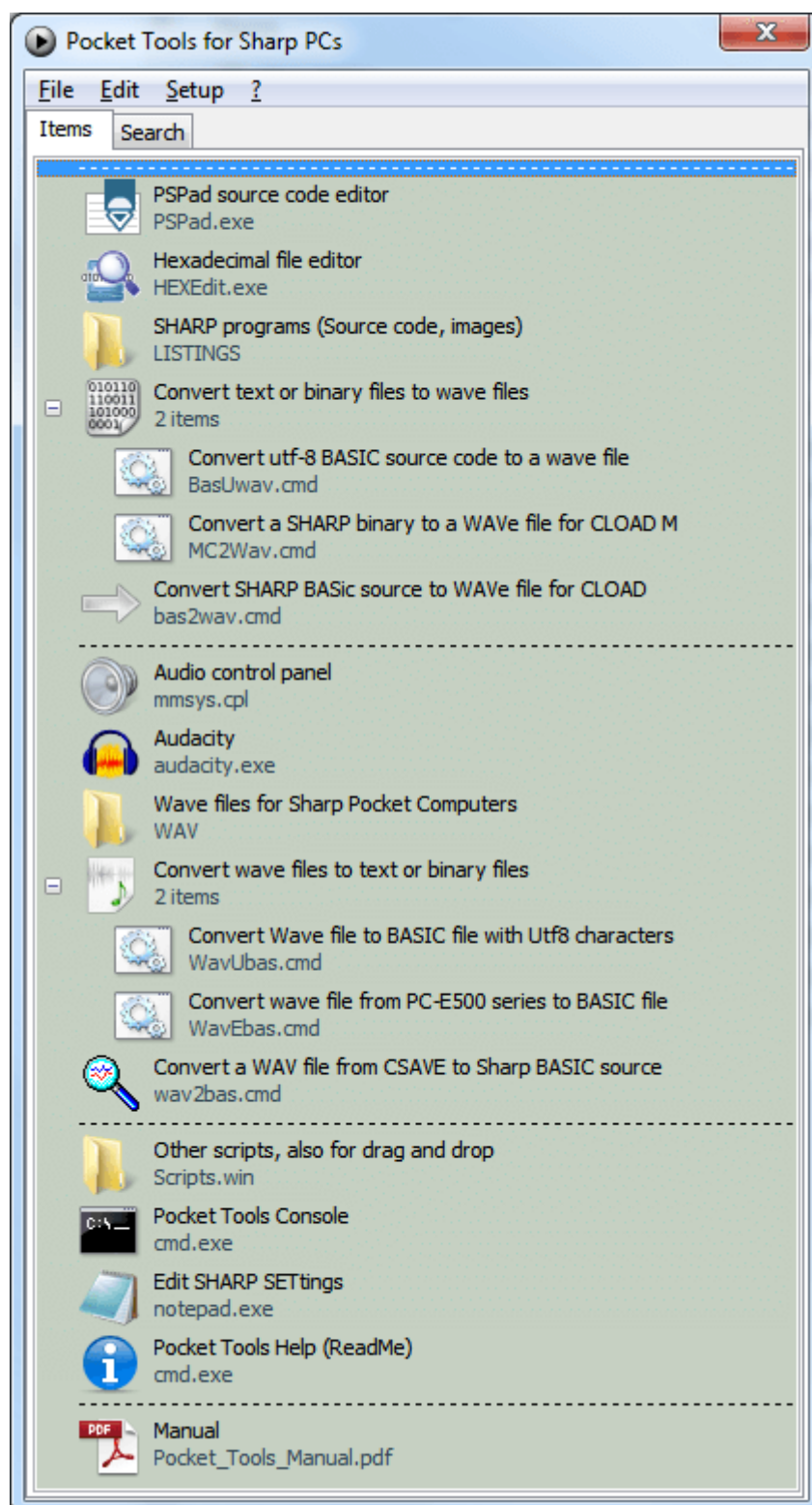


## Pocket Tools 2.1 for Sharp Pocket Computer



User Manual Revision 11b  
for Pre-release 2.11c1b  
14.05.2021

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	5
3.	Recording and playback	5
3.1.	How to record a wav file from a Pocket Computer?	5
3.2.	Recording guide for Audacity	5
3.2.1	What should the recorded wav file signal look like?	6
3.2.2	What should the recorded wav file not look like?	7
3.3.	How to prevent processing errors with Wav2bin from a wav file?	8
3.3.1	Recommended procedure for digitizing tape recordings	9
3.4.	How to transfer programs or data to a Pocket Computer using the tape interface?	10
4.	Feature list of Pocket Tools	12
4.1.	Technical limitations or not implemented	13
4.2.	Limitations on reading and playing wav files	14
5.	Handled file types	15
6.	Command line options	18
6.1.	Bin2wav	19
6.1.1	Bin2wav generatable sample frequencies (changed in PocketTools 2.1)	20
6.2.	Wav2bin	21
6.3.	Bas2img	23
6.4.	Understanding the command-line options	24
6.4.1	Pc=Number	24
6.4.2	Type=Type	24
6.4.3	Device=Type	25
6.4.4	Command Line Options for text formatting and others	25
7.	Old Parameter format	26
8.	Error and other codes, returned to OS	27
9.	Supported SHARP formats, cassette files and commands	28
10.	Installation and use, start menu, scripts, operating systems	30
10.1.	Installation for Windows	30
10.2.	Using Pocket Tools with Windows	31

10.3.	Installation instructions for other operating systems	31
11.	Use of the Tools, chain and examples	32
12.	Default start addresses for (loading of) binary code	35
13.	Supported and tested Sharp Pocket Computers	35
13.1.	Naming convention for Sharp Pocket Computers with special interface	36
14.	Description for some special characters and abbreviations for BASIC	36
14.1.	Important abbreviations for the source code with Bas2img	36
14.2.	Special characters, partially usable across generations	37
14.3.	Processing Flow of special characters by Bas2img	37
15.	How To	38
15.1.	What type of conversion of special chars should be used with Wav2bin type BAS	38
15.2.	How can you scan and enter Basic source code?	38
15.3.	How to specify file names with spaces and special chars	39
15.4.	Best method to transfer BASIC source files to PC-E500 series	39
15.5.	Convert PC-E500(S) to BASIC source files with Wav2bin	39
15.6.	Transferring of assembler and C sources from the PC-G850V(S)	40
15.7.	How to handle wav files from BASIC images containing binary code	40
15.8.	Editing BASIC variables with "Transfile PC" transfer to emulator on Android	40
15.9.	Using the very fast SuperTape format for PC-1500	41
15.10.	Using the Quick-Tape format for the PC-1500	41
15.11.	Steps to backup and archive a BASIC program	42
16.	Changes in version 2.1 compared to version 2.0	43

## **1. Introduction**

The Pocket tools are a set of programs with which you can convert between audio recordings of SHARP Pocket computer's programs, data, binary files and text files to edit and store on a personal computer.

You can edit these files on your PC with any editor and convert them back to audio files to load them into the SHARP Pocket computer or an emulator.

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

If you have a serial adapter cable for each of your SHARP Pocket computers that you can connect to a port of your PC, and 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 work, then you do not need the pocket tools. Especially for the newer SHARP series a suitable serial cable is often the most comfortable solution. The Pocket Tools can then only assist you with text formatting.

If you want to make sure that your programs and data can still be used with your SHARP Pocket computers for many years to come, then we recommend to learn to use the Pocket Tools.

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

### 1.2.1 Hardware

You will need a Personal Computer with a sound adaptor or sound card with a microphone input and headphone output. Some computers only have combined blue audio inputs, which you must first configure for the microphone inputs, and generally usable audio outputs (green).

The adaptor of your PC should support microphone pre-amplification. The cables to the audio terminal should not run next to power supply wires with high current. For very insensitive microphone inputs an additional external preamp may be required.

Second, you will need an audio interface, usually a cassette interface, for your Pocket computer. In general, the SHARP interfaces also work without batteries or weak batteries. All cables, interfaces or printers that contain rechargeable NiCd batteries should be charged about twice every quarter 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 SHARP Pocket computers: CE-124, CE-126P or another suitable (i.e. CE-123P, CE-129P or CE-120P) 11-pin commercial cassette interface. You can also use a DIY self-built interface adapter (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. Depending on the emulator software, you can load the text, binary, tap or wav files processed with the Pocket Tools directly into your emulator.

