(.wav/.tap) [DstFile(.typ)]

SrcFile	WAVE file (PCM, normalized, recommended is mono 16-bit with a sample rate from 11025 (PC-1500) or 22050 (recommended for other PCs) to 192000 or original wave files made by Bin2Wav, alternative a tap file or image file (also a st- or shc-file).																										
DstFile	Destination file (BASIC-program text or Binary image file)																										
Options																											
-t, --type=TYPE	Destination file type <table><tr><td>bas</td><td>BASIC-program text file, line by line (default)</td></tr><tr><td>asm</td><td>(asm5, asm6, c): Assembler or C source text</td></tr><tr><td>sha</td><td>Transfile PC BASIC-program text file</td></tr><tr><td>shc</td><td>Transfile PC image file (BASIC program, machine code, data)</td></tr><tr><td>img</td><td>BASIC program as a binary image file (interpreter code), line by line with line format check</td></tr><tr><td>bin</td><td>machine code or other binary data,</td></tr><tr><td>dat</td><td>variable data</td></tr><tr><td>dim</td><td>block of all dimensioned data</td></tr><tr><td>imb</td><td>Binary image block of BASIC program with code, not line by line</td></tr><tr><td>rsv</td><td>ReSeRVe mode data (image)</td></tr><tr><td>def</td><td>Def key image (PC-1500, used for example in PC-BASIC 84)</td></tr><tr><td>tap</td><td>Raw byte format (replaces wave files for some emulators)</td></tr><tr><td>raw</td><td>raw binary data for debugging, =rawdat with quaters swapped</td></tr></table>	bas	BASIC-program text file, line by line (default)	asm	(asm5, asm6, c): Assembler or C source text	sha	Transfile PC BASIC-program text file	shc	Transfile PC image file (BASIC program, machine code, data)	img	BASIC program as a binary image file (interpreter code), line by line with line format check	bin	machine code or other binary data,	dat	variable data	dim	block of all dimensioned data	imb	Binary image block of BASIC program with code, not line by line	rsv	ReSeRVe mode data (image)	def	Def key image (PC-1500, used for example in PC-BASIC 84)	tap	Raw byte format (replaces wave files for some emulators)	raw	raw binary data for debugging, =rawdat with quaters swapped
bas	BASIC-program text file, line by line (default)																										
asm	(asm5, asm6, c): Assembler or C source text																										
sha	Transfile PC BASIC-program text file																										
shc	Transfile PC image file (BASIC program, machine code, data)																										
img	BASIC program as a binary image file (interpreter code), line by line with line format check																										
bin	machine code or other binary data,																										
dat	variable data																										
dim	block of all dimensioned data																										
imb	Binary image block of BASIC program with code, not line by line																										
rsv	ReSeRVe mode data (image)																										
def	Def key image (PC-1500, used for example in PC-BASIC 84)																										
tap	Raw byte format (replaces wave files for some emulators)																										
raw	raw binary data for debugging, =rawdat with quaters swapped																										
-p, --pc=NUMBER	SHARP pocket computer, for BASIC-program text, determines token table, is required for PC-1421 and PC-1600 Mode 1, optionally for other PCs																										
-c, --cspeed=VALUE	Ratio of CPU frequency to original (use it with a modified Pocket Computer, if speedup was switched on, 0.2 to 2.7)																										
-u, --utf8=TYPE	Convert special characters (only for BASIC-program text) - see also "HowTo" <table><tr><td>no</td><td>to ASCII transcription with brackets [reversible]</td></tr><tr><td>yes</td><td>to UTF-8 (default)</td></tr><tr><td>2dos</td><td>to DOS-US</td></tr><tr><td>2asc</td><td>to ASCII transcription for later serial transfer</td></tr><tr><td>2esc</td><td>to ASCII with escaped special chars</td></tr></table>	no	to ASCII transcription with brackets [reversible]	yes	to UTF-8 (default)	2dos	to DOS-US	2asc	to ASCII transcription for later serial transfer	2esc	to ASCII with escaped special chars																
no	to ASCII transcription with brackets [reversible]																										
yes	to UTF-8 (default)																										
2dos	to DOS-US																										
2asc	to ASCII transcription for later serial transfer																										
2esc	to ASCII with escaped special chars																										
-w, --width=VALUE[_]	(1-st) Minimum width of line number (only for BASIC-program text) <table><tr><td>2 to 10</td><td>(default: 5)</td><td>for text format</td></tr><tr><td>0</td><td>indentation off</td><td>for pure serial transfer</td></tr></table>	2 to 10	(default: 5)	for text format	0	indentation off	for pure serial transfer																				
2 to 10	(default: 5)	for text format																									
0	indentation off	for pure serial transfer																									
=[VALUE]: or ; , . !	Colon: Line number ends with colon, {;} col+spc, {,} col+spc+col, {.} not, {!} del																										
-w, --width=VALUE[_]	(2-nd) Level of spaces to insert before commands (for BASIC-program text) <table><tr><td>0 to 8</td><td>(default: 5)</td><td>for text format</td></tr></table>	0 to 8	(default: 5)	for text format																							
0 to 8	(default: 5)	for text format																									
=[VALUE].	Line number ends with Point: delete unnecessary spaces after commands																										
-o, --codepoints=File	File with definition of the code page of a pocket computer, usable 2 times, with list of special characters, format of one line: (Utf8-) Char=HH (Pocket)																										
-k, --keywords=KeywordFile	Token file with tokens of hardware options, 3-times usable List of tokens, format of one line: Tokenstring=HHHH																										

Usage: wav2bin [Options] SrcFile(.wav/.tap) [DstFile(.typ)]

-iOFFSET, --img= Source file: IMAge, ST or SHC format, no wave, Start offset in bytes: first=0
 --tap Source file: Emulator tap format (with header and checksums), no wave
 --parameters=CfgFile Write header values and parameters to CFG file, -m(default: DstFile.CFG)
 -x, --exit=TYPE Exit after header processed with value of: error (default), type, ident, pcgrpid
 -q, --quiet Quiet mode (No display output)
 --version Display version information
 --help Display help screen, more screens with =r : recording FAQ, =l : level(debug)
 -e, --endmark Append an EOF mark to a file, can used with BAS, ASC or IMAges, TXT
 -l, --level=VALUE Special and debug options, 2-times usable
 a hexadecimal integer (0x____) or sum of it

0x800 Ignore false checksums and continue
Destination file must be corrected manually!
 0x200 Finish lines before, when a CR [0D] is read

 0x1000 Depress base frequency tuning, use fixed base frequency only
 0x2000 Use flat method of Wav2Bin 1.5 to detect H/L-transitions,
 no detect amplitudes
 0x4000 No last amplitude based gain, count amplitudes only
 0x8000 No analysis of wave file, no pre-amplification

 0x80 Print lines of text written (BASIC only)
 0x40 Print all Bytes and checksums in brackets
 (Checksum readed = Sum calculated) - see also Bin2wav
 0x400 Byte list + wave time (after the byte)
 0x80000 BASIC commands of BMC MC-12 are used with priority
 (BASIC of PC-1500/1600 only)
 0x100000 Deactivate build in token, use loaded (-k) tokens only
 0xC00000 Convert to lowercase,
 0x800000 commands, 0x400000 characters outside of strings
 0x3000000 Remove leading comment characters (x1) or apostrophes (x2)
 0x4000000 Insert no space after REM (before transfer with CE-158)
 More options exist for low level debugging - see the source code in PrintHelp
 and Wav2bin_Debugging_Hints.txt

More write options: --type=tap convert wav to tap file, no checksum control
 --type=raw for debugging and =rawdat (quaters swapped)

