Documentation for PCFtool Interface

This document describes the communication with the USB interface between the frontend and the PCFtool, which enumerates itself as virtual serial port. For communication every serial command line tool (e.g. hterm) could be used.

PCFtool itself implements an interface for communicating with PCF79xx automotive key transponder ICs via Monitor and Download interface (MDI). It enables lots of interactions with the PCF79xx chip like writing and reading to EEROM and EROM, unlocking (=erasing) locked chips.

PCFtool Communication Protocol

In order to interact with the PCFtool, the host (PC) sends a command packet consisting of 5-byte via COM-Port. For special commands, the command packet must be extended by payload data and 4-byte CRC32 checksum packet. After each command was executed, the tool sends back one status byte. This feedback can indicate if a command was successfully executed.

Short Description: PCFtool Commands

CMD	Parameter	Description	Chapter
09	0X 00 0Y 00	**Connect** Erase Chip: X = 0: Normal connect X = 1: Connect and Erase chip MDI Sequence Type Y=0: 26A0700 Y=1: PCF7945	**1.1**
0A	00 00 00 00	**Erase**	**1.2**
1A	00 00 00 00	**Protect**	**1.3**
2A	00 00 00 00	**PCF_Tool Update Mode**	**1.4**
2B	YY XX ZZ VV WW_0…WW_x	**Program ER Buf** Start Address:	**2.1**

	MM_0…MM_3	YY: Start Adress low byte XX: Start Adress high byte Data Length: ZZ: Length low byte VV: Length high byte Data payload: WW_0…WW_x: Data to be written Checksum: MM_0…MM_3: CRC32 highest byte to lowest byte. Can be left all zero, then no CRC is used	
3B	YY XX ZZ VV WW_0…WW_x MM_0…MM_3	**Program EE Buf** See command „Program ER Buf“	**2.3**
4B	00 00 00 00	**Program ER**	**2.2**
1B	00 00 00 00	**Program EE**	**2.4**
5B		**Program EE Manual**	
6B	XX YY 00 00	**Program Special Bytes** (Page 127) XX: Page 127 byte[2] YY: Page 127 byte[3]	**2.6**
0D	00 00 00 00	**Read EROM**	**3.1**
1D	00 00 00 00	**Read EEROM**	**3.2**
2D	00 00 00 00	**Read EROM Buffer**	**3.3**
3D	00 00 00 00	**Read EEROM Buffer**	**3.4**
4D		Read ER Buf CRC	
5D	00 00 0Y 00	**Read PCF memory checksum** Y=0: EROM_NORM Y=1: EROM Y=2: EEROM Y=3: ROM	**3.5**

Feedback (Answer from Tool) Bitmask:

Bit Nr.	Info	Detail
0	OK	Performed Action was successfully executed
1	Too less data	PCF returned less data than expected
2	Err. State received	PCF responded with an error status to the cmd / data former sent
3	Receive Timeout	PCF did not respond (return data) to the last action or cmd. No SCL edges were detected.
4	Prog special pages error	After writing special page 127 (byte 2 and 3) to EEROM, PCF did not confirm afterwards
5		
6		

1 General commands

1.1 Connect

Before any operation to the PFC memory can be performed, it must be initialized in MDI mode.

Parameter description:

09	0X	00	0Y	00	Description
	00				Normal Connect
	01				Connect MDI and directly delete chip. A locked chip must be erased after connecting MDI prior to any other operation. Afterwards it is unlocked and can be accessed
			00		**Chip type:**26A0700 (found in cheap chinese replica keys)
			01		**Chip type:** PCF7945C05

1.2 Erase

This command the EROM and EEROM of a PCF chip and sets it to “Initial” state. It requires a successfully initialized MDI connection, which means that this command can only be used alone if the chip is not in protected mode. If the latter applies, then the connect & erase procedure described in 1.1 is to be used.

1.3 Protect

After the chip is successfully programmed, it can be protected against further read-out via MDI interface. In order to unlock it again, it needs to be erased as described in 1.1. In order to execute this command, it must have been connected successfully before.

Parameter description:

1A	**00**	**00**	**00**	**00**	**Description**
					-

1.4 PCF_Tool Update Mode

The PCF_Tool is equipped with a bootloader, which enables the update of the device firmware. This allows to add new functionalities or to support more chip types.

2A	**00**	**00**	**00**	**00**	**Description**
					-

After the command is issued to the PCF_Tool, the COM-Port device disconnects and a new USB storage device appears. In order to flash a new PCF_Tool firmware, the corresponding .hex file just needs to be copied to the USB drive. Afterwards, the USB storage gets disconnected and PCF_Tools COM-Port is recognized again.

2 Write data to PCF

The PCF tool provides a buffer into which the payload data needs to be written prior to the final programming of the chip. This means writing data to a PCF is generally done in two steps, first the transmission to PCF_tool's buffer and second the flashing to PCF. Please take notice of the memory naming: EROM means the flash, EEROM the actual EEPROM.

Recommendation: In order to program data to a PCF, first prepare a bin file which contains the 5 command bytes, followed by the data payload + the 4 checksum bytes. This complete file can then be send to the PCF_Tool all at once.

2.1 Write EROM Buffer

This command writes EROM data to PCF_Tools's internal buffer. In the command, the address of the chip where the data shall be programmed to and the length of the data needs to be specified. Right after, the final data must be send. The data can be either terminated by 4 bytes CRC32 checksum of the data, or by 4 bytes zero which will switch off the checksum verification. The CRC32 can be calculated e.g. with online calculator https://crccalc.com/.