### 1.2.2 Software

You will need some audio recording software or an audio editor with recording function to convert the audio output of SHARP pocket computer to wav files, such as BASIC programs and data saved to cassette. But you can use the media player of your system for output.

The software you use must play wav files exactly and unchanged on the audio output.

Take note of the playback volume of the system (master) and the volume control of the player that work best without producing distortion. No driver is needed.

To edit programs, you will 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 or binary files can be edited with a hex editor. Please see chapter 5 for a list of supported file types.

### 2.2. Bas2img 6.1

With Bas2img you can convert a source text file containing SHARP BASIC language, C or assembler 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

For the processing of source texts (e.g. BASIC) you usually have to use Bas2img beforehand.

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. You should play them back unchanged and unfiltered with a media player to transfer the programs and data to the pocket computer.

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 CLOAD the program from the Personal Computer correctly.

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

## 3. Recording and playback

### 3.1. How to record a wav file from a Pocket Computer?

A sample rate of 11025 Hz may be sufficient for the PC-1500, but this reduces the fault tolerance. It is recommended to record from all PCs with a sample rate of at least 22050 Hz.

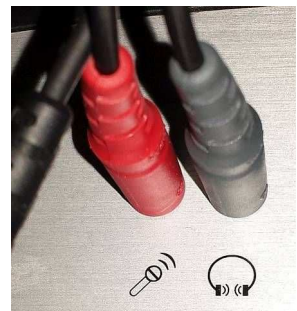
PC-1500 Quick-Tape and tapes should be recorded at a sample rate of 48 kHz (or 44.1 kHz).

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 only high levels and clear signals with 8-bit. It is recommended to record with 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 and mode (for example, as a microphone input instead of line-in). Disable all mixers and filters. Especially with extension cables, check that the plugs have not been switched. 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 start after a silent time. The preceding silent period can be omitted for Wav2bin, 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.

From version 2.0 this should not be done.

The wav file should be normalized for control purposes, but this is not absolutely necessary.

### 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 versions.

#### 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.

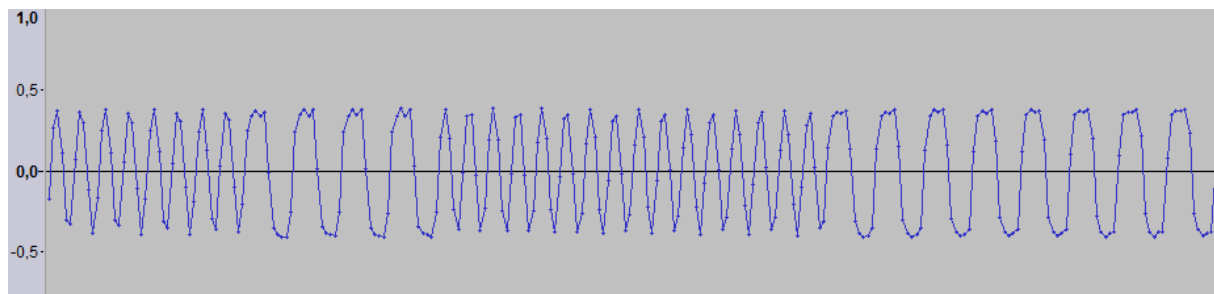
Switch microphone pre-amplification on in the preferences of your microphone-device in order to achieve a recording level higher than 5%. You may need to stay a little below the highest possible gain to ensure sufficient quality (if you can adjust it).

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 wav file signal look like?

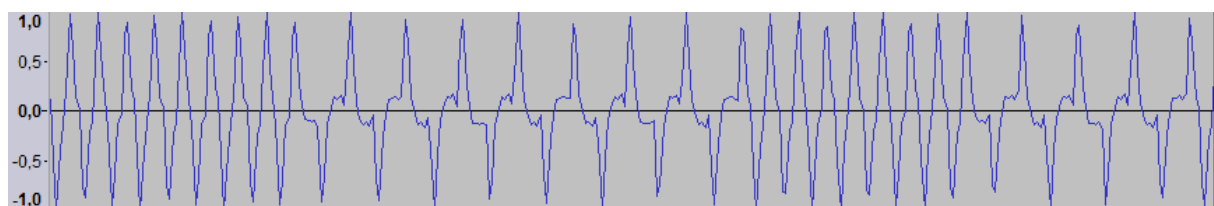
*Part of a wav file (PCs from PC-1210 to PC-1475)*



The number of signal 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).

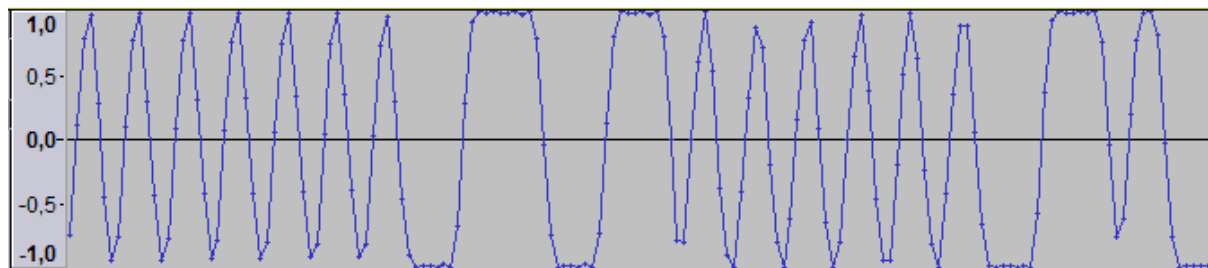
For more information, please see Heise c't 5/88, page 116 "Draht zum großen Bruder" and "Ergänzungen und Berichtigungen" in c't 5/88, page 116 or see the article by Norbert Unterberg "SHARP <--> PC" chapter 5.2.

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



The number of signal 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 wav file of the PC-1600 with CE-1600P or PC-E- and -G-series



There are only signal waves allowed 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.

With the older series you should NOT override the recording in such a way that the amplitudes constantly hit both sides and (especially with the PC-1500) form double peaks.

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 cannot fully convert and will report errors.

### 3.2.2 What should the recorded wav file not look like?

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

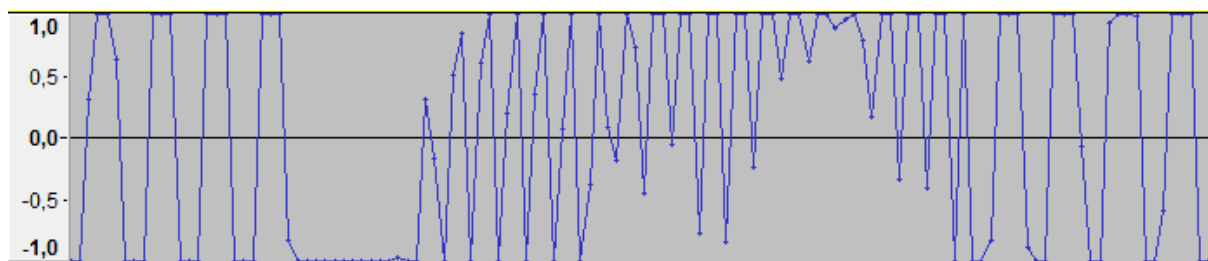
Unfortunately, it sometimes happens that through system activities e.g. 10 ms of the recording have been cut out or replaced by dummy data (last picture). Do not waste too much time debugging, but try a new recording first.

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

Shown below are some typical errors.

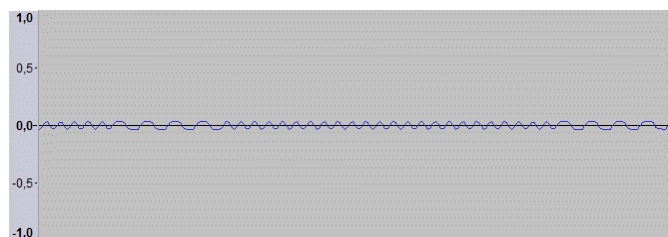
The first picture is a tape recording. Here you have to pull up the missing 4 waves in the gap in the front part with an audio editor and pull down at least one amplitude on the right.