Device specific filters: --device=CAS: (CS) for recording directly from tape with unstable signal
 --device=EMU: Wave from emulator via system sound, set --cspeed and -pc
 --device=BIN : Wave from Bin2wav or digital captured from Pocket port, set -pc
 --device=AIR : PC-1245-1475 buzzer -> Mic (tricky, avoid near reflections,quiet)

6.3. Bas2img

Usage: bas2img [Options] SrcFile [DstFile]

SrcFile : BASIC program text file

DstFile : Binary image file or special ASCII file (default: SrcFile.img or SrcFile.asc)

Options:

- p, --pc=NUMBER : Sharp pocket computer, currently supported
 1150, 1211, 1245, 1248, 1251, 1261, 1280, 1350, 1360, 1401
 1402, 1403, 1421, 1425, 1430, 1445, 1450, 1460, 1475, 1500 (default)
 1600, E220, G850, and more, also E500
 Only at E500 series must be used the commands 'TEXT' and then 'BASIC' after an Image transfer.
- t, --type=TYPE : destination file type (default: img)
 - img BASIC program binary image with intermediate code
 - txt TEXT mode image without token but with binary line numbers
 - asc ASCII file (for device CAS: or Text Editor Menu Cmt)
 - asm(5|6) asm for PC-1500 MACBAS (assembler source code in Basic lines),
asm5 für PC-1500 PC-MACRO, PC-E500 Dump Tool
 - c asm6 für PC-1600 Assembler
- o, --codepoints= File with definition of the code page of a pocket computer, usable 2 times,
with list of special characters, format of one line: (Utf8-) Char=HH (Pocket)
- k, --keywords= Token file with tokens of hardware options, 3-times usable
List of tokens, format of one line: Tokenstring=HHHH
- q, --quiet : Quiet mode (minimal display output)
- help : Display help information
- version : Display version information
- e, --endmark : Append a Pocket (0xff) or serial (0x1a) EOF mark,
usable for an emulator or for serial communication
Please use only if it is really necessary!
- u, --utf8=TYPE : (yes) convert special characters (BASIC program text only) - see "How To"
no ASCII/ANSI without graphics characters but Japanese (grp.NEW), Chap.6.4.1
dos for texts with special characters, from DOS-based software, see Chap. 6.4.4
- a, --auto(=STEP) : (default=10) Replace missing line numbers according to an AUTO instruction
The first line number should be specified in the source text or
corresponds to the step size. No renumber in the lines!
- l, --level=SUM a hexadecimal integer (0x____) or sum of it, 2-times usable
 - : 1 Don't/do compile fixed line numbers (inline)
 - : 2 Append missing apostrophes at end of line
 - : 4 Don't replace shortcuts(.) with commands
 - : 8 Don't convert to upper case
 - : 0x10 Deactivate preprocessor with special chars conversion
 - : 0x80/(0x20) Print lines in /(hex in)
 - : 0x40 Print values out
 - : 0x100 Force convert Katakana characters (SJIS) from UTF-8 to [A1]-[DF],
also for Japanese ANSI files together with utf8=no
 - : 0x200 like --auto, if in addition to --auto specified: clears all
existing line numbers except before the first line of text
 - : 0x800 Depress some line errors, result may not editable
 - : 0x1000 Switch conversion of E/G -labels into string „labels“ off/-on
 - : 0x2000 Insert an apostrophe at the beginning of each line (type asm5, asm6 etc.)
 - : 0x100000 Deactivate build in token, use loaded tokens only

All comments outside lines with numbers will be removed if the --auto option was not used (lines beginning with '). Even empty lines or meaningless content after the line number are removed. Bas2img should also be used, to convert from general BAS-files to Sharp-specific ASC-files for device CAS or serial communication.

6.4. Understanding the command-line options

6.4.1 Pc=Number

This command-line parameter is necessary for writing Basic source code by Wav2bin if there are conflicting token tables for the same cartridge file format, such as 1401/1421 or 1500/1600 M1.

For the other tools, this parameter is always necessary (except for the PC-1500). It is possible and in some cases necessary to use strings for this parameter, but these are internally mapped to numbers, for example "1403H" to the number 1403.

Then, in the second step, the numbers of the Pocket Computers with the same format are grouped internally. For each group a uniform token table is used.

The following group formation is applied within Bas2img.

Group 1211	1210, 1211, 1212	
Group OLD	1150, 1245 - 1255	including group 1211 tokens
Group NEW	1260 - 1262, 1350, 1401/02, 1430/31, 1450	(1421 separately)
Group NEW3	1403, 1425, 1440, 1445, 1460	including group NEW tokens
Group EXT	1280, 1360, 1470, 1475	
Group E	E500 series and successor,	
Group G	E220, E200 and G801 - G850S, G850V(S)	three subgroups
Group 1500	1500 series	(1600 Mode 1 separately)
Group 1600	1600 series	including group 1500 tokens
Group MZ	(experimental only)	

Only in the few cases when, despite the same file format on the Pocket Computer, the token numbers within the different models are assigned with different commands, additional token tables are used within a group.

Because separate token tables are not kept for all Pocket Computer models, versions, and options, there are limitations due to Bas2img to BASIC variable names. The variable names must not only differ from all the commands of the respective Pocket Computer, but also from the commands of the other computers in the group.

Where this is ignored, these variable names are tokenized by Bas2img and then displayed on the Pocket computer as a tilde character "~". Such a program is not executable and must be reworked.

The tilde characters are also displayed if non-100% compatible BASIC programs from another Pocket Computer are used, or if Bas2img uses a different PC option.

6.4.2 Type=Type

For Wav2bin, this parameter specifies which file format to use for the personal computer (for example, BASic source code, binary IMAge, SHA, SHC, tap file) or which method (eg, IMG or IMB, Raw or Rawdat) to convert a tape file.

For the other tools, the parameter must specify which target format for the Pocket Computer the source file is to be converted into. This can not be sufficiently determined on the basis of the content of the source file alone.

6.4.3 Device=Type

With this parameter, the Pocket Tools are provided with information about the interface and the source of the digitization. With Wav2bin this changes the internal parameters for filtering and amplifying sample values and thus the detection of amplitudes and zero crossings.

- Device=CAS (2) This parameter can optionally be used when digitizing directly from a cassette recorder (from tape). As a result, fluctuations in the zero line (due to DC voltages) are better compensated, but other errors possibly worse, such as low levels of the PC-1500. For PC-1600 / E/ G you have to use the parameter as soon as a whole wave is one-sided to the zero line, for PC-1245-1475 at low 4kHz signal.
- Device=CAS0-CAS4 Use these options when recording asymmetric signals, which causes the waves in the audio editor to appear at the top or bottom edge.
- Device=CS0-CS4 Use this option for symmetric balanced signals only weak at points. Use CAS0 for constant high signal but up to CAS4 for occasionally very weak signals (Gap found), especially for PC-1500 (not useful for PC-1245-1475).
--device=CS is like: --device=CAS and --level=0x4000
- It may be possible to correct for fuzzy and oscillating signals (CAS0-2) or clean signals with soft points (CS2-3), but not both at the same time.
- Device=BIN This option can be used if files created by Bin2wav are to be back-converted. Analog errors do not have to be compensated.
- Device=EMU This option should be used for wave files recorded by an emulator through the system sound. Since the speed does not correspond to the original, the parameters --Pc (to select the synchronisation frequency) and --Cspeed (speed factor) must always be set at the same time.
- Device=AIR This option should be used for wave files recorded without a cassette interface with a microphone from the piezo beeper on a Pocket Computer. Even in a very quiet room, errors can hardly be avoided, so that the parameter --Level 0x800 should always be set at the same time.
- Device=INV This parameter for Bin2wav mirrors the signal at the zero line and must be applied to ASCII formats via inverting cassette interfaces.

