Passport communications protocol

The Serial, Bluetooth and USB interfaces all act as COM ports on the PC side, so once the interface is implemented in the CAT system, it should work for all three device communications methods as long as the user can choose a COM port.

The serial port defaults to 38,400 baud, no parity, 8 data bits, 1 stop bit.

When the user hits a stroke on the writer, the writer sends out a stroke immediately in the following format:

<strokeNumber/strokeText/timeStamp[/optionalFlags]>

the strokeNumber will start at 0 and increment once for each stroke.

the strokeText is in the following format: LetterShadowLetterShadowLetterShadow...

It will send only those letter keys that have moved at all during the stroke. The shadow value is a hexadecimal character from 0 - f (note the lower-case letters for hex digits.) A key with at least a shadow value of 8 counts as registered. Lesser values are unregistered keys.

The timestamp is a Windows FILETIME value converted down to milliseconds (i.e. divided by 10000.) The first timestamp in the job (the timestamp for stroke 0) is the absolute time. The rest of the strokes in the job will be the number of milliseconds elapsed since that starting timecode.

EXAMPLE: If the 15th stroke in the job, 43 seconds into the job, was the user hitting SAT on the steno machine, lightly dragging the L key (not registered) and lightly hitting the T key (registered), it might look something like this:

<14/SfAfL5Tc/43782>

The keys on the Passport keyboard are indicated by the following characters in the stroke data:

! (mark)
# number bar

```
^STPH ~ FNLYD
+CKWR * QBGXZ
    AO EU
```

The CAT software can also optionally request strokes that were missed during a bench conference or if the user started writing before starting realtime by transmitting a stroke request command to the Passport in the following format:

<sStrokeNumber/StrokesToSend>

Example: <s15/50> would send up to 50 strokes starting at 15 (strokes 15 through 64.) It is permissible to ask for more stokes than the writer contains and it will send as many as there are in the file.

Passport note file format

jobxxx.pnt - the steno data

jobxxx.ptx - the translated text (not relevant for anything other than readback.)

jobxxx.pnf - job information

jobxxx.wav - A standard WAV audio file (only on SD card)

The .pnt file contains data stored in the following structure format:

```
struct Stroke
{
        unsigned char keys_[14];
        unsigned short flags_;
        FILETIME time_;
};
```

That's 24 bytes per stroke altogether. The flags_ data is reserved for future expansion.

The keys_ data is in the following order, four bits per key, which maps to the 0-f shadow data indicated in the realtime interface (0 indicates that the key did not move at all.)

Byte #

0	1	2	3	4	5	6	7	8	9	10	11	12	13
!#	^+	SC	TK	PW	HR	AO	*~	EU	FR	PB	LG	TS	DZ

The time_ value is the time/date when that stroke was hit, using the Microsoft FILETIME structure.

In order to synchronize with the WAV file, it will be necessary to read the information from the .pnf file. That is a text file that contains a variety of information about the job delineated by [type] blocks, much like a .ini file. The [wavlog] segment contains a sequence of numbers, one per line. This is a series of FILETIME values indicating the exact date/time when the recording was started or stopped during the job. If the high bit is set (0x8000000000000000) then it indicates a stop. Since the user can pause the recording (which can also be done by the auto-pause feature) it will be necessary to use this information to calculate where the gaps are in the WAV file so that it's possible to re-adjust the synchronization between the notes and the WAV file in the CAT software using whatever method the CAT system supports for keeping track of audio sync.

Passport dictionary format

In this description, each single character represents a byte, so A is a byte, BB is a word, CCCC is an int

All values are little-endian. The file data is in the following format:

SSSSMMMMNNNNLLAAAABBBBCCCCDDDDEEEEFFFF....XXXXYYYYYYYYYYYZZZZZZZZZZZZZZ

SSSS is the header size (all global data before the index starts, including itself)
MMMM is the location of the first entry (after the index)
NNNN is the total number of entries in the dictionary (number of indexes used)
LL is the language identifier

AAAA through XXXX are the indexes themselves, however many there are.
Note that the index should be pointing to entries sorted in reverse steno order, from -Z to S. The actual order of the entries is irrelevant as long as the index to them is sorted.

YYY is the blank space that takes up the rest of the available index memory. It can be any size, including zero.
ZZZ is the sequential list of entries, as described below

Entry list format:

SAAAABBBBCCCCTTXXXXXXXX

S is the total number of steno strokes (maximum of 255 strokes per entry)
AAAABBBBCCCC are the strokes themselves, at four bytes per stroke
TT is the total length of the text (maximum of 65535 characters of text per entry)
XXX is the text data in wchar_t format (two bytes per character)

This format is used for both main dictionaries and job dictionaries. The Passport has functions for copying dictionaries created externally into its own memory and copying its dictionaries out to external media, so it is helpful to have features for both importing and exporting this dictionary format in the CAT software. Though the Passport has an RTF converter built-in, it takes a long time to convert a large dictionary using the Passport's limited memory and processor speed, so a native dictionary converter in the CAT system is preferable.