*Dropouts (PC-1245-1475)*



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

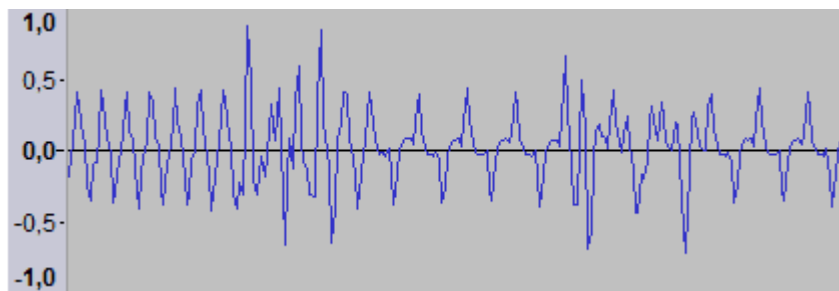
*Low Level (PC-1500): The signal is very sensitive to disturbances.*



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.*



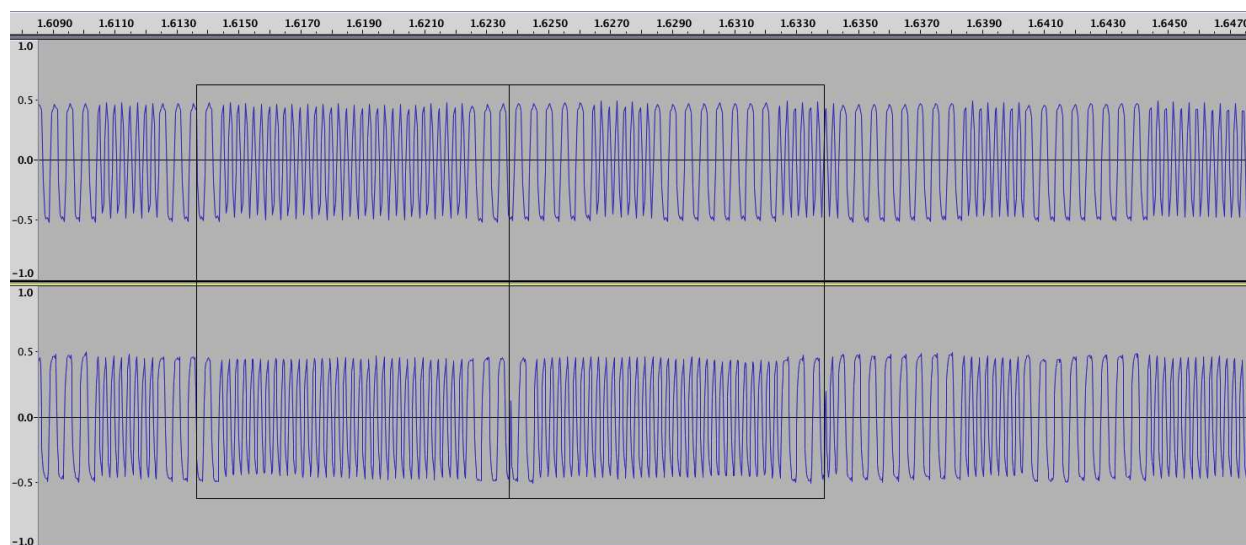
Cause: Induction from other cable to the audio cable (internal power supply cable)

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 signals can be detected by spikes in the curve or incorrect signal wave numbers.

*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 wav 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) if they are relatively loud. The same problem arises if the preceding silent period of PC-16/E/G-series is overlaid with a disturbing frequency. You should no longer cut this introduction for the current version of Wav2bin, but use the --start parameter in the event of errors.

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 wav editing was tested with SoX and Audacity.

Do not convert the sample rate! If for some reason you need to do this or need to cut the recording, you should use a recording with a sample rate of 48 kHz or higher.



Try to use the device specific filter rules:

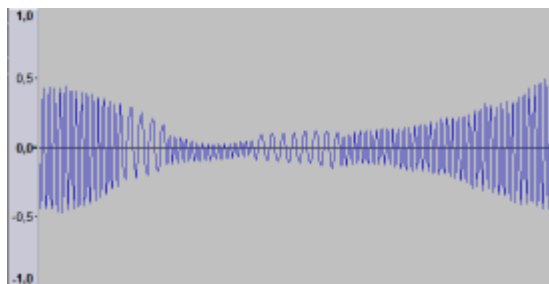
1. For sound via system from emulator also set `--device=EMU` and `cspeed`. For sound recorded via microphone from air use a silent room, avoid near (about 5 cm) reflections (for example a display), set `--device=AIR` and `--level 0x800`.
2. For recordings direct from cassette recorder (only with fluctuating DC bias) `--device=CAS` or `--device=CS` may be helpful.

### 3.3.1 Recommended procedure for digitizing tape recordings

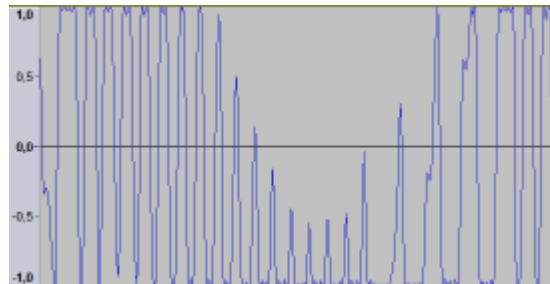
If tape material is faulty, digitizing can be time consuming or it can become impossible to completely recover the data. Always use Wav2bin with the **pc parameter** and the parameters described below. If an error occurs, go to the next step. The following order is recommended.

1. If no synchronization is found at all (error 9), then use the `--start` parameter, see below.
2. At first only use the `pc` parameter but not any device-specific device parameters
3. 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`).
4. 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.e. with type-specific default values.

5. Use the parameters `=CS0` to `CS4` or `=CAS0` to `CAS4` one after the other
6. If the amplitudes of the recording are permanently up to the stop on both sides, the recording may have been overridden and shows double peaks, then you can completely switch off the processing of amplitudes for the 1234 and 1500 series with `--level=0x2000` as a test.
7. Use a hi-fi tape deck instead of a tape recorder as the player
8. If you constantly get errors from a tape, use `--level=0xC0`, then `0x400` to get the signal wav time of the error, and fix the amplitudes with an audio editor. With `0x800` you convert it with checksum errors. Or 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,	Quick-Tape PC-1500: about 5000 Hz
All others PCs	4000 Hz	

If the frequency deviates by more than 5%, 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, then it is now recommended to use the `"start"` parameter. The given time (in seconds.tenth-seconds) should point into the area of the synchronisation tone, whereby a distance of about 0.5 seconds to the usable data must be maintained. The beginning of the usable data can be seen in the audio editor that two tones are used from this time instead of one frequency.

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

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

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.  
Never connect the original small black plug (for the remote control) to 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 the previously generated wav file of a very short, e.g. 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-1211, PC-E500S and PC-G850 series may need a higher or even the maximum 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. Check that no faders, mixers or filters are active in your media player and that only the signal from the wav file is passed through unchanged to the audio output. If the end of the transfer is not recognized by the pocket computer despite the high volume, then test the wav file in Audacity and play it back there.  
If the transfer works with Audacity and the cause is only an active fader, you can work around this by artificially lengthening the wav file with Bin2wav -l 0x100.  
Otherwise you should switch to a simpler and more precise audio player for data transfer.

Creating a hard file end with -l 0x2000 and appending file ends of successful recordings (via SoX) can make a media player play back true to the original, but this is very experimental.

6. Some audio interfaces invert (mirror out of phase) the audio signal via other SHARP interfaces, which does not work with certain data formats of the last series of pocket computers without adaptation. From version 2.1.1 onwards, the phase position is calculated by Bin2wav using the PC and DEV parameters, but can also be switched with --device=INV.  
If the interface has become too quiet (e.g. CE-121 due to aging), you can increase the level for the PC-1234 series with the parameter --device=MAX or INX (INV+maX).
7. If the problem is that the transfer is progressing well with the older series, but the end of the transfer is not recognized, you can also test it with the parameters -dSTD0 to -dSTD2, -dINV0 to -dINV2, -dMAX0 to -dMAX2 or -dINX0 to -dINX2 for the older series.  
This problem particularly affects the first models of the PC-1400 series when the combination of audio player, sound system and cassette interface generates a signal at the end of the transmission which the Pocket Computer interprets as a start bit for further transmission.