6.4.4 Command Line Options for text formatting and others

- Utf8=Type: see Chap. 15.1-2 Special Chars and und how to type in source code
=yes: to graphic characters, =no: [TAG], =escx, esc: \TAG, 2asc, 2dos
- Width=number(sep)(eol) For example, "-w5:" replaces the standard space after the line number with a colon, and "-w5;" inserts it between them.
-w, left justified format '123 : ABC' (with text modules. ": CMD ")
such as -w. '123ABC' (ASCII format for PC-G series)
-w! deletes all leading line numbers
Eol: n =LF, r =CR, default =CRLF (for example -w0_n)
- Width=number(.) (2nd) 0-8, level of spaces before commands, '.' deletes them behind commands
Example: Much spaces: -w10, -w8 No blank as possible: -w0. -w0.
- Endmark Use for memory images, not for transfer to tap files or wav files
if the option -e was used for an image, then apply it to all tools with this file
- Keywords= token file, cOdepoints= codepoint file (Additional tokens or special characters)
paraMeters=cfg file : see Chapter 5 (CFG file)

7. Old Parameter format

This format was reactivated for compatibility reasons only and does not support new parameters. Use it only with old software, that needs this format!

WAV2BIN *SrcFile DstFile [T:type] [G:graph] [D:level]*

<i>SrcFile</i>	WAV file, which must be compliant to the following rules :
<i>DstFile</i>	Output file, the format will depend of the source file : - ASCII file, if the source WAV contains a Basic program, - Binary file, if the source WAV contains Assembly Program or Datas,
<i>type</i>	Destination file type (option for the BASIC sources files) : BAS or IMG
<i>graph</i>	Convert special characters (option used with T:BAS) : NO or YES
<i>level</i>	Print debug traces

BIN2WAV *SrcFile DstFile T:type PC:num [A:addr] [S:sync] [N:name]*

<i>SrcFile</i>	Binary file
<i>DstFile</i>	WAV format output file.
<i>type</i>	Source file type : BIN or IMG
<i>num</i>	Destination PC number : for example: 1500
<i>addr</i>	Start adresse for the binary files (option for T:BIN), e.g. 0x0000
<i>sync</i>	Synchro duration, expressed in seconds: 1 (2) ... 8
<i>name</i>	Sharp file name of length 7 / 16 (PC-1500)

BAS2IMG *SrcFile [DstFile] [PC:type] [/Q] [/?]*

<i>SrcFile</i>	Basic source file in ASCII
<i>DstFile</i>	Binary Image destination file
<i>type</i>	PC type : for example: 1500
<i>/Q</i>	Quiet mode (disable display)
<i>/?</i>	Display the help

8. Error and other codes, returned to OS

Exit codes	Error description
ERR 1	Arguments missing, syntax error or nothing to do
ERR 2	Misplaced bit order in the wave file, nibbles in bytes or other
ERR 3	Arguments problem, for example: pocket not implemented
ERR 4	Error with line numbers
ERR 5	File I-O error or file used by another software
ERR 6	Image line too long or other buffer overflow
ERR 7	False or unknown format of wave or SHARP file header
ERR 8	Transmission Error, Checksum read differs from calculated result or result would not be readable
ERR 9	No synchronisation found in wave file or unexpected lost
ERR 10 (and higher)	Multiple errors found: Error number = last error + 10
Exit codes	from wav2bin --exit=type
	after reading the header detected type of Sharp file
TYPE 1	Binary code (machine code or data)
TYPE 2	Image with Basic intermediate code
TYPE 4	Special binary variable data file
TYPE 5	Image of ReSerVable keys memory with intermediate code
TYPE 6	PC-E/G/1600 ASCII Data
TYPE 7	PC-E/G/1600 ASCII Source
TYPE 8	Basic text modus image without token
TYPE 9	Image of PC-1500 definable keys memory with intermediate code
TYPE 10	PC-1500 Quick-Tape, binary image of all dimensioned data variables
Exit codes	from wav2bin --exit=ident
	Identity in the header of the Sharp file or used internally by Wav2bin
511	Raw modus was selected
257	PC-1600 Basic
258	PC-1600 RSV
272	PC-1600 IMG-Format from SAVE CAS without ",A"
All other	Corresponds to the identity byte in section 10. "Cassette file identity"

Not all operating systems support all return codes.

By using these return values as well as the parameter files generated with wav2bin -m and readable with bin2wav -m, it becomes possible to integrate the Pocket Tools into a separate user interface.

9. Supported SHARP formats, cassette files and commands

PC-1211, 1210, 1212

0x80	PC-1211 Basic image <u>or</u> RSV image CSAVE / CLOAD
0x8F	PC-1211 data, <u>one</u> data block of standard variables PRINT # / INPUT # Data stored with header of Old data 0x24, A(27) and following removed from end, if equal zero (use CLEAR after any changes of the program) No binary available, because for 4-bit CPUs not available
Shc	Supported, use Pctyp: PC-1251, but avoid unknown token for 1211

PC-1251-1255, 1245, 1246-1248, 11x0

0x20	Old format, Basic image <u>or</u> RSV image CSAVE / CLOAD
0x21	Old format, Basic image with password (only Wav2bin)
0x24	Old data, <u>multiple</u> blocks, PRINT # / INPUT #
0x26	Old binary, assembly program or other datas CSAVE M / CLOAD M no binary for 4-bit CPU available (PC-1246-1248)
Shc	Supported

PC-1260-1262, 1280, 1350, 1360, 1401-1475

0x70	New format, Basic image <u>or</u> RSV image CSAVE / CLOAD, also tokens of 2 byte length are supported (for CLOAD@ with PC-E200 series, use -p1460)
0x71	New format, Basic image with password (only Wav2bin)
0x72	Extended format, Basic image <u>or</u> TEXT modus image <u>or</u> RSV img. CSAVE / CLOAD (PC-1280, 1360, 1470U, 1475) (for CLOAD@ with PC-E500 series, use -l 0x1000 / 2bin: -p475)
0x73	Extended format, Basic image with password (only Wav2bin)
0x74	New/Ext. data, also <u>multiple</u> blocks, PRINT # / INPUT #
0x76	New/Ext. binary, assembly program or other datas CSAVE M / CLOAD M no binary for 4-bit CPU is available (PC-1430/31)
Shc	Supported

PC-1500, PC-1600 Mode (--pc=1600M1)

Id	Sub-Id	
0x A 0		PC-1500 binary, assembly program or other datas, CSAVE M / CLOAD M
0x A 1		PC-1500, Basic image CSAVE / CLOAD,

		for PC-1600 tokens use --pc=1600M1
0x A 2		PC-1500/1600 RSV image CSAVE / CLOAD from RSV mode
0x A 3		PC-1500 DEF key image, for later versions of PC-1500 LOAD with special software only
0x A 4		PC-1500/1600 data, multiple blocks, PRINT # / INPUT #
Shc		Not supported, but it is between new/old data and PC-1500/1600 variables converted by Bin2wav

PC-1600 with CE-1600P (--pc=1600P)
Default CAS: I/O only, No Double write!

Id	Sub-Id	
0x 0	0	ASCII Data, OPEN CAS:, INPUT#n / PRINT#n <u>or</u> Basic/ Binary image, splitted in blocks. with SAVE/COPY TO CAS: (without ,A :only with Wav2bin)
0x 1	0	PC-16/E/G binary, assembly program or datas, CSAVE M / CLOAD M
0x 2	1	PC-1600, Basic image CSAVE / CLOAD, if 16-byte file header included: Wav2bin -i16 or Bin2wav --type=i16
0x 2	2	PC-1600/1500 format RSV image CSAVE / CLOAD from RSV mode
0x 4	0	ASCII source file, SAVE/LOAD CAS: (with ,A) <u>or</u> ASCII comment lines, SAVE* CAS:
0x 8	4	PC-1500/1600 data, multiple blocks, PRINT # / INPUT #

PC-E500- E650, 1480-1490, U6000

Default CAS: I/O parameter only!

