

**Spud Ai** is a computer program that is an attempt to create artificial intelligence (AI). The program is built by taking my understanding of "the mind" and turning that understanding into a computer program. The program is also a practical tool and is used for numerous tasks in my everyday life.

# From Wikipedia

Artificial intelligence (AI) is the intelligence exhibited by machines or software, and the branch of computer science that develops machines and software with intelligence. Major AI researchers and textbooks define the field as "the study and design of intelligent agents",[1] where an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success. [2] John McCarthy, who coined the term in 1955,[3] defines it as "the science and engineering of making intelligent machines.

The Ai's basic function is to process information in an intelligent manor and respond accordingly. It has numerous processing functions and algorithms that act on information in different ways most of these are language based which gives the Ai the ability to work as a chat bot. There is a main database called the Brain Database (BD), entries in the BD are called Cells. A cell can either hold information or it can be 'activated', when activated the cells neural network takes the input code, processes it through the network and produces the output code. How the input code is processed and the output it produces is dependant on how the network has been trained. The output nodes can be connected to other cells input nodes, the Tag field is also used to control how the output is used. As the Ai evolves the BD grows in size.

When used as chat bot the language processes take the input and processes it in different ways. The majority of these processes come from flashes of inspiration and whether these inspirational bits of code form some sort of mind or produce replies that show some form of intelligence only time will tell as these work by

forming connections that evolve as the Ai builds up its data base through either data inputted manually or by the various learning processes (LM). Learning processes add data which is then worked on by the various processes, this gives the Ai the ability to output correct information without having being told about the specific subject but from associations only. Other processes are designed to give specific answers to specific questions or use some of the input in the output.

The Ai is packed full of functions which gives the Ai the ability to perform many practical tasks. As the Ai evolves more functions are added. Keywords associated to a cell word are used for many of the functions. Cells 800 to 2000 are know as system cells these are used by the various functions and processes including various input output devices from hardware to networking. Cell 2000 and upward are used for data storage like words and patterns

The things the Ai can be used for is an ever growing list, its designed to be flexible in usage. There are functions within the Ai that are there because i've needed software to do a certain task. Instead of creating a new program i just add a new function to the Ai.

Here is a list of things i use the Ai for: searching indexed files for information, working as a chat bot on a web page, creating price list information for my company web site, indexing job files for my companies database, taking a picture of my solar controller and sending it to the web server, playing music, controlling relays in my house, monitoring and controlling power distribution from my renewable energy systems, object recognition for the robot mapping functions, controlling a robot, quick information like finding out tomorrows weather and the main news story and talking to.

The Ai is designed to do things i haven't even thought of.

The program is under constant test and runs for long periods of time it is designed to be as stable as possible, there are some functions that monitor program stability and react if any problems are found, either by fixing the problem automatically or by alerting the user.

The programs main input is text words which can be generated by voice to text programs, network messages, text files, keyboard, web page form etc. There is also ways of inputting data directly into the neural network.

The input is processed by the program and then it produces an output which could be to run a script, program, create a web page send a message through a network, create a sentence, activate the neural network.

The answer is created according to how the input is translated using various functions within the program which use information in the Brain Database.

The Ai program is available as a free download as long as you accept and agree to the following:

We take no responsibility for any consequences that may occur from using the program.

You must not train the Ai to break its own law, know as Robot Law which is a text file and is part of the download.

You must not disassemble the program in any way.

You may use, pass on, redistribute as you like as long as you don't remove any copyright notices. If you pass the program on you must make sure you also supply the link to this page.

The documentation may be unclear or at times confusing, this situation will change over time as the project develops.

© 2004 - 2017 Piet Defoe

# Click here to download Ai Beta...

We would be very interested to know what you think about the program so please email us.

If you have a use for the Ai and need some form of extra functionality or a change in the program to adapt to you own needs then this may be possible but expect to pay something!

The program has zero funding and is a labor of love so if you want to invest in this project or any other Spud projects then please get in touch.