If the PC-1600, -E500, and -G800 series do not see an error until the end, the problem is likely due to other causes. Because with these models the checksum is only compared at the end of the block, all other errors can be the cause.

If the transmission hangs for a long time and does not finish, the level is usually too low.

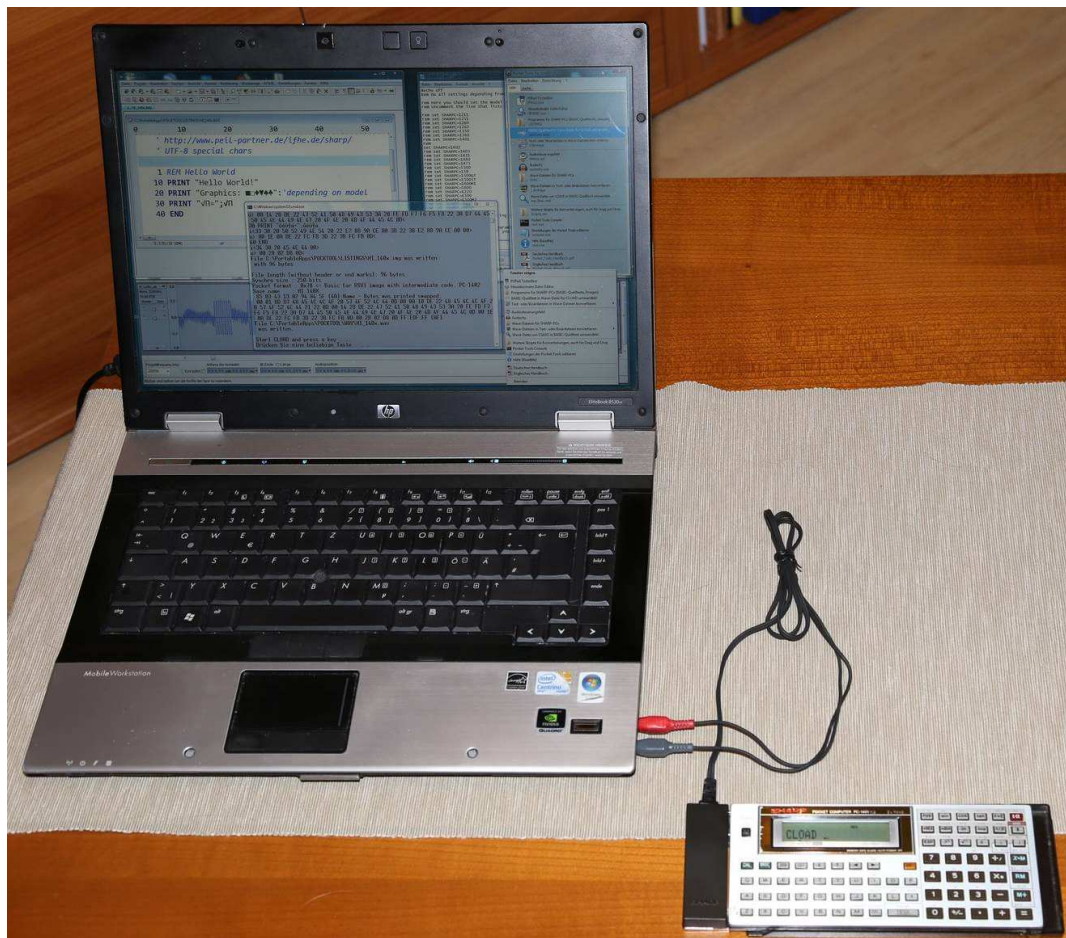
8. After performing a short transfer in the previous step, optimize the settings by transferring a longer BASIC program. Make a note of the setting values of both volume controls after successful transmission.

9. 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.
10. Starting with version 2.1 of Bin2wav, the **standard sample frequencies** and the effect of the -l2 parameter have been changed. By default (**without -l2**) a sample rate of 44.1 kHz is now used for the generated wav file of the PC-1500 and **48 kHz** of the other pocket computers. This change was necessary because the lower sample rates were poorly supported by the PC hardware.  
If you haven't had any success so far, but it worked earlier, then test everything with the options -l2 or -l1 (small "L", for scripts with the parameters: ? -l2). With -l2 the wav files of Bin2wav 2.0 are now generated: lower sample rate and trapezoidal waveform.

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 options in the BIN2OPT variable.

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

1. Set for transfer in the file Sharpset.bat: set SHARPC=1402, set SHARPCDEV=124
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 and EL-5400 to EL-5520 series.  
This includes Tandy TRS-80 SHARP-based models PC-1, PC-2, PC-3, and PC-8.
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,  
database data, 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 variables 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: 'P.' for PRINT
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 characters 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 wav audio files without any conversion or finishing, writes different waveforms  
(apply it hardware dependent)
11. Supports wav- and tap-files from emulators and wav files from pocket computers with hardware  
speedup
12. Reads and writes Quick-Tape formats with Quick-Tape or standard headers of the PC-1500,  
writes SuperTape wav files (but does no read directly, only by using "SuperTape for Windows" )
13. Automatic correction of many disturbances in wav audio files and named filters for recording from  
cassette recorder and microphone, all checksums are checked for 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 to build  
Pocket Tools
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 are supported
18. Continuous debug traces and more special options exist, see the help screens  
see Wav2bin\_Debugging\_Hints.txt for improved low level debugging

#### 4.1. Technical limitations or not implemented

1. During the run of a BASIC program on a PC-E500(S) series, the PC-E500(S) changes the jump targets in the source file image from line numbers to absolute positions. Read "HowTo" chap. 15.5 on how to get a reusable BAS file by using Wav2bin twice from a wav file that contains a runtime-optimized BASIC image of a PC-E500(S). With Windows you can use WavEbas.cmd for this. Bas2img fully supports the E500 series intermediate code since Pocket Tools 2.1.
2. Support for MZ series (700) is experimental 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. The Quick-Tape version with standard header of the PC-1500 does not support any special flags.
4. SuperTape is supported by Bin2wav, but not by Wav2bin directly. "SuperTape for Win32" is recommended to convert a wav 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, Heikel FSL, TOS.
6. No stream processing: wav2bin needs the complete wav file.
7. For the pocket computer a conversion by Bin2wav is completely reversible after a faultless conversion 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. If you want to use the "CLOAD ?" command, you should not generate the wav file used for this comparison from a source text. Use the image that was created directly from the wav file that was read by this pocket computer.
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. The installation path for the Pocket Tools must not contain any spaces or special characters.
10. 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 of the 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.
11. 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 (see script ihx2wav.)
12. The longer sync signals generated by Bin2wav are shorter than the original. At the beginning of the transmission, between data blocks and especially with the PC-1211, they are reduced to the minimum length necessary for secure transmission in order to enable faster transmission. Use the -l 0x400 parameter and, if necessary, the --sync parameter to adapt the lengths to the original signals for comparison purposes.

Even then, the generated audio signal is not exactly the same on the wave. For the PC-1211, the last stop bits per quarter are rounded up to the bit length. In certain constellations, additional stop bits can also be generated at the end of the transmission. In addition, the playback speed will differ slightly from an original recording. None of this has any functional effect.

## 4.2. Limitations on reading and playing wav files

1. You must use PC and audio hardware that does not suffer from crackling or rustling noises. 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 should consider using different hardware for the Pocket Tools.
2. To play the wav files created with Bin2wav, you must use an audio player that plays the content directly and unchanged (bit-accurate) at the output (e.g. Audacity). No faders may be integrated that pull the signal level down at the end. In addition, this application must not contain any mixers and filters that mix in the signal from other applications or change the signal. A simple player (e.g. VUPlayer) is much more likely to succeed than a feature-overloaded audio player.
3. By default, uncorrectable errors in recordings 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.
4. 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-1100 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 wav file is not readable, then please note the timestamp 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" wav files, where only a few milliseconds (inside or just after the end) are missing, cannot be read.