Id	Sub-Id	
0x 1	0	PC-16/E/G binary, assembly program or datas, CSAVE M / CLOAD M
0x 2	1	PC-E500, Basic intermediate code image <u>or</u> TEXT modus image, CSAVE / CLOAD,
0x 4	4	ASCII data, OPEN CAS: , INPUT#n / PRINT#n <u>or</u> ASCII source file in blocks, SAVE/LOAD CAS: <u>or</u> Basic image, max. 3821 bytes, COPY TO CAS: (only with Wav2bin)

PC-E200, E220,G801-G850

Default CAS: I/O parameter only!

Id	Sub-Id	
0x 1	0	PC-16/E/G binary, assembly program or datas, CSAVE M / CLOAD M
0x 2	1	PC-G800, Basic image CSAVE / CLOAD (G850V(S): BSAVE/ BLOAD)
0x 4	1	ASCII data, OPEN CAS: , INPUT#1/PRINT#1 <u>or</u> ASCII source file in blocks, with the "Text Editor Cmt" menu - except G850V(S): With the G850V(S), however, you can transfer ASM and C source texts with Bas2img / Bin2wav via BASIC to the text editor, but only with comment mark before each line unadulterated back.

PC-1500 Quick-Tape (--pc=**1500QT**)

Id	Sub-Id	
0x A 5		PC-1500, Basic image QSAVE old version with standard (0xA1) header, with Wav2bin only
0x A 6		PC-1500 binary, assembly program or datas, QSAVE M old version with standard (0xA0) header, with Wav2bin only
0x 0AA0	0x42	PC-1500, Basic image QSAVE / QLOAD new version with Quick-Tape header and blocks with standard length
0x 0AA0	0x4D	PC-1500 binary, assembly program or datas, QSAVE M / QLOAD M add to start address for ME1 0x40000, PV1 0x20000, PU1 0x10000
0x 0AA0	0x52	PC-1500 format RSV image QSAVE R / QLOAD R
0x 0AA0	0x44	PC-1500 format DIM image (all dimensioned variables together) QSAVE D / QLOAD D
0x 0AA0	0x56	PC-1500 data, multiple blocks, variable names lost QSAVE V, with Wav2bin only, retransfer with INPUT #only tested with IWS interface from ECPS and last EPROM with the QTape4 options
	Shc	Not supported, but it is between new/old data and PC-1500/1600 variables converted by Bin2wav

PC-1500 SuperTape (Output only, for input use c't SuperTape)

0x00	PC-1500, Basic image <u>or</u> BIN image SuperTape for PC-1500 LOAD only with Bin2wav, (use --pc= 1500ST) (use --pc=ST3600 without intro)
0x80	(use --pc=ST7200) SuperTape 7200 baud, for MZ700, MZ800 (MZ80)

10. Installation and use, start menu, scripts, operating systems**10.1. Installation for Windows**

1. The Pocket Tools can be started directly from a USB stick and used completely or simply copied to a folder at your hard disk. On hard drives, a subfolder of the folder \PortableApps\ is recommended.
2. Replace the PStart.xml file of the POCKTOOL folder with the version in your preferred language from the PStart subfolder.
3. Install an audio editor if you want to digitize tape recordings. Even if you only want to transfer something directly from a pocket computer, this is recommended.
If you are not yet using an audio editor, we recommend Audacity 2.
For portable use, "Audacity Portable" or the version without installer (ZIP file) is recommended.

4. Install a source code editor. If you do not yet use a source code editor for BASIC files, PSPad 5 is recommended as a ZIP file (without Installer!).

Read the file \PortableApps\pspad.add\PSPad_INI_changes.txt. Copy the two provided syntax and context files for "SHARP Pocket BASIC" into the two folders of your PSPad installation and adapt PSPad.ini according to the instructions.

Simultaneous use of PSPad for Visual BASIC *.bas must be turned off.

For conversion to UTF8 files it is also useful to use Notepad ++.

5. If you do not want to use PSPad, copy a hex editor into PortableApps, or at least a viewer for binaries (images). The HEX Editor was tested by MiTeC.cz. For intensive use, the Hex Editor XVI32 can also be used.
6. **Before using the Tools first time, you need to apply all global settings in accordance with your environment by editing the SHARPSET.bat file with a text editor!**

Uncomment or add our SHARP PC type (without "PC-").

Change the environment variables to match your directories and editors.

7. Customize the PStart menu to your used applications and paths.
Create shortcuts to Pstart.exe (and any separate cmd files) on your desktop or elsewhere.

You can integrate more scripts from Scripts.win into the PStart menu, if you need them. For example, WavEbas.cmd converts a wave file containing an image of the PC-E500 to a source file.

10.2. Installation for other operating systems

For Linux, you can use the enclosed scripts from Bernhard and thus also compile the source code. Also for OSX, you need to compile the source code and use alternative wrappers.

10.3. Use with Windows

Scripts for "Windows" (NT4, tested with 10) are included., see _ReadMe.txt.

There are the following options for using the scripts.

- A) Use the start menu of "PStart.exe". PStart can also be accessed via the taskbar.
The menu can be edited and adapted to personal use.
It is stored in the PStart.xml file.
- B) Create desktop shortcuts for the required scripts. Drop the files to be converted onto the desktop icon (drag & drop). Drag & Drop to a linked icon of a script as well as directly to the script are supported. An external tool for file selection is included, see the GetFName script.
- C) Open a custom console window (Pocket Tools Console).
The **first parameter** for each script must be a "?" or a **filename**.
If you use a "?" for the first parameter of a script, the File Selection dialog will open, for example:
bas2wav ? --additional -parameters

Start to use the Pocket Tools with the scripts Bas2wav and Wav2bas.

Notes

The file names must not contain any spaces or special characters, for use of this see chap. 15.3.

Some scripts (for MC) also use an .ADR (.CAL) file or the newer parameter file (.CFG).

Before you customize scripts for your environment, you should check if this can be done by setting in the SHARPSET.bat file. If you need a conversion that does not come with a script, it is recommended to copy a script with a similar task and adjust the call parameters.