Packet Format:

2B YY XX ZZ VV $Data_0$… $Data_x$ $CRC32_0$… $CRC32_3$

Parameter description:

2B	YY	XX	ZZ	VV	Description
		XX			Low Byte of Chip's start address where data are programmed to
	YY				High Byte of Chip's start address where data are programmed to
				VV	Low Byte of Data Length
			ZZ		High Byte of Data Length

2.2 Program EROM Buffer to PCF

The EROM Buffer former written to the PCF_Tool can be finally programmed to the chip with the command described below. It must be assured that a connection to the chip is still existing.

Parameter description:

4B	00	00	00	00	Description
					-

2.3 Write EEROM Buffer

This command writes EEROM data to PCF_Tools's internal buffer. In the command, the address of the chip where the data shall be programmed to and the length of the data needs to be specified. Right after, the final data must be send. The data can be either terminated by 4 bytes CRC32 checksum of the data, or by 4 bytes zero which will switch off the checksum verification. The CRC32 can be calculated e.g. with online calculator https://crccalc.com/.

Packet Format:

3B YY XX ZZ VV $Data_0$… $Data_x$ $CRC32_0$… $CRC32_3$

Parameter description:

3B	YY	XX	ZZ	VV	Description
		XX			Low Byte of Chip's start address where data are programmed to
	YY				High Byte of Chip's start address where data are programmed to
				VV	Low Byte of Data Length
			ZZ		High Byte of Data Length

2.4 Program EEROM Buffer to PCF

The EROM Buffer former written to the PCF_Tool can be finally programmed to the chip with the command described below. It must be assured that a connection to the chip is still existing.

This command does automatically write the special bytes 2 and 3 of EEPROM Page 127 to the value specified in buffer. Page 0 and 126 are skipped as they are read only.

Parameter description:

3B	**00**	**00**	**00**	**00**	**Description**
					-

2.5 Program EEROM special bytes to PCF

The EEPROM of each PCF has some pages which are reserved for special information needed by the chips basic software. A page is means a memory block consisting of 4 bytes.

The table below gives an overview of the these special pages for an PCF7961. Note that only page 127 has data which can by modified, which is marked with the [RW] tag.

Byte Number									
0		**1**		**2**		**3**			
Bit Number									
0..3	**4..7**	**8..11**	**12..15**	**16..19**	**20..23**	**24..27**	**28..31**	**Page**	**Address**
PCF's ID. Same as used in HiTag2 data. [RO]								0	0
Trimming Information written at factory [RO]								126	504
EEDMx [RO]	EEDM [RO]	CAL [RO]		TMODE [RW]		ID [RW]		127	508

Writing to the two mentioned bytes is done automatically done with the Write EEROM command. However it is possibly to write only these two bytes with an explicite command. Please note that this operation is only possibly under certain conditions which are not exactly known. For example it was discovered that the command only worked directly after chip erase command was performed (no other memory programming was done).

Parameter description:

6B	XX	YY	00	00	Description
	XX				Data byte programmed to Page 126 byte 2
		YY			Data byte programmed to Page 126 byte 3

3 Read Data

Reading data from PCF's memory does not use the Buffer mechanism which is used for programming data to PCF. Thus, for reading out the chip, it is sufficient to send a single command. Please keep in mind that the very last byte received is the status byte and thus does not belong to the chip's data.

3.1 Read EROM

After the below described command is issued, PCFtools reads the whole EROM of the chip and sends it to the host PC.

Response:

$Data_0$… $Data_x$ Status

Parameter description:

0D	00	00	00	00	Description
					-

3.2 Read EEROM

After the below described command is issued, PCFtools reads the whole EEROM of the chip and sends it to the host PC.

Response:

$Data_0$… $Data_x$ Status

Parameter description:

1D	00	00	00	00	Description
					-

3.3 Read EROM Buffer

The content of PCFtool's EROM write buffer, which is used for programming data to the chip, can be read back. This might be used to check the buffer to have the correct content before final programming of the PCF.

Response:

$Data_0$… $Data_x$ Status

Parameter description:

2D	00	00	00	00	Description
					-

3.4 Read EEROM Buffer

The content of PCFtool's EEROM write buffer, which is used for programming data to the chip, can be read back. This might be used to check the buffer to have the correct content before final programming of the PCF.

Response:

$Data_0$… $Data_x$ Status

Parameter description:

3D	00	00	00	00	Description
					-

3.5 Read PCF's memory checksum

PCF series chips are providing 3 byte checksums for content-verification of their different memory sections. These checksums can be read out. However, not all different checksums are provided by each specific PCF chip types. If the PCFtool answers a checksum command with an error status, then it's most likely that your PCF does not support the checksum for the specified memory region or the PCF is in protection state in which the issued command is not supported (in protected state, only normalized EROM signature is supported)

Response:

CKS_0… CKS_2 Status

Parameter description:

5D	00	00	0Y	00	Description
			0		Normalized Signature of the EROM. Bytes individual for each specific key (calibrated during manufacturing) are substituted by 0). Command possible even in protected mode
			1		Signagure of EROM. Not normalized. Only possible with unprotected PCF
			2		Signature of EEROM. Not normalized. Only possible with unprotected PCF
			3		Signature of ROM. Only possible with unprotected PCF