## 5. Handled file types

- wav file**      RIFF wav file, mono or up to four channels (first channel 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 (CS)  
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, mostly with errors)  
The wav file should be recorded using an audio editor with a sample frequency about 22050 Hz and 16-bit. If more editing is required, and for Fast-/Quick-/SuperTape please use 48000 Hz (or 44100 Hz).  
The wav files made by Bin2wav are aligned and therefore the frequency can be lower.  
A wav file could be normalized and minor errors can be 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 wav file must be a "public readable type" (with fewer chunks).
- tap file**      Raw binary format of quaters (PC-1500) or bytes (all other PCs), represents the content of the wav 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 it is impossible to make a wav 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.  
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.

<b>img file</b> (type <b>txt</b> )	<p>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 TEXT and CSAVE.</p> <p>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 in ASCII format.</p> <p>This format is supported with PC-1360, 1475, all older pocket computers with a serial interface and the PC-E500 series. It is <u>not a recommended format for BASIC</u>, but it is for source texts of assembler and C programs if they cannot be transmitted completely in ASCII format with a pocket computer via the audio interface (PC-G850V and others).</p> <p>No ID, no name, no file headers and no checksums are included in this files type.</p>
<b>casl img</b> <b>capx img</b>	<p>Binary byte format of Wav2bin, Bas2img (and Bin2wav), which represents the internal assembler source code format of the pocket computers PC-1416/17G and PC-1440/45. It is similar to the "txt" format but is not compatible with it.</p>
<b>rsv file</b> (or in <b>img</b> )	<p>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.</p> <p>No ID, no name, no file headers, no file end and no checksums are included in this files type.</p> <p>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.</p>
<b>def file</b>	<p>Def key image, PC-1500 V2 or 1500A (used in software, eg. PC-BASIC 84, PC-WORK)</p>
<b>dat file</b>	<p>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 #.</p> <p>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.</p> <p>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. It may be necessary to specify the type of conversion with an -l option.</p> <p>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).</p>
<b>dim,</b> <b>mem file</b>	<p>Image of all dimensioned variables from the PC-1500 with Quick-Tape, Image of all databases and dimensioned variables on the PC-1140,-1150,-124xDB</p>
<b>tel,scd,not,</b> <b>crd file</b>	<p>Image of a single database (phone, scheduler, notes) on the PC-1100, Image of the entire RAM card with programs and data(bases) of the PC-1100</p>
<b>bin file</b>	<p>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 For images with BASIC intermediate code, please use the other file type "img".</p> <p>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.</p> <p>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.</p> <p>For PC-E500 note the following regarding the switch -dINV.</p> <p>For ihx files use Hex2bin to convert into a bin file first.</p>

<b>ihx file hex file</b>	<p>ASCII data formats for the storage and transmission of binary images, mostly in hexadecimal notation. These formats are not supported directly.</p> <p>Additional software is needed to convert these files into "raw" BIN files as well as onward.</p>
<b>asc file</b>	<p>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</p> <p>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.</p> <p>Also, some of PC-G series can write but not read this format from CAS: correctly. For use with INPUT #n this is <u>not recommended for G850 series</u></p> <p>No ID, no name, no file headers, no fill bytes, no end mark, and no checksums are included in this file type.</p> <p>The format is similar to bas, but there are differences in the header.</p>
<b>bas file</b>	<p>ASCII source file format (not binary) 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</p> <p>SAVE "CAS:filename" (,A with PC-1600) was used.</p> <p>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 may not understand it.</p> <p>The user manual of PC-1600 points to a similar problem at page 6-24 (4).</p> <p>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.</p> <p>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).</p>
<b>asm file c file casl) file</b>	<p>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.</p> <p>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</p> <p>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.</p> <p>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 PIC assembler only the file type "asm" should be used and "casl" (capx) for CASL (CAP-X).</p> <p>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.</p>
<b>sha file</b>	<p>ASCII Text format of the Software "Transfile PC plus" or an OEM version, Copyright Yellow Computing or licensed partners</p> <p>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.</p> <p>The type, the name and all header bytes are included in this file first text line of a block, but no checksums. Possibly ASCII end marks are included.</p> <p>This type of format is very similar to bas files.</p> <p>Special chars are encoded for DOS with CP437 only.</p>

- st file** Binary image format of the open source and cross-system software "SuperTape" from the computer magazine c't of the publisher Heise.  
The name, addresses and more information of the SuperTape header are included in this file. The image follows the header (without end marks or checksums).
- This type of 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 type of 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 type of 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 type of file is required for the commands of some hardware extensions of PC-1500.
- This type of file was introduced by Eric Millesamps, 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 type of 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.

For Windows, scripts and from Pocket Tools Version 2.1 also a start menu are included.  
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	WAV 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>i16</td><td>Image of PC-1600 with integrated header of the disk file</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>var</td><td>"dat" without a file header (for appending to a wav data file)</td></tr><tr><td>dim</td><td>Block of all DIMensioned data (PC-1500 Quick-Tape)</td></tr><tr><td>mem</td><td>All dimensioned data, tables and DB of PC-1150/40</td></tr><tr><td>crd,</td><td><u>tel, scd, not</u>: RAM card or special databases of PC-1100</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><u>asm5, asm6, asm8, casl, capx</u>: Assembler or C source text</td></tr></table>	img	BASIC-program binary image (default)	i16	Image of PC-1600 with integrated header of the disk file	bin	Binary mc of assembly program or data	dat	Data variable blocks (binary data)	var	"dat" without a file header (for appending to a wav data file)	dim	Block of all DIMensioned data (PC-1500 Quick-Tape)	mem	All dimensioned data, tables and DB of PC-1150/40	crd,	<u>tel, scd, not</u> : RAM card or special databases of PC-1100	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,	<u>asm5, asm6, asm8, casl, capx</u> : Assembler or C source text
img	BASIC-program binary image (default)																																
i16	Image of PC-1600 with integrated header of the disk file																																
bin	Binary mc of assembly program or data																																
dat	Data variable blocks (binary data)																																
var	"dat" without a file header (for appending to a wav data file)																																
dim	Block of all DIMensioned data (PC-1500 Quick-Tape)																																
mem	All dimensioned data, tables and DB of PC-1150/40																																
crd,	<u>tel, scd, not</u> : RAM card or special databases of PC-1100																																
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,	<u>asm5, asm6, asm8, casl, capx</u> : Assembler or C source text																																
-p, --pc=NUMBER	Model number of SHARP pocket computer, currently available: 1100, 1150, 1211, 1245, 1251, 1261, 1280, 1350, 1360, 1401, 1402, 1403, 1421, 1445, 1450, 1460, 1475, 1600, 1600M1, E500, E220, G850, G850V, 1500ST (SuperTape), 1500QT, 1500QT1 (Quick-Tape) and other (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, E500:0xFFFFFFFF (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 wav file)																																
--version	Display version information																																
--help	Display this information																																
--help=l	show level option screen																																
-d, --device=TYPE	INV, INX: interface with inverting level converter (mirrored to zero line) MAX, INX, 121, 125: Raising the level for the output signal No subsiding signal at the end of transmission, but: <u>std0, inv0, max0, inx0</u> : silence, <u>std1...inx1</u> : level high, <u>std2...inx2</u> : sync bits																																
-e, --endmark	Ignore last byte of Img/Txt, Bas/ Asc, if BAS/ASC-EOF mark included																																
-l, --level=VALUE	Option bits and Print debug traces (2-times usable) a (hexadecimal) integer (0x____) or sum of it:																																

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

Waveform and frequency, default sample rate (changed): 48 or 44.1 kHz

- 0 Wav file with 48 kHz (PC-1500 formats: 44.1), near rectangle for sound chips that process other frequencies incompatible
- 1 Force low sample rate with triangle waveform for base frequency (old compact format of Bin2wav 1)
- 2 Sample rate of Bin2wav 2.0 (trapezoidal waveform, 4x base freq.)
- 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
- 0x2000 Hard end of transmission without a subsiding signal, to add a player-specific WAV end
- 0x4000 Write no file header (VAR), have to merge dat wav files manually
- 0x8000 Data variable block (IMG) is from Wav2Bin 1.5 or version before
- 0x800 Write also, if i.a. checksum bug will be activated (not readable)
- 0x400 Write long synchronisation like the original measure
- 0x200 Write long synchronisation like Tech. Ref. Manual
- 0x100 Additional long synchronization tone after the end of transfer
- 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 generatable sample frequencies (changed in PocketTools 2.1)

Option: level=	1	2	<b>(0)</b>	3
PC-1210 – PC-1475:	8 kHz, 16 kHz,	<b>48 kHz</b>		
PC-1600, E/G series:	6 kHz, 12 kHz,	<b>48 kHz,</b>	16 kHz	
PC-1500:	5 kHz, 10 kHz,	<b>44,1 kHz,</b>	16 kHz	
Quick-Tape	10 kHz, 20 kHz,	<b>44,1 kHz,</b>	16 kHz	
SuperTape	22.050 Hz, 36 kHz,	<b>44,1 kHz,</b>	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 with absolutely no errors.