# Ai Web site

### **Email Us**

last update 08-01-2017

© 2004 - 2017 Piet Defoe

# **Brain Database (BD)**

The BD holds all the information required by the various process within the program. Each entry in it is called a Cell. Each cell in the BD has a 9 bit neural network which is activated by various processes

Some of the fields in the cell have a specific use and others are used for different things.

The BD is a combination of two databases merged together. Ai versions previous to version AiV016a had two databases called TF Translation file, and NNE Neural Network Extended. This is why there are some references to the TF or NNE in keywords and the help file in general. These references will change over time as and when the help file is updated.

cell 1 holds the length of the BD cells 2 to 802 have no set use cells 802 to 2000 are System Cells and are used by the program for various functions details here <a href="mainSystemCells.htm">BrainSystemCells.htm</a> cells 2000 upward is data added to the database by various functions

#### A field either holds a number or text

number or text	name	Description.
		Input code used in the previous activation, mostly used by there has been a change to the input.
number 1 or 0	input code	This code is put into the neural network which then loads the process and sets the output code. The input code can either processes within the program.
number 1 or 0	output code	The output from the neural network.
text	filename	Either the training or weights file. Training files have a .nnt .nnw extension.
text	cell action	Used to bypass the processing of the input code by the neur
text	emotion	this is a code held in tags that is used by the program to sto Emotion routines. Cell 935 holds information used by the e <f:42> the number of times the word has been found in an every time the word is found in the input. If the word count</f:42>

		to 0. This means the word moves down the list in cell 935.
number	node 1 to node to node 9 to node	Connects the output nodes to input nodes of other cells. The changed when the cell is activated.
number	in cell	Target cell used when connecting output to input nodes.
text	tags	During cell activation the tags field is read. Click here for c
text	field 11	General use and extra keywords.
text	file or f9	File name where required.
text	detail	The dictionary description of the word.
number	mode	Controls how the input is processed cell 832 is the default s 3, 2008 learning routines are activated, if cell 832 mode=3 5, translate, translates the cell word with the word in f7 and to the output 6, looks for the sentence word in the detail field and then be words. This mode uses the last word found so the words in recently added. The number of times words are found mear generated, though this output will be random and be void o 7, puts input into output, bypasses ProcessInput and goes st
number	operation	Activates a system command, these need a command word field set to 33 proceeding the operation word  1, System command on the file field, field 10 is the extra parexecutable, field 8 is the reply  2, copy file to filename in field 10 if the type field is 35 the which will enable copy even if the file exists and will be ov  3, activate the cell  4, delete file in file field, full path  5, move file to the file name in field 10 type field must be 2 existing file  6, show pictures in file field, full path  11, turn ON digital channel output 1  12, turn ON digital channel output 2  13, turn ON digital channel output 3

		14, turn ON digital channel output 4 15, turn ON digital channel output 5 16, turn ON digital channel output 6 17, turn ON digital channel output 7 18, turn ON digital channel output 8 21, turn OFF digital channel output 1 22, turn OFF digital channel output 2 23, turn OFF digital channel output 3 24, turn OFF digital channel output 4 25, turn OFF digital channel output 5 26, turn OFF digital channel output 6 27, turn OFF digital channel output 7 28, turn OFF digital channel output 8
number	order	The order field is a bit like operation field but instead of pe used to carry out functions like combining files and creating function word is needed to activate the function, word type $0 = \text{unused}$ $1 = \text{unused}$ $2 = \text{unused}$ $2$ , reply with the contents of 7. $3$ reply with the contents of field 8, PR Tags can be used to
number	type	This field is used for word type. Different word types are processed in different ways by the The Sentence Code (SC) is created according to the word ty number is used to define the character used to represent the defined here
text	cell word	The word associated to the cell.
number	cell number	The number in the database.
text	keyword	Brain Keywords
	field 7	Various uses.
text	field 8	Various uses.
text	field 10	Web addresses and various other uses.

number	score,	The score achieved during the last neural network training
	generations,	number of generations up to the maximum number of gene
	max	better the score, depending also on complexity of the traini
text	in	Used by FIND the string is searched for in the field.
number 0 and 1	data	The cell training data, this is stored in a text file. When the neural network is trained using the data. Weights are create hidden field in the cell data.