11. Use of the Tools, chain and examples

Convert a

Wav file into a BASIC source text (CSAVE, except PC-E500)	wav2bin <i>WavFile</i> < <i>BasFile</i> > recommended --pc= <i>Number</i> optional: --type=BAS --utf8=yes(no...)
Wav file into a BASIC source text for PC-E500 only (for CLOAD)	1. wav2bin <i>SrcFile</i> < <i>ImgFile</i> > --type=IMG recommended: --pc= <i>Number</i> 2. wav2bin <i>ImgFile</i> < <i>BasFile</i> > --img recommended: --pc= <i>Number</i> optional: --type=BAS --utf8=yes(no...)
BASIC source text into Wav file (CLOAD)	1. bas2img <i>SrcFile</i> < <i>ImgFile</i> > --pc= <i>Number</i> 2. bin2wav <i>ImgFile</i> < <i>WavFile</i> > --pc= <i>Number</i> optional: --name= <i>SharpName</i>
Wav file into a BASIC image file (with intermediate code normally)	wav2bin <i>WavFile</i> < <i>ImgFile</i> > recommended: --type=IMG --pc= <i>Number</i>
BASIC image file into a wav file (CLOAD)	bin2wav <i>ImgFile</i> < <i>WavFile</i> > --pc= <i>Number</i> optional: --type=IMG --name= <i>SharpName</i>
Wav file into a RSV image file (with intermediate code normally)	wav2bin <i>WavFile</i> < <i>RsvFile</i> > mandatory for PC121x-1475: --type=RSV (img) recommended: --pc= <i>Number</i> for PC-1475/1280 see Techn. Report P-055 and use -l 0x20000 for bin type
RSV image file into a wav file (CLOAD)	bin2wav <i>RsvFile</i> < <i>WavFile</i> > --pc= <i>Number</i> required for PC1500/1600: --type=RSV optional: --name= <i>SharpName</i>
Wav file into a DAT image file (PRINT#)	wav2bin <i>WavFile</i> < <i>DatFile</i> > --type=DAT(img) recommended: --pc= <i>Number</i>
DAT image file into a wav file (INPUT#)	bin2wav <i>DatFile</i> < <i>WavFile</i> > --pc= <i>Number</i> --type=DAT optional: --name= <i>DataName</i> recommended for PC-1600: -s1 -s2
Wav file into a BIN image file (CSAVE M)	wav2bin <i>WavFile</i> < <i>BinFile</i> > --type=BIN (img) recommended: -m --pc= <i>Number</i>

	<u>Note the start (and entry) address</u> or use parameter <code>--m</code> or write it to filename.ADR and filename.CAL with scripts.
BIN image file into a wav file (CLOAD M)	<pre>bin2wav BinFile <WavFile>--pc=Number</pre> <pre>--type=BIN</pre> <pre>recommended: --addr=StartAddress</pre> <pre>optional 1500/1600/E500 2nd.address :</pre> <pre>--addr=EntryAddress</pre> <pre>optional: --name=SharpName</pre> <pre>recommended (PC-E500): --device=INV/std</pre>
Wav file into a SHC image file (CSAVE (M) / PRINT#)	<pre>wav2bin WavFile <ShcFile</pre> <pre>--type=SHC</pre> <pre>recommended: --pc=Number</pre>
SHC image file into a wav file (CLOAD (M) / INPUT#)	<pre>bin2wav ShcFile <WavFile>--pc=Number</pre> <pre>--type=SHC</pre> <pre>optional: --name=SharpName</pre> <pre>--addr=StartAddress</pre>
Wav file into a SHA source text (CSAVE)	<pre>wav2bin WavFile <ShaBasFile></pre> <pre>--type=SHA</pre> <pre>recommended: --pc=Number</pre>
Wav file into a BASIC image PC-1600 - SAVE (CAS: without ,A)	<pre>wav2bin WavFile <ImgFile></pre> <pre>--type=IMG</pre> <pre>recommended: --pc=Number</pre>
Wav file into a BASIC source text, Text Editor Cmt-SAVE (CAS:,A) for PC-G/E/1600	<pre>wav2bin WavFile <AscFile></pre> <pre>--type=ASC</pre> <pre>recommended: --pc=Number</pre>
BASIC source into Wav file Text Editor Cmt - LOAD (CAS:) for PC- PC-G/E/1600	<pre>1. bas2img SrcFile <AscFile></pre> <pre>--type=ASC/bas</pre> <pre>--pc= Number</pre> <pre>2. bin2wav AscFile <WavFile>--type=BAS</pre> <pre>--pc=Number</pre> <pre>optional: --name=SharpName</pre> <pre>recommended (PC-E500 series):</pre> <pre>--device=INV/std (depends on interface model)</pre>
Wav file into a Source text from Text Mode CSAVE	<pre>wav2bin WavFile <BasFile></pre> <pre>see WavFile with intermediate code into BasFile</pre>
BASIC source text into Wav file Text Mode – CLOAD for 1280/1350-60/1450-75/(PC-E)	<pre>1. bas2img SrcFile <TimgFile>--type=TXT</pre> <pre>--pc= Number</pre> <pre>2. bin2wav TimgFile <WavFile>--type=IMG/txt</pre> <pre>--pc=Number</pre> <pre>optional: --name=SharpName</pre>

Wav file into ASCII data (OPEN CAS:, PRINT #n)	wav2bin <i>WavFile</i> < <i>AscDataFile</i> > --type=ASC recommended: --pc= <i>Number</i>
ASCII Data text into Wav file (OPEN CAS:, INPUT #n) <u>Problems described above.</u>	bin2wav <i>AscDataFile</i> < <i>WavFile</i> > --pc= <i>Number</i> --type=ASC optional: --name= <i>DataName</i> recommended (PC-E500 series): --device=INV/std (depends on interface model) (only if I/O error after 2. block)
Wav file into a binary raw tap image file (no check of plausibility or correctness)	wav2bin <i>WavFile</i> < <i>tapFile</i> > --type=TAP recommended: --pc= <i>Number</i>
tap file into Dstfile	wav2bin <i>tapFile</i> < <i>DstFile</i> > --tap all other parameter: see <i>WavFile</i> into <i>DstFile</i>
Image File into tap file (Bin2tap)	bin2wav <i>File</i> < <i>tapFile</i> > --tap all other parameter: see <i>File</i> into <i>Wav</i>
Image File into Basic text file	wav2bin <i>ImgFile</i> < <i>basFile</i> > --img --pc= <i>Number</i>
SHC-File PC-1234 without password into a Basic text file	wav2bin <i>ShcFile</i> < <i>basFile</i> > --img=9 --pc= <i>Number</i>
ST-File of SuperTape for Windows into a Basic text file	wav2bin <i>ShcFile</i> < <i>basFile</i> > --img=26 --pc= <i>Number</i>
BASIC source file for PC-1500 into a Wave file with SuperTape format (QuickTape similar, but --pc=1500QT)	1. bas2img <i>SrcFile</i> < <i>ImgFile</i> > --pc=1500 (also 1500ST) 2. bin2wav <i>ImgFile</i> < <i>WavFile</i> > --pc=1500ST optional: --name= <i>SharpName</i>
ASM source file to BASIC PC-Macro	bas2img <i>SrcFile</i> < <i>ImgFile</i> > --pc= <i>Number</i> --type=asm5 <--auto=5> (format macro assembler via BAS cload)
BASIC source file for ASCII serial transfer	bas2img <i>SrcFile</i> < <i>AscFile</i> > --pc= <i>Number</i> --endmark --type=ASC (format text for transmission via serial interface)
BASIC source file into Emulator Image	bas2img <i>SrcFile</i> < <i>ImgFile</i> > --pc= <i>Number</i> --endmark (copy it into the RAM of the emulator with end mark)
Recorded wav file into clean synthetic wav file	wav2bin <i>WavFile</i> < <i>File</i> > --type=IMG --parameters <= <i>ConfigFile</i> > < --pc= <i>Number</i> > bin2wav <i>File</i> < <i>WavFile</i> > --parameters <= <i>ConfigFile</i> > (only for formats that are supported not only by wav2bin but also by bin2wav)
More variants exist.	see PHelp, Scripts.win_ReadMe.txt

12. Default start addresses (for load) of binary code

Read a system manual of your Pocket Computer, how to reserve the memory, before you use this:

PC number	Default load address (hex)	Before entering "CLOAD M" you have allocate memory with:
1245-1255	B830	Pointer Begin Basic: C6E1/2, NEW
1260	5880	Pointer Begin Basic: 66E1/2, NEW
1261, 1262	4080	Pointer Begin Basic: 66E1/2, NEW
1401, 1421	3800	Pointer Begin Basic: 46E1/2, NEW
1402	2000	Pointer Begin Basic: 46E1/2, NEW
1450	2030	Pointer Begin Basic: 5F01/2, NEW
1350	2030	Pointer Begin Basic: 6F01/2, NEW
1403	E030	Pointer Begin Basic: FF01/2, NEW
1403H, 1425, 1460	8030	Pointer Begin Basic: FF01/2, NEW
1360	8030	Pointer Begin Basic: FFD7/8, NEW
1475, 1280	8030	Pointer Begin Basic: FFF0/1, NEW
1500 (1555, 1559, 1561)	40C5 (38C5, 20C5, 00C5)	NEW Address_after_code or set 7865/6
1501, 1500A	7C01	NEW Address_after_code or set 7865/6
1600	C0C5	NEW "Sn",length+C5 or set F865/6 or CALL &02DD,Len (barcode rsv, bgnptr F034/5)
E200, E220, G series	0100	MON, *USER end_address
E500 series	BE000	Set BFD1A-C to BE000 (before BF21B) with: POKE &BFE03,&1A,&FD,&B,0,&1C,0:CALL &FFFD8 Caution! Images of binary disk files may contain a header of 16 bytes length before the executable machine code

Please backup all your programs and data before you modify the pointers, use NEW or CLOAD M !
Read in the manual how to reset your system in case something unexpected happens.