With older versions the option -l2 had to be set in order to set a sample rate of 48 kHz or 44.1 kHz, which would achieve the opposite from Pocket Tools version 2.1. The wav 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 sometimes lead to 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	WAV file, PCM, possible is a sample rate of 11025 (only PC-1500), but recommended is <u>mono, 16-bit, normalized with a sample rate of 22050</u> (recommended for all PCs) to 192000 or original wav 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 bas BASIC-program text file, line by line (default) asm, <u>asm5, asm6, asm8, casl, capx, c</u> : 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 also for other binary data (PC-1100 <u>crd, tel, scd, not...</u> ) bin machine code or other binary data, dat BASIC variable data (special binary data of PC-1100...2500) dim block of all DIMensioned data (Quick-Tape) mem block of all tables and variables (PC-1150/40,-DB) 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 wav 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" no to ASCII transcription with brackets [reversible] yes to UTF-8 (default) 2dos to DOS-US 2esc to ASCII with escaped special chars 2asc to ASCII transcription for later serial transfer
-w, --width=VALUE[_][_]	(1-st use) Minimum width of line number (only for BASIC-program text) 2 to 10 (default: 5 characters ) for text format, beginning/end of the line 0 indentation off for pure serial transfer
=[VALUE]: or ; , . !	Colon: Line number ends with colon, {;} col+spc, {,} spc+col+spc, {.} no separation, {!} del line number, for end of line and more see chapter 6.4.4
-w, --width=VALUE[_]	(2-nd use) Level of spaces to insert before commands (for BASIC-program text) 0 to 8 (default: level 5 ) for text format, within the line
=[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)]

**-s, --start=TIME** in seconds, default parameter: 0.0 (processing from the beginning)  
instead of the automatic search for the beginning of the synchronisation tone

2<sup>nd</sup> time: see file Wav2bin\_Debugging\_Hints.txt

**-s, --start=debug\_time** Debug window for debugging on sample, amplitude and bit level

**-iOFFSET, --img=** Source file: IMAge, ST or SHC format, no wav, Start offset in bytes: first=0

**--tap** Source file: Emulator tap format (with header and checksums), no wav

**--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 wav 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 + wav 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 =rawdats (quarters swapped)

Device specific filters: **--device=CAS:** (CS) recording from tape with unstable signal, see chap. 6.4.3  
**--device=EMU:** Wav from emulator via system sound, set --cspeed and -pc  
**--device=BIN :** Wav 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, E500, E220, G850, G850V and more
- 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 for PC-1500 PC-MACRO, PC-E500 Dump Tool  
asm6 for PC-1600 assembler
  - asm8 PC-G8/E2 Z80 assembler sources, asm for PIC assembler
  - casl, capx for CASL or CAP-X assembler sources
  - c C sources of PC-G850 or (note: different format!) of PC-G850V(S)
- 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)  
Syntax: see enclosed cfg files
- k, --keywords=** Token file with tokens of hardware options, 3-times usable  
List of tokens, format of one line: Tokenstring=HHHH  
Syntax: see enclosed key/cfg files
- 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
- auto(=STEP)** : (default=10) Replace missing line numbers according to an AUTO instruction
- a** 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
  - : 0x400 Don't delete (empty) lines with at least one space
  - : 0x800 Depress some line errors, result may not editable
  - : 0x1000 Switch conversion of E/G „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, unless the parameter Bas2img -l 0x400 was used and they also contain blanks.  
 Bas2img should also be used to convert from general BAS-files to Sharp-specific ASC-files for device CAS: (of the pocket computer) 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 default 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	1100, 1260 - 1262, 1350, 1401/02, 1430/31, 1450	(1421 tokens separately)
Group NEW3	1403, 1416G/17G, 1425, 1440/45, 1460	including group NEW tokens
Group EXT	1280, 1360, 1470U, 1475	
Group E	E500, E500S and successor, 1480U, 1490U, U6000	
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.

In practice, this is a rare problem. However, if such long variable names have been used, 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 cannot be sufficiently determined on the basis of the content of the source file alone. Bin2wav implicitly switches some types according to the file extension.

### 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 signal 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 signal 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 wav files created by Bin2wav or other synthetic wav files are to be back-converted. Analog errors do not have to be compensated.
- Device=EMU      This option should be used for wav 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 wav 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, INX      This parameter for Bin2wav mirrors the signal at the zero line (DC) and can be necessary to ASCII formats via inverting cassette interfaces.
- Device=MAX, 125...      These parameters increase the signal strength for CE- 125, 121, 122 and switch the phase position: 120, 123, 124, 126, 129, 150, 162, ecps, mfe, nos, ati
- Device=STD0, STD1, STD2, INV0...2, MAX0...2, INX0...2: Instead of the decreasing signal at the end of the transmission, silence (0), high level (1) or sync bits (2) are output.

### 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(.) (2.use)      0-8, level of spaces before commands, '.' deletes them behind commands  
Example: Much spaces: -w10, -w8      No blank as possible: -w0. -w0.

Endmark                      Used for memory images, not for transfer to tap files or wav files.  
                                  If the -e option was used for an image, then  
                                  it should be applied to all tools for this image.

#### Other parameters

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 of Pocket Tools Version 1 has been reactivated for reasons of compatibility and does not support any new parameters. Only use this with legacy software, that requires this format!  
 The old parameters are mapped internally to new parameters.

**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 wav file, nibbles in bytes or other Error when compiling a (decimal) number (Bas2img for PC-E500)
ERR 3	Arguments problem, for example: pocket not implemented
ERR 4	Error with line numbers (BASIC)
ERR 5	File I-O error or file used by another software (e.g. the media player)
ERR 6	Image line too long or other buffer overflow
ERR 7	False or unknown format of wav 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 wav file or unexpected lost
ERR 10 (and higher)	Multiple errors found: Error number = last error + 10
<b>Exit codes</b>	<b>from wav2bin --exit=type</b> 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
TYPE 110	PC-1150/40, 1248/46DB, MEM: all database tables and dim. variables
97, 98, 100, 106	PC-1100 phone book, scheduler, notes PC-1100 RAM card
TYPE 122	PC-1416G/17G, PC-1440/45 assembler: CAP-X / CASL
<b>Exit codes</b>	<b>from wav2bin --exit=ident</b> 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, 1140, 1150

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 data CSAVE M / CLOAD M no binary for 4-bit CPU available (PC-1246-1248)
0x6E	All dim. variable and database data, one image block, 4-bit addresses CSAVE MEM
Shc	Supported

### PC-1260-1262, 1280, 1350, 1360, 1401-1475, 1100

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 data CSAVE M / CLOAD M no binary for 4-bit CPU is available (PC-1430/31)
0x7A	PC-144x, 141xG assembler CASL or CAP-X
0x6A	PC-1100 OUT B/P full RAM card
0x61,62,64	PC-1100 OUT Databases: Telephone, Scheduler, Notes
Shc	Supported

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

Id	Sub-Id	
0x A 0		PC-1500 binary, assembly program or other data, 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 data, 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 data, 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, G850V

Default CAS: I/O parameter only!