#### **Buttons**

**IN OUT to TRAIN** writes the input and output code to the training data.

**TRAIN** trains the Neural Network using the training data.

**GOTO CELL** opens the cell number in the box to the left of the button. If the box is empty then the last cell in the BD is opened.

**ADD** adds the contents of the box to the right of the button to the BD. A new entry is created and the contents of the new cell are that of the cell currently displayed but with the new cell word.

**FIND** finds the contents of the box to the left of the button in the field displayed after in.

**UP** moves up to the next cell, data is saved before the next cell opens.

**DOWN** moves down to the previous cell, data is saved before the next cell opens.

**DEFRAG** de fragments the BD if enabled in Settings

**CLEAR CELL** clears all the fields in the cell.

**UPDATE** writes the contents of the cell fields to the BD.

**ACTIVATE** activates the cell.

**TO TEXT** writes the contents of the cell to the text file NewBrainData.txt in the data folder. This text file can then be used to update the Brain of another Ai. Copy the file into the data folder of the Ai to update then run the keyword UpdateBrain

Tags are used to perform various functions when the cell is activated. A colon: is used in front of normal numbers equals = is used in front of neural network codes.

	1
Tag	description
<activate-b10+></activate-b10+>	cell number to activate is the output converted to base 10 the number after the + cells 2 upward are activated.
<activate-cell:></activate-cell:>	activates a cell in the Brain database
<all></all>	use all outputs sets NNETagAll = 1
<allowrepeat></allowrepeat>	sets the allow repeat variable allows repeated outputs. Norm a repeated output until 9 other outputs have been used.
<atmfiic></atmfiic>	add to matrix file if the input changes atmfiic.txt in weights
<b10-o></b10-o>	converts output to base 10 send to output
<checkwords></checkwords>	this tag checks the ToPrint output variable against words in words and words at the end of the brain are checked process words are jumbled up so its not useful as a output filter but i like auto learn which sometimes output weard codes from w
<clear></clear>	empties the Ai output buffer, sets the ToPrint variable to not
<code-o></code-o>	sends the 9 bit digital output to output
<dtlno></dtlno>	DT Learning creates new connections this tag stops the DTI cell numbers in the NNE
<eventactivate></eventactivate>	activate the cell if there is a Brain event. More detail bellow
<eventcell></eventcell>	writes the cell number to the file NNEEventCells.txt in the v found as an event
<eventno></eventno>	this stops the cell from being scanned by the nne event scan expected changes to input or output nodes of the cells 1000 t
<f7-ifoabove:></f7-ifoabove:>	the value is a base 10 number the output is converted to base to the number eg. <f7-ifoabove:5> if the output converted to 5 then the tag processor starts reading field 7 instead of the t</f7-ifoabove:5>
<f8-ifobelow:></f8-ifobelow:>	the value is a base 10 number the output is converted to base to the number eg. <f8-ifobelow:5> if the output converted to</f8-ifobelow:5>