13. Supported and tested Sharp Pocket Computers

All Sharp PCs are supported. The following Pocket Computers was tested with the Pocket Tools.

PC-1211, PC-1212 with CE-121(122)

PC-1245, PC-1248, PC-1251

PC-1260, PC-1262, PC-1280

PC-1350, PC-1360,

PC-1401, PC-1402, PC-1403, PC-1403H

PC-1421, PC-1425, PC-1450, PC-1460, PC-1475 with CE-126P, CE-124, other, also 3rd-party,

PC-1500, PC-1500A, PC-1500 Quick-Tape, PC-1500 SuperTape, with CE-150 and CE-162E

PC-1600, with CE-1600P and with CE-150

PC-E500, PC-E500S with CE-126P, also 3rd-party,

PC-E550 tested from emulator only

PC-E220, PC-G850S only with CE-126P, full volume,

PC-G850V(S) tested with CE-126P (and from emulator)



13.1. Naming convention for Sharp Pocket Computers with special interface

Remove the string "PC-" from the name of the pocket computer before you use it with the parameters --pc=, e.g. with PC-1403H use --pc=1403H .

For some combinations special names are provided:

1600P	PC-1600 + CE-1600P
1600M1	PC-1600 Mode 1, e.g. PC-1600 + CE-150
150	PC-1500 + CE-150 with original formats only
158	PC-1500 + CE-158 for retransfer (for wav2bin only)
1500ST	PC-1500 with SuperTape software (for bin2wav only)
1500QT	PC-1500 with Quick-Tape (hard- or software)
1234	Unknown PC with 4 kHz base frequency (for wav2bin only)
E475	PC-E500 in the format of PC-1475 (for read back with wav2bin only)
E500M2	PC-E500 with character set 2 (for special chars only)

14. Description for some special characters and abbreviations

14.1. Important abbreviations for the source code with Bas2bin

For example the following abbreviations are replaced by complete commands by Bas2img.

P.	PRINT	U.	USING	I.	INPUT
G.	GOTO	GOS.	GOSUB	RE.	RETURN
T.	THEN	N.	NEXT	B.	BEEP

14.2. Special characters, partially across generations

The following special characters (tags) can be parsed from Pre-processor in Bas2Img with Ascii text (case sensitive). They are also generated by Wav2bin depending on the selected character set, also in file names.

√ [SQR] Square root	Π [PI] Pi	π [pi] pi	€ [E] Exponent of old series
¥ [Y] Yen	□ [INS] Insert Cursor	■ [FUL] Full Cursor (\BX)	
[SUB] [EOF] End of Ascii file, applied when at the begin of line [HH] byte, two hexadecimal digits			

Katakana: ([FE]) [A1] - [DF], ♦ [DIAMOND], ♥ [HEART], ♠ [SPADE], ♣ [CLUB],
 円 [YEN], 年 [YEAR], 月 [MONTH], 日 [DAY], "時 [HOUR], 分 [MINUTE], 秒 [SECOND]

Utf8 characters are internally converted to variable tags if possible before further processing.

Variable tags and Japanese characters are converted by the pre-processor in Bas2img, depending on the computer model, into different character codes that match the desired purpose. Fixed hexadecimal tags are converted to a fixed value that may have a different meaning in another computer model. When converting special characters for the PC-E/G series and -1600 Mode 0, the special character is used within fixed character strings and in comment lines, but it is resolved to the ASCII equivalent for other commands.

14.3. Processing Flow of special Chars

1. All codepoint files (.cfg) are read in and checked. The basic format is:
 UTF-8 character = hexadecimal value or [tag]
 The hexadecimal value is one byte (except Japanese characters FEHH for PC-1260 and newer).
 For Bas2img tags [] are allowed instead of the hexadecimal value. These are ignored by Wav2bin.
 If you are working with tags, it is recommended to define the first entry with tag (for Bas2img) and then a second one as hexadecimal value (for Wav2bin).
2. The imported Utf 8 characters are examined to see if they contain the graphic characters for the specially supported variable tags SQR, PI to FUL. If they are included, the hexadecimal value of the character (the value at the Pocket Computer) is registered for it.
3. For models with a code page integrated in Bas2Img, the same procedure is used for the internal table if the values have not already been registered in the previous step.
4. For unregistered characters (or if the code is above the range recommended for the interpreter), a default hexadecimal value is set according to the pocket computer character table, taking mode or active code page into account.
5. Then the line-by-line processing of the program text takes place.
 The program text may contain supported utf8 characters as well as variable or fixed tags that have been entered, copied from a codepoint file, or generated by Wav2bin.

Preprocessor

6. All Utf-8 characters of the code point tables are searched for in the program text and replaced by fixed tags or, if specified, variable tags.
7. When utf8 is turned on, the half-width katakana characters and most graphic characters of the internal tables become fixed tags as well
 the other supported Japanese characters, the playing card symbols and
 variable special characters converted into variable tags according to the previous section.
 With --utf8=no (for ANSI) only the katakana characters are converted with --level=0x100 (SJIS).
8. Variable special characters in the format of escape sequences are converted into variable tags.
9. Special characters from DOS files are converted into variable tags.
10. All variable tags are model-specifically converted to their value on the Pocket Computer, taking into account the registered values (2-4).
11. All fixed tags are converted to bytes (not for C and ASM files). If the parameter utf8=escx is set, then escaped fixed characters \xHH are converted to bytes.
12. If unprocessed Utf-8 characters or other bytes remain that correspond to the 1st byte of 2-byte tokens, a warning is issued due to the resulting list and editing issues on the Pocket Computer.

Tokenizer

13. Orphan wildcards when using codepoint files: [SQR] → SQR, [PI] → PI
14. Special characters that are outside of strings are converted to commands if they are not supported by the model and mode as special characters:
√ → SQR, Π, π → PI, ' → REM

Note: The alternate KeyWords files are for tokens, not special characters, but may contain commands in the variable tag format. The associated hexadecimal values (length 1 or 2 bytes) must match the length and with the first byte to the ranges of token tables known for the particular model in order for Wav2bin to process them.

15. How To**15.1. What type of conversion of special chars should be used with Wav2bin type BAS**

When using a modern editor that can work with UTF8 characters, then use `--utf8=yes` (default) This way you can handle supported graphic chars directly. You can copy the supported Utf8 characters from the cfg files SJIS, G850, E220, E500M2 and E500M1, 1600M0 into your BASIC source code. The last two files are already integrated and should only be used as a template. These CFG files are included. You can easily make others yourself.

Utf8 characters unsupported for your model generate bas2img warnings and invalid characters on the Pocket Computer.