Id	Sub-Id	
0x 1	0	PC-16/E/G binary, assembly program or data, CSAVE M / CLOAD M
0x 2	1	PC-G800, Basic image, G850V(S) also for C and assembler



		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.
<b>PC-1500 Quick-Tape</b> (--pc=1500QT)		
	Id Sub-Id	(--pc=1500QT1)
	0x A 5	PC-1500, Basic image QSAVE old version with standard (0xA1) header
	0x A 6	PC-1500 binary, assembly program or data, QSAVE M old version with standard (0xA0) header, without special flags (--pc=1500QT4)
	0x 0x42 0AA0	PC-1500, Basic image QSAVE / QLOAD new version with Quick-Tape header and blocks with standard length
	0x 0x4D 0AA0	PC-1500 binary, assembly program or data, QSAVE M / QLOAD M add for flags to start address ME1: 0x40000, PV1: 0x20000, PU1: 0x10000
	0x 0x52 0AA0	PC-1500 format RSV image QSAVE R / QLOAD R
	0x 0x44 0AA0	PC-1500 format DIM image (all dimensioned variables together) QSAVE D / QLOAD D
	0x 0x56 0AA0	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
<b>PC-1500 SuperTape</b> (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= <b>1500ST</b> ) (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 drive or simply copied to a folder at your hard disk to start from there. Also read the Installation\_for\_Windows.txt file.  
On hard drives, the subfolder POCKTOOL of the folder \PortableApps\ is recommended.  
The installation path should not contain any spaces or special characters!

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 and available for download 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 working with UTF-8 files and for cmd files, it is also useful to use Notepad ++ editor.

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 Pocket 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 your 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 wav file containing an image of the PC-E500 to a source file.

## Notes

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

In the file POCKTOOL\Scripts.win\\_ReadMe.txt all scripts are listed with their purpose. 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.  
Some scripts (for MC) also use an .ADR (.CAL) file or the newer parameter file (.CFG).

## 10.2. Using Pocket Tools with Windows

Scripts for "Windows" (NT4, tested with 10) are included., see \_ReadMe.txt, Tools Console "phelp".  
The following options are available for using these 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.

## 10.3. Installation instructions for other operating systems

For Linux, you can use the enclosed scripts from Bernhard and thus also compile the source code.  
Most Linux users will prefer to use the console.  
Also for MacOS or Android, you need to compile the source code and use alternative wrappers.

## 11. Use of the Tools, chain and examples

To 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> > recommended: --type=DAT(img) --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> > recommended: --type=BIN (img) -m --pc= <i>Number</i> <u>Note the start (and entry) address</u> or use parameter -m or write it to filename.ADR and filename.CAL with scripts.

BIN image file into a wav file (CLOAD M)	bin2wav <i>BinFile</i> < <i>WavFile</i> >--pc= <i>Number</i> --type=BIN recommended: --addr= <i>StartAddress</i> optional 1500/1600/E500 2nd.address : --addr= <i>EntryAddress</i> optional: --name= <i>SharpName</i> recommended (PC-E500): --device=INV/std (depends on interface model)
Wav file into a SHC image file (CSAVE (M) / PRINT#)	wav2bin <i>WavFile</i> < <i>ShcFile</i> > recommended: --type=SHC --pc= <i>Number</i>
SHC image file into a wav file (CLOAD (M) / INPUT#)	bin2wav <i>ShcFile</i> < <i>WavFile</i> >--pc= <i>Number</i> --type=SHC optional: --name= <i>SharpName</i> --addr= <i>StartAddress</i>
Wav file into a SHA source text (CSAVE)	wav2bin <i>WavFile</i> < <i>ShaBasFile</i> > recommended: --type=SHA --pc= <i>Number</i>
Wav file into a BASIC image PC-1600 - SAVE (CAS: without ,A)	wav2bin <i>WavFile</i> < <i>ImgFile</i> > --type=IMG recommended: --pc= <i>Number</i>
Wav file into a BASIC source text, Text Editor Cmt-SAVE (CAS:,A) for PC-G/E/1600	wav2bin <i>WavFile</i> < <i>AscFile</i> > recommended: --type=ASC --pc= <i>Number</i>
BASIC source into Wav file Text Editor Cmt - LOAD (CAS:) for PC- PC-G/E/1600	1. bas2img <i>SrcFile</i> < <i>AscFile</i> > --type=ASC/bas --pc= <i>Number</i> 2. bin2wav <i>AscFile</i> < <i>WavFile</i> >--type=BAS --pc= <i>Number</i> optional: --name= <i>SharpName</i> recommended (PC-E500 series): --device=INV/std (depends on interface model)
Wav file into a Source text from Text Mode CSAVE	wav2bin <i>WavFile</i> < <i>BasFile</i> > see "Wav file into a BASIC source text" (just like with intermediate code)
BASIC source text into Wav file Text Mode – CLOAD for 1280/1350-60/1450-75/(PC-E)	1. bas2img <i>SrcFile</i> < <i>TxtImgFile</i> >--type=TXT --pc= <i>Number</i> 2. bin2wav <i>TxtImgFile</i> < <i>WavFile</i> >--type=IMG/txt --pc= <i>Number</i> optional: --name= <i>SharpName</i>
Wav file into ASCII data (OPEN CAS:, PRINT #n)	wav2bin <i>WavFile</i> < <i>AscDataFile</i> > recommended: --type=ASC --pc= <i>Number</i>

<p>ASCII Data text into Wav file</p> <p>(OPEN CAS:, INPUT #n)</p> <p><u>Problems described above:</u> chap. 3.4(6), 5 asc</p>	<p>bin2wav AscDataFile &lt;WavFile&gt;</p> <p>                                --pc=Number                                 --type=ASC optional:                    --name=DataName</p> <p>recommended (PC-E500 series):</p> <p>                                --device=INV/std (depends on interface model) (mandatory if I/O error always occur after the 2nd block without this parameter)</p>
<p>Wav file into a binary raw tap image file (no check of plausibility or correctness)</p>	<p>wav2bin WavFile &lt;tapFile&gt;</p> <p>                                --type=TAP</p> <p>recommended:                --pc=Number</p>
<p>tap file into Dstfile</p>	<p>wav2bin tapFile &lt;DstFile&gt;</p> <p>                                --tap</p> <p>all other parameter:        see wav file into DstFile</p>
<p>Image File into tap file (Bin2tap)</p>	<p>bin2wav File &lt;tapFile&gt;</p> <p>                                --tap</p> <p>all other parameter:        see File into wav file</p>
<p>Image File into Basic text file</p>	<p>wav2bin ImgFile &lt;basFile&gt;</p> <p>                                --img --pc=Number</p>
<p>SHC-File PC-1234 without password into a Basic text file</p>	<p>wav2bin ShcFile &lt;basFile&gt;</p> <p>                                --img=9 --pc=Number</p>
<p>ST-File of SuperTape for Windows into a Basic text file</p>	<p>wav2bin StFile &lt;basFile&gt;</p> <p>                                --img=26 --pc=Number</p>
<p>BASIC source file for PC-1500 into a wav file with SuperTape format (Quick-Tape similar, but --pc=1500QT)</p>	<p>1.                            bas2img SrcFile &lt;ImgFile&gt;                                 --pc=1500 (also 1500ST acceptable)</p> <p>2.                            bin2wav ImgFile &lt;WavFile&gt; --pc=1500ST</p> <p>optional:                    --name=SharpName</p>
<p>ASM source file to Image for PC-Macro</p>	<p>bas2img SrcFile &lt;ImgFile&gt; --pc=Number --type=asm5 &lt;--auto=5&gt; (format macro assembler via BAS cload)</p>
<p>BASIC source file for ASCII serial transfer</p>	<p>bas2img SrcFile &lt;AscFile&gt; --pc=Number --endmark --type=ASC (format text for transmission via serial interface)</p>
<p>BASIC source file into Emulator Image</p>	<p>bas2img SrcFile &lt;ImgFile&gt; --pc=Number --endmark (copy it into the RAM of the emulator with end mark)</p>
<p>Recorded wav file into clean synthetic wav file</p>	<p>1. wav2bin WavFile &lt;File&gt;</p> <p>                                --type=IMG --parameters &lt;=ConfigFile&gt; &lt; --pc=Number&gt;</p> <p>2. bin2wav File &lt;WavFile&gt; --parameters &lt;=ConfigFile&gt; (only for formats that are supported not only by wav2bin but also by bin2wav)</p>
<p>More variants exist.</p>	<p>See Pocket Tools Console "PHelp" (in the file folder Scripts.win\_ReadMe.txt) Take a look at the use of the parameters in the scripts and use them for your commands.</p>

## 12. Default start addresses for (loading of) binary code

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

PC number	Default load address (hex)	Before entering "CLOAD M" you have to 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 PC series are supported as well as the rebranded series EL-54xx to EL-55xx and TRS 80 PC-1 to PC-3 and PC-8. The following Pocket Computers were tested with the Pocket Tools.