	then the tag processor starts reading field 8 instead of the tag
<f7-cell:> <f8-cell:> <f10-cell:> <f11-cell:> <f12-cell:></f12-cell:></f11-cell:></f10-cell:></f8-cell:></f7-cell:>	use the contents of the field from the cell specified for the outhe cell number found from a <lookup-cell> or <lookup-ac></lookup-ac></lookup-cell>
<f7-ifa> <f8-ifa> <f10-ifa> <f11-ifa></f11-ifa></f10-ifa></f8-ifa></f7-ifa>	go to the field and continue processing tags if the cell is activ
<f7-ifo=> <f8-ifo=> <f10-ifo=> <f11-ifo=></f11-ifo=></f10-ifo=></f8-ifo=></f7-ifo=>	go to the field and continue processing tags if output code ec
<f7-ifi=> <f8-ifi=> <f10-ifi=> <f11-ifi=></f11-ifi=></f10-ifi=></f8-ifi=></f7-ifi=>	go to the field and continue processing tags if input code equ
<filter:></filter:>	UNDER DEVELOPMENT <filter:n> where n is the filter number  filter:1 = input code used to control the words in the ToPrint  1 then corresponding numbered word in the sentence is used omitted.  filter:2 = this is an upgrade for the prwordafter function. It is word is a pattern, as defined by the Type field being set at 60 looked for in the SC code created by the input if found then set to the words after the word that the pattern corresponds to used to control the position of the pattern in the SC code the position it must be in the trigger the filtering. If Order = 0 th anywhere in the SC code. The input code is set to 11000000 of filter have been set and the filter used its set to 001100000 or</filter:n>
<grammar></grammar>	processes the output through the grammar checker changing

	english
<i=cell:></i=cell:>	changes the cell input to the input of the cell numbered
<ifpi=i-ac:></ifpi=i-ac:>	if the input is the same as the previous input than activate the
<inouttrain></inouttrain>	like the button <b>O</b> to <b>Train</b> this tag puts the input and output box then trains the cells network the new data is only added exist it doesn't check for conflict data. An example of conflict input code with a different output code, since only one output the network would never achieve 100% accuracy.
<keyword-ifo=></keyword-ifo=>	use keyword in the keyword field if the output equals
<keyword-ifi=></keyword-ifi=>	use keyword in the keyword field if the input equals
<keyword-ifa></keyword-ifa>	use keyword if the cell is activated
<log></log>	logs the cell activation
<lookup-ac></lookup-ac>	look in brain for the text output and if found activate the cell
<lookup-ac-add></lookup-ac-add>	same as <lookup-ac> but adds the contents of ToPrint variab found.</lookup-ac>
<lookup-cell></lookup-cell>	This sets the variable LookUpNumber to the cell number of up.
<outtoin></outtoin>	puts the output into the Ai input
<rq=></rq=>	sets the reply quality variable
<rq=nne></rq=nne>	sets the reply quality to the NNE reply quality weight
<nodewords-ifo=></nodewords-ifo=>	use the words connected to the nodes if the output equal
<nodewords-ifi=></nodewords-ifi=>	use the words connected to the nodes if the input equal
<nodewords-ifa></nodewords-ifa>	use the words connected to the nodes if the cell is activated
<once></once>	stops the cell being activated more than once during an activ
<o-udp-cell:></o-udp-cell:>	send the output via UDP to the input of the cell
<playmidi></playmidi>	see below for details on this tag
<sayprint></sayprint>	goes to the SayPrint routine which prints the output variable
<speak></speak>	uses windows speech to speak the cells output. The contents ToPrint is sent to the speak routine that uses the windows co Mode = 0 Asynchronous mode, similar to mode 2 but withou Mode = 1 synchronous mode the program halts until the speprogram processes are halted including timers so if you have

