

Token Processing System Manual

Stephen Lavelle

July 2016

1 Architecture

The *TPS-5* is a versatile system for manipulating tokens. It can store in total 224 tokens, divided into seven buffers of 32 tokens each.

There are seven buffers - one (*DATA*) is for public input/output, to be read by external software, the remaining six are private buffers for your programs. Three (*TPS#1, TPS#2, TPS#3*) are where you will write your code, and the other three (*STACK#1, STACK#2, STACK#3*) are stacks for temporary storage. All six of the internal buffers have individual TPS microcontrollers.

A TPS microcontroller has access to three buffers - data, script, and stack. At any one time only one TPS microcontroller can be active - a TPS microcontroller can give control to the TPS that runs on its data buffer (if there is one - *DATA* is not interpreted by any microcontroller), or it can relinquish control back its parent controller. The controller supports 16 instructions.

2 Instructions

BOTTOM Moves the child cursor to the last place.

END Halts termination of program.

EXEC Looks at the current child token, and executes it (without changing context).

IF The two tokens following *IF* are called the *antecedent* and the *consequent* respectively. If the currently selected child token is the same as the *antecedent*, then control goes to the consequent, otherwise control goes to the token one past the consequent.

IN Transfers control to a child TPS, if one exists.

OUT Transfers control to a parent TPS, if one exists.

PREV Moves the child cursor back one, if possible.

NEXT Moves the child cursor forward one, if possible.

POP	Sets the currently highlighted child token to be the currently highlighted stack token.
PUSH	Adds a copy of the current item to the stack at the position of the stack cursor.
REMOVE	Deletes the token currently pointed to in the child data.
RESTORE	Restore the child cursor's position to that specified by <i>SAVE</i> .
SAVE	Remember the position of the current child cursor.
SWITCH	Switches the buffer and the stack.
TOP	Moves the child cursor to the first place.
WHILE	As for <i>IF</i> , keeps executing the consequent so long as the antecedent is the same as the currently highlighted child token. If this is no longer true, control moves to the token after the consequent.
	All other tokens result in the interpreter doing nothing.