PC-1211, PC-1212 with CE-121, CE-122 full volume

PC-1100, PC-1150, PC-1245, PC-1248, PC-1251, PC-1260, PC-1262  
with CE-126P, CE-123P, CE-125 full volume

PC-1280, PC-1350, PC-1360, PC-1401, PC-1402, PC-1403, PC-1403H, PC-1416G, PC-1421, PC-1425, PC-1445, PC-1450, PC-1460, PC-1470U, PC-1475  
with CE-126P, CE-129P, CE-124, other, also 3<sup>rd</sup>-party,

PC-1500, PC-1500A,

PC-1500 Quick-Tape (Hardware: IWS interface with EPROM of ECPS, Software: RWE tool 4.2), CE-150

PC-1500 SuperTape, with CE-150 and CE-162E

PC-1600 with CE-1600P and PC-1600 mode 1 with CE-150

PC-E500, PC-E500S with CE-126P, also 3<sup>rd</sup>-party,

PC-E550 tested from emulator only

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

PC-G850VS tested with CE-126P full volume (and from emulator)



### 13.1. Naming convention for Sharp Pocket Computers with special interface

Remove the prefix "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 one way)
1500QT	PC-1500 with Quick-Tape (hard- or software, e.g. PC-WORK)
1500QT1	PC-1500 with Quick-Tape (software with PC-1500 standard header)
1234	Unknown PC with 4 kHz base frequency (1100-1475, 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 for BASIC

### 14.1. Important abbreviations for the source code with Bas2img

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

The abbreviations are based on those documented in the SHARP user manuals.  
Most commands can be abbreviated in various ways.



## 14.2. Special characters, partially usable 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]

UTF-8 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 characters by Bas2img

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 in the codepoint file 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 UTF-8 characters as well as variable or fixed tags that have been entered, copied from a codepoint file, or generated by Wav2bin.

### Preprocessor (line by line)

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 UTF-8 is turned on, the half-width katakana characters and most graphic characters of the internal tables become fixed tags as well as 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 (line by line)

13. Resolve orphaned 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 works with UTF-8 characters, use --utf8=yes (default). This way you can handle supported graphic characters directly. You can copy the supported UTF-8 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.

UTF-8 characters unsupported for your model generate bas2img warnings and invalid characters on the Pocket Computer.

When transferring data simply between different systems, transferring programs between different pocket computer generations, or using an 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 UTF-8 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 characters to codepage 437.

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

### 15.2. How can you scan and enter 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, e.g. to enter √ and π, see point 7 below. 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 Pocket Tools do not require any separation, see also point 8. If you only use jump labels and 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 these spaces. Only for the PC-E/G series most spaces are 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, WAV) 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. How to specify 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 entered as [HH] and will be replaced by Bin2wav in file names.

You can create tape file names with spaces in this way.

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

We recommend using the simple CLOAD and CSAVE commands for PC-E500 series.

The image of the program text created by BAS2img is now not only complete, but also the intermediate code in the current version is fully executable. After the transfer to the pocket computer, you no longer have to switch to TEXT mode and back to BASIC mode (but only if images are used that were created with older versions of Bas2img or possibly in the event of inexplicable errors).

The simple CSAVE format is much faster with the PC-E500 than the other formats and has a stable transmission performance. Alternatively, the following possibilities with other restrictions exist.

1. The format of older series (with Bin2wav -I 0x1000 ) for CLOAD@ is very slow  
(returnable from synthetic wav 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 wav 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.

The PC-E500 performs further runtime optimizations in the code itself during the first RUN.

When saving a file with CSAVE to a PC, for archival purposes you should always keep a (synthetic) wav file (alternatively IMG + CFG file), not just the BAS file. The wav file for this must be saved in an image by a tested, executable program via CSAVE and recorded. The synthetic wav file generated from this must then be checked with "CLOAD ?", see also chap. 15.11.

### 15.5. Convert PC-E500(S) to BASIC source files with Wav2bin

The PC-E500 series replaces line numbers after GOTO and other jump instructions 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 ensure that all text is checked and that intermediate code is generated if necessary. In addition, it still contains binary line numbers instead of binary jumps.
- B) Add a dummy line and delete it, e.g. 65279:END or edit the text anywhere just before the transfer.
- C) Working in two steps, as does the wavEbas script:  
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. Transferring 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 etc, 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. How to handle wav files from BASIC images containing binary code

1. If the lines are constructed properly and contain only characters that are allowed, an image can be 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, another 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. Editing 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, app 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          TOKEN 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 change 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. Using the very fast SuperTape format for PC-1500

Those who frequently transfer long programs to the PC-1500(A) will appreciate SuperTape.

1. Download c't SuperTape für Windows (ct.de/0410022), if you want to convert back wav 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), see the SuperTape\_ReadMe.txt file.

#### Transfer to the Pocket Computer (recommended)

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 wav files from images or Bin2wav --type=st --pc=1500ST to create SuperTape wav 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 and Pocket Tools 2.0).

6. Play the wav 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 (not recommended)

7. Save your program with the commands of your implementation and record the wav 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 option --img=26.

### 15.10. Using the Quick-Tape format for the PC-1500

1. Also read the Quick-Tape\_ReadMe.txt file. There exist different formats and versions of RWE Quick-Tape and their licensees (IWS, RVS PC-WORK, IWS Video Interface, ECPS, etc.). Not all formats are written. Fast tape formats from other manufacturers (Tramssoft etc.) that differ 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. You need a good 48 kHz recording just from the PC. The volume must be set precisely and slightly higher than is otherwise required for the PC-1500.
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-1245 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 (see Chap. 9, Sub-Id 0x4D, only supported for version with Quick-Tape-Header).

### 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 wav file
3. Convert the recorded wav file with wav2bin into an IMG (no Bas file) and then convert it to a synthetic wav file with bin2wav (see end of chapter 11)
4. Check the synthetic wav file created with bin2wav with CSAVE? "NAME"
5. Compress the synthetic wav 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 WAV file into a BAS file, indelible print on good paper and dry storage

With the synthetic (tidy and checked) wav file and a paper printout, you can restore your program any time later, regardless of the operating system and without Pocket Tools.

If your software contains machine code, necessary data, etc., then you should also save the default of 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. Improved 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, resulting in fewer errors, more precise start bit position every quater, fewer read errors for PC-1500 and others
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, especially for digitizing cassette recordings, many other devices for Bin2wav, eg. -d125, MAX.
6. Wav2bin type IMB (read image as a binary block) as an alternative to IMG (line by line), e.g. for basic programs that contain binary code
7. Writing in SuperTape format with limited support for Non-Sharp computer
8. Support for PC-1500 Quick-Tape, also with PC-1500 standard headers
9. Sharp file names with special characters
10. More unknown keywords and Japanese characters (reversible)
11. Priority of tokens switchable for BMC MC-12, Token for RVS PC-BASIC 84 and Ursoft
12. Using custom 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 V2 and PC-1500A
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 (Z80, PIC, CASL) 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-G850V
22. Complete support for the PC-G850 series, PC-G850V(S) separately
23. Support of CAP-X and CASL of PC-1445/40, PC-1417G/16G
24. Support of the database formats of PC-1100, PC-1150/40 (1248DB/46DB)
25. Improved compatibility with other compilers (Android Clang 10, Apple GCC, Linux, Windows)
26. Full image for the PC-E500(S) series with no need to go through TEXT mode
27. Better recognition of the synchronisation signal through Wav2bin and --start parameter
28. Improved signal forms in Bin2wav, different levels, variable design of the file end
29. Number of stop bits (PC-1234) and phase position (Bin2wav --device) as with original recordings
30. Reordering from Bin2wav to a standard sample rate of 48kHz (44.1) kHz  
**ATTENTION! The effect of the -l2 parameter has been changed.**

## 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, Robert VAN ENGELEN

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, Wolfgang KLEIN, Walter FISCHER, Michel van ZELST

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, Michael ZIRPEL,  
Yoshimine HORIUCHI, Peter MÖRTH, Burkhard EICKE

Horst SCHAEFFER für sein Batch-Tool "Wfile", Jens GRUSCHEL (Pegtop) für sein "PStart",  
Jan FIALA für PSPad und audacityteam.org,

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, May 2021

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!