	activates whilst speech is happening then it will be ignored. Mode = 2 Asynchronous mode, the program continues to rul happening, the main timer is restarted after a calculated time can be overrun or ignored which can cause problems. WARI 2 if the Ai is going to read out a file, the program crashes for reason use mode 3 instead.  Mode = 3 this is a combination of mode 1 and 2 normally sp asynchronous mode but if ReadFile keyword has been invok out using synchronous mode.  If the swopvar is empty when a speak tag is invoked then the set to the output sent to the speak process.  The Speak tag will only work on Windows Vista and above.
<stop-m></stop-m>	stop processing the matrix file if one is being used to activat
<stow></stow>	scan to weights see below for more details
<swopvar></swopvar>	some functions set the swop var this tag sets the Brain outpu of swopvar, not yet fully tested
<swopvar=toprint></swopvar=toprint>	Sets the SwopVar variable to the value of the output variable
<swopvar-udp></swopvar-udp>	Send the contents of SwopVar variable via UDP. The do: sw to output
<tagfile></tagfile>	the file field is the file name of the text file holding more tag is less than 250 caricatures
<toprint-f7> <toprint-f8> <toprint-f9> <toprint-f10></toprint-f10></toprint-f9></toprint-f8></toprint-f7>	sets field mentioned of the cell number to the contents of the tags <lookup-cell> or <lookup-ac> has been used then the cellookup if found.</lookup-ac></lookup-cell>
<toprint+swopvar></toprint+swopvar>	contents of SwopVar are added to ToPrint instead of replacir
<toprint=swopvar></toprint=swopvar>	Sets the output variable ToPrint to the value of the SwopVar
<uwbrowser></uwbrowser>	This tag creates a browser window, mode left, operation dow height. Field 8 is the startup url if empty then it defaults to the page. If activated the contents of the output are sent to the broan be created at the moment. At present there are no naviga than the right click options and functions on the page. The broaden are the page of the page of the page of the page.

	intended to replace a normal browser and lacks some of the embedded browser functions require Internet Explorer 4.0 or on the system. If the browser is used then the Ai's memory a increased, depending on the contents and elements within the settings, history etc is accessed using internet explorer.
<uwbutton></uwbutton>	User Window tag, creates a button in the user window using button text is reset after activation so the text can be changed contents of field 7, mode=left position, operation=top position width. When the button is pressed the cell is activated. The fare the button images, file is the main image and field 11 is to Images should be windows bitmap .bmp and should be in the name in the file and 11 fields should be the image name only text color and field 10 is the button color. If either the file or no image is loaded. If field 8 or 10 are empty then no colors defaults to gray with black text
<uwface></uwface>	User Window tag, draws face animation mouth and eye moves by the input code. If there is a NN code in field 11 then this is the output code is used for the mouth.
<uwimage></uwimage>	User Window tag, image for user window. The image file na If Order and Type fields are 0 then no scaling is applied the ithat of the image.  If Order field is 0 and the Type field is higher than 0 then the size of the user window  Mode is the horizontal starting position and Operation is the position.
<uwprint></uwprint>	User Window tag, moves the print position using the mode a then prints the output
<uwtextin></uwtextin>	User Window tag, the text from the input box is put into field field 10 is the font, mode, operation, type and order are the c When the cell with the tag is activated the ToPrint output valcontents of the input box which is then cleared.
<uwtextout></uwtextout>	User Window tag, sends the output from the cell to the output show the output after processing an input tick in settings Act and use the tag <activate cell:=""> to activate the cell used by usize field 10 is the font, mode, operation, type and order are size</activate>

	This turns the cells words output into the cells input code. The variable are converted into a number using there ASCII valu together until it is higher than 255 then converted into a nine by the neural network. The variable NNIn is the cells neural this is altered to the new value.
<words-ifo=></words-ifo=>	use the words if the output equals
<words-ifi=></words-ifi=>	use the words if the input equals
<words-ifa:></words-ifa:>	use the words if the cell is activated

#### **More Detail**

<eventactivate>, the cell is activated if an event occurs. An event is only recorded if a cell changes unexpectedly if a cell is constantly changing then this is classed as normal and the cell doesn't record an event. If the cell is activated then no more events will be recorded from that cell until the Ai has processed an input.

**<stow>** Scan TO Weights. This tag scans a section of the image 18 x 9 pixels and converts the color value of the pixel into a weight value. There are 160 weights used in each cells neural network. This tag is an attempt to put information directly into the neural network. This enables one cell to hold the information from 160 pixels and then use that data to process a 9 bit input code and produce a 9 bit output code. The weights data is stored in the weights variable associated to the cell like normal this data can be overwritten by training the cell using the TRAIN button. The data is averaged in order to cope with difference of light and dark. at the moment this routine is working well at recognizing different images very quickly. Future development is to use this method to work out objects in an image.

field	contents
input code	000100000
tags	<stow><words-ifo=111100111the image="" jump=""><words-ifo=11100 image=""><sayprint></sayprint></words-ifo=11100></words-ifo=111100111the></stow>
cell word	stow

<playmidi> Field 10 is the format for play midi if mode = 1 then the midi file is

played in the background if not then the program freeze until the midi string has finished being played.