When transferring data simple between different systems, transferring programs between different pocket computer generations, or using a older editor, use `--utf8=no` . All special chars are converted to placeholders in square brackets. Bas2img will transfer it back depending on the target model. The option `--utf8=escx` converts to `\xHH` instead of `=no` (format [HH]).

With the options below, graphic characters (except SQR and PI) and special characters are neither converted to [tags] nor Utf8 characters, but are left in the row text.

If you want to transfer the source text via serial transfer later, (or use very old basic text for a later pocket generation,) use `--utf8=2asc` . PI and SQR are resolved, and some stars * are added between old variable names. The option `--utf8=esc` works identically, but variable tags are marked with a \ character.

If you want to use the source text with a DOS-based software, then use the command line option `--utf8=2dos`. This will convert many of the special chars to codepage 437.

This is necessary for "Transfile PC" and old DOS programs for serial transmission.

15.2. How should you type in the Basic source code?

1. First try to find the source code on the Internet or on a CD about Pocket Computers. If the quality is sufficient, you can try to scan the text and apply an OCR software for text recognition.
2. Otherwise, you must enter it with a text editor.
Use an editor that supports ASCII and utf-8 characters.
It is best to use an editor with syntax highlighting. There are text editors that allow the use of text modules. These text blocks can be assigned to common BASIC statements.
3. The line numbers do not need to be aligned. Although they can be separated by spaces and a column of the subsequent program text, the Tools do not require any separation, see also point 8. If you only use jump labels und not line numbers, you can create all line numbers new with Bas2img `--auto -l0x200` and remove all with Wav2bin `-w!` . The *LABEL s of the E/G series are converted by Bas2img into string "LABEL"s for PC models that do not support E-labels.
4. Spaces outside of strings (not enclosed in apostrophes) and comments can be ignored.
You do not need to enter this spaces. Only for the PC-E/G series are most spaces preserved.
5. Do not worry about upper and lower case, except within strings (and PC-1421: i, n as well as PC-1500+CE-158: CSAVE/CLOAD/MERGE a/r). For Pocket Computer that do not support lowercase letters, strings are also converted into uppercase by default.

6. Use abbreviations, for example, 'P.' or 'PR.' for 'PRINT'.
Abbreviations can be found in the manual for each command of the PC-1350 and later models.
7. Enter characters as utf-8 characters or with the following variable transcriptions:
Π [PI], √ [SQR], ¥ [Y], □ [INS], ■ [FUL], old exponent € [E], other hex code 0xhh: [HH] fixed.
8. Convert the source code into a IMAge (or TAP, WAVE) file and convert it back into a BAS file to resolve the shortcuts and format the text with spaces. Use the characters of both width[_] parameters of Wav2bin to generate the correct line number format for your editor.
9. Enter blank lines and comments outside lines with leading ' (without --auto) at the very end of your work, because they are lost each time you convert the text. Comment characters within the line are converted to REM for the PC models where the ' character is not supported.

15.3. Use file names with spaces and special chars

1. All underline-characters _ are replaced by spaces for PC-1210 upto PC-1500 in file names.
2. All characters 0xhh can be described as [HH] and will be replaced by Bin2wav in file names.

15.4. Best method to transfer BASIC source files to PC-E500 series

We recommend to also use the simple CSAVE command for PC-E500 series.

If you generate a image file from a source file with Bas2img, then the numeric data in the lines is not converted to BCD and the binary distances for jumps are not generated. This is a limitation of this version of Bas2img for the format of PC-E500 series only. The transmitted program text is complete, but the intermediate code is not fully executable. That's why the pocket computer has to be switched to TEXT mode and back to BASIC after the transfer to perform these necessary optimizations.

Nevertheless, you should use this format because it is faster than the others and has a stable transmission performance. Alternatively, the following possibilities with other restrictions exist.

1. The format of older series (with Bin2wav -l 0x1000) for CLOAD@ is very slow (returnable from synthetic wave file with wav2bin --pc=E475).
2. The CSAVE format of the TEXT modus would be a "one way" for E500 series and
3. the ASCII wave format must sometimes be level inverted (depending from the interface), is slower and was optimized for an audio interface, that switches a cassette recorder off and on before every data block. Use ASCII format for short sources and to MERGE sources.

For archival purposes you should always keep a synthetic wave file (alternatively IMAge+ CFG file), not just the BASic file (especially for the E500 series). The wave file for this should be backed up by a checked executable program via CSAVE (for an image) and the synthetic wave file with CLOAD? be checked, see also chap. 15.11.

15.5. Convert PC-E500 to BASIC source files with Wav2bin

The PC-E500 replaces line numbers after GOTO and other jump instructions in the TEXT image file with absolute jumps, after the BASIC program was started once (Runtime optimization).

There are three ways to get the BASIC program with correct branch numbers from PC-E500.

- A) Switch to TEXT mode and back to BASIC. In this way you can make sure that all the text checked and, if necessary, the intermediate code is generated, that had not generated by Bas2img. In addition, it still contains line numbers instead of binary jumps.
- B) Add a dummy line and delete it, e.g. 65279:END or edit the text anywhere before the transfer.
- C) Working in two steps:

```
Wav2bin --type=img program.wav temp.img
Wav2bin --img -pc=E500 temp.img program.bas
```

If the source is an image file, Wav2bin can recover the line numbers from the jumps in 2. step.

15.6. Transfer of assembler and C sources from the PC-G850V(S)

Unfortunately, when developing the G850V(S) of the G850(S), the "TEXT EDITOR Cmt" was removed.

1. The source code must be saved as **Rfile** (RAM data file) *.DAT, for example T.DAT .
This file type must first be initialized in sufficient size.
2. Load the program **CMT.C** into the PC-G850V(S), execute it and thereby create a file CMT.DAT from the source text, whereby at the beginning of each line a comment mark is inserted.
Also the file CMT.DAT has to be initialized once before.
3. Load the created **Rfile** CMT.DAT into the Text Editor and convert it to **Basic<-text**
4. Switch to BASIC and transfer the file via **BLOAD** to the personal computer
5. For Wav2bin, use the options --pc=G850V --type=c or --type=asm6 , to create a source code on the Personal Computer. The line numbers can be deleted with -w!

For the transfer to the G850V(S) the comment characters are not required: --type= c | asm.

15.7. Handle wave files from BASIC images, containing binary code

1. If the lines are constructed properly and contain only characters that are allowed, a image can transferred with --type=img. This will be done line by line and checks every line.
2. If the line structure is disturbed by the binary data included or line terminators are included, an other method should be used with --type=imb. Since the data is read in as a binary block and analysed less, the error detection and correction in this method is generally lower. However, some images can be read in such a way that the standard method breaks off beforehand.

Mostly the binary code is located in the first lines of the program text. You can then use a hex editor with the generated img file to find the 0D character after a BASIC program starts. Note the position of the byte after this 0D and use Wav2bin --img=position to create the program text.

3. With a hex editor or -dump, e.g. XVI32, you can copy the binary code from the IMG file as hex strings, then replace the spaces with "]" "[" and and paste them back into the lines as text.