The midi string is made up using a code described bellow n is the number value A-G  $\{\#\}n$ 

Plays a note. The # specifies a sharp. n is the duration of the note and can be 1,2,4,8,16,32 or 64 representing whole note, half note, quarter note, eighth note, and so on.

On

Specifies the octave of the notes that follow. There are 11 octaves from 0 to 10 with octave 5 containing middle C. Each octave begins with the C note and ends with the B note.

In

Sets the instrument (patch) to be used for the notes that follow. n can be from 0 to 127.

Vn

Sets the velocity, or volume, of the notes that follow. n ranges from 0 which is completely silent to 127 for full velocity.

Nn

Sets the channel of the notes that follow. n is the channel number and can range from 0 to 15. The default channel is 0. Channel 9 is the built in drum channel for most sound cards.

Rn

Inserts a rest. n specifies the duration of the rest and are the same as note durations.

Tn

Sets the tempo of the music in beats per minute (bpm). Default is 120. n can range from 10 to 300

\*

Begins a chord. Any commands between the \* and ; are played at the same time index.

;

Ends a chord.

eg: T120 N0 I25 O5 \*C2EG;D4EFG\*A2O6CE; \*C2EG;D4EFG\*A2O7CE;

number	SC neumonic	word type
1	#	more info required don't use for translation
2	%	Noun
3	&	Adjective
4	1	Verb
5	(	Adverb
6	)	Pronoun
7	*	Article
8	+	Conj
9	,	Int
10	-	plural
11		pronoun
12	/	abbr
13	0	prep
14	1	past
15	2	singular single letter
16	3	prefix
17	4	obj
18	5	verb and adjective
19	6	adjective and noun
20	7	noun and plural
21	8	past and past part
22	9	poss pron
23	:	variant
24	;	adjective and colloq
25	<	pres and part
26	=	past part
27	>	predicted adjective
28	?	3rd sing. present
29	@	verb and noun, multiple meaning
30	A	word found in\data\UserDictionary.txt
31	В	bad word word not found in BD or user dictionary
32	C	sentence code if tags are found in f8 then these tags are converted otherwise just the contents of f8 are used. If f3=3 then tags are overruled and f8 is used for the output even if
32	С	then tags are overruled and f8 is used for the output even

tags are present. This is due to keeping legacy code working	g
and provides a fall back output this is why the RQ is	
reduced.	

33	D	start command
34	E	question interrog. adv
35	F	absolute order forces a command like copy if a file exists then by default it wont be over written but if this is 35 then it will
36	G	Search find locate
37	Н	you your
38	I	compliments polite words courteous words
39	J	swear words rude words bad language
40	K	small words one or two letter words is, i, ip, etc
41	L	programs
42	M	number
43	$\mathbf{N}$	number
44	O	describe, translate
45	P	equal type words
46	Q	stop commands exit finnish etc.
47	R	is means are
48	S	ok and yes colloq adj (starting to change old FI codes
49	T	used by the CON process.
50	U	special character or word maybe a code
51	V	name
52	W	times *
53	X	divided by /
54	Y	plus +
55	Z	minus -
E.C.	1 .	

56 = slang or word used in taxing like errr or L8 for late etc.

57 = noun and pron

58 = command and what

59 = autoit script

60 = pattern that might be found in a sentence code looks in sentence code for pattern used in TFsub01 look in Pattern paragraph for more

61 = only use pattern for translation

62 = function request so you don't need a function word bypass it

63 = animal

64 = swear word

- 65 = chemical symbol
- 66 = course, slang, noun
- 67 = subject used by subject routine
- 68 = used by word in input expected routine put 10 