15.8. Sample: Edit BASIC variables with "Transfile PC", transfer to emulator on Android

1. PRINT # variable data inside the commercial pocket emulator to a tap file
2. Export saved tap file to an public accessible directory with the emulator menu and
3. copy the tap file to a personal computer (USB cable or a Webdav server)
4. Open a command line window on your Personal Computer or use a script and
5. run Wav2bin with the options --tap --type=shc
This makes a shc file from the tap file.
6. Start Transfile PC plus 5.55 (runs inside DOSbox)
[ESC] [ESC] [F10]

```

      >Pocket
      >PC type      TOKE n table
      >Binary file load
[ESC] [ESC]
Edit the ASCII file: first line includes the filename and dimensions
[F10]      (Double Precision is not supported in SHA files.)
      >Pocket
      >Binary file save

```
7. Run Bin2wav with the options --pc=.... --type=shc --tap
This makes a new tap file from the edited shc file
8. Move the file to the Android device and import the tap file to the emulator
The OS file name must be exact identical to the SHARP-file-name (without ".tap")!
9. Use INPUT # to read the new variable data inside the emulator

Note: Everytime you have changed the PC type inside "Transfile PC", you have to open/load the source file or data text file (.SHA) again.

Two additional steps Tap->Dat, Dat->TapPc1350 are required before step 5 if the emulator of the PC-1500 is involved.

15.9. Use the very fast SuperTape format for PC-1500

1. Download [c't SuperTape für Windows](#), if you want to convert back Wave files into Binary Images or digitize cassette recordings. Otherwise it is not needed.
2. Use a SuperTape implementation for your Pocket Computer, for PC-1500 it is known at this time. A BASIC-installer for PC-1500(A) with CE-150 or CE-162E exist, and RSV files for convenience.
3. Install SuperTape to your Pocket Computer (PC-1500A: free RAM from 0x7C01 recommended).

Transfer to the Pocket Computer

4. Create an IMG with Bas2Img from BASIC Source or take a IMG or take a binary file from Supertape for Windows (.ST)
5. Use Bin2wav --type=img --pc=1500ST to create SuperTape wave files from images or Bin2wav --type=st --pc=1500ST to create SuperTape wave files from SuperTape binaries

For other computers use --pc=ST, ST3600 or ST7200.

With Bas2img exist limited support for MZ700 (not tokenized completely, as with PC-E500).

6. Play the wave file and load it with the commands of your implementation. The volume control is more sensitive than the other formats, but otherwise stably usable at 3600 Bd.

Transfer to the big PC

7. Save your program with the commands of your implementation and record the wave file with a sample rate of 44.1 kHz or better. The quality and volume are much more sensitive than the other formats. If the level is low while recording, a microphone preamplifier must be used mandatory.
8. Start SuperTape for Windows, set Input to "Raw-Datei" and Output to "Bin-Datei", Klick on the microphone symbol to open the file of your recording.

The conversion was only successfull, if the text window displays both:

Header Prüfsumme=ok

Daten gelesen, Prüsumme:ok

The result is saved as Name.ST automatically – look for:

Name.ST abgespeichert

9. If you want to convert the ST-File to a BAS or an IMG file, then use Wav2bin with the option --img=26 .

15.10. Use the Quick-Tape format

1. There exist different formats and versions of RWE Quick-Tape and their licensees. Not all formats are recognised or supported in both directions. Fast tape formats different from RWE are not supported at all.
2. The Quick-Tape format is sensible to quantisation defect from digitisation. Short bits may recognised for long bits. Mostly you can not read back a good 48 kHz r recording just from the PC. You have to create a file with Wav2bin and then create a new synthetic wave file by Bin2wav. This will be better readable than the original recording and easy to use.
3. QSAVE V specific informations for data variables are lost after processing with Wav2bin. The resulting DAT file (binary image) is stored in the same format as all the data files of the PC-1210 upto PC-1600. Use the PC-1500 standard format (INPUT #) for retransfer.
4. The special flags for QLOAD M must be passed with an additional upper byte of the start address.

15.11. Steps to backup and archive a BASIC program

1. Testing the runnability of the program (RUN) and afterwards the editability (insert comment line or edit) on the Pocket Computer
2. Backup the program from the Pocket Computer with CSAVE "NAME" and recording as a wave file
3. Convert the recorded wave file with wav2bin into an IMG (no Bas file) and then convert it to a synthetic wave file with bin2wav (see end of chapter 11)
4. Check the synthetic wave file created with bin2wav with CSAVE? "NAME"
5. Compress the synthetic wave file as a ZIP file, to better protect against changes, Keep the compressed file on a secure disk in a safe place
6. Conversion of the IMG or Wave file into a BAS file, indelible print on good paper and dry storage

With the synthetic (tidy and checked) wave file and a paper printout, you can restore your program later, regardless of the operating system and Pocket Tools.

If your software contains machine code, necessary data, etc., then you must also secure it. Make a note of the memory addresses and, if necessary, the length on the printout.

16. Changes in version 2.1 compared to version 2.0

The following changes to version 2.0 have been implemented.

1. Better support for PC-1600 and option --pc=1600M1 switched to Mode 1 (with CE-150/162E)
2. Convert from IMG (SHC, ST) to BAS directly
3. Resolve PC-E500s inline branch numbers (after run created) through two-stage conversion
4. Revised routine to synchronize start bit per block for PC-1600/G/E series, less errors
5. New --device=CAS with better compensation of some DC-errors, also for PC-1600/G/E series, more and improved options CAS, CS, CAS0-4, CS0-4 for PC-1500, CAS0-4 for PC-1234.
6. Wav2bin type IMB (read image as a binary block) as an alternative to IMG (line by line)
7. Writing in SuperTape format with limited support for Non-Sharp computer
8. Support for PC-1500 Quick-Tape
9. Sharp file names with special chars
10. More unknown keywords and japanese chars (reversible)
11. Priority of tokens switchable for BMC MC-12, Token for RVS PC-BASIC 84 and Ursoft
12. Using own external keyword files -k (token tables) and -o codepoint tables (special characters) for BASIC source code. PC-E220 and G850 enclosed cfg file, integrated for PC-E500 and PC-1600 (cfg file as a copy template for characters)
13. File type "Def Keys" for software of the PC-1500
14. Write information and options into and read them from -m parameter files (per IMG-file)
15. Test mode wav2bin -x to read the header of the Sharp file, returning values to the OS
16. Parameter --endmark to attach EOF marks to images (for use with emulator) and ASCII files
17. Different formats with colon, also distance-free, for the line numbers by wav2bin, as well as internal distances and the end-of-line character, AUTO line numbers
18. PC-Macro format, other assembler and C formats
19. Revised English and now also a German manual
20. Portable start menu with PStart, Linux Scripts from Bernhard
21. Support of Japanese characters for PC-1260 to PC-1475, PC-1500, G801-G850
22. Complete support for the PC-G850 series

Thanks to

the original authors

Jean-Francois LAROCHE, Joachim BOHS

the authors who had developed the software in the meantime

Norbert ROLL, Marcus VON CUBE, Manfred NOSSWITZ, Olivier DE SMET

for the debugging and other help

Christophe LE GLATIN, Emmanuel BEAUREPAIRE, Simon LEHMAYR, Christoph LAU,

Manfred BECKER, Edgar PÜHRINGER, Michael (irrlichtproject.de), Harald RICHTER

Eric MILLESCAMPS, Heinz-Bernd EGGENSTEIN, Bernhard LEX, Thomas SONNENBERG, Patrick FAVRE

for the support with hardware and information

Raymond BAKKER, Ralf DERSCHEID, Rik HARTL, Rainer KRATZER, Klaus DITZE,

Reto AMBÜHLER, Andrea WEIGERT, Jörg WALENTA, Thomas STAHL

Horst SCHAEFFER for his batch tool "Wfile", Jens GRUSCHEL(Pegtop) for his "PStart",

the GCC team, the makers of the MinGW project and Code::Blocks,

my family and all the others who have contributed to it, that this project could be realized

Torsten MÜCKER, developer of this new version, January 2020

This is free software. There is NO warranty, not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. We are not reliable for any consequences arising from the usage of it.

Only use it at your own risk.

All Trademarks, company or product names belong to their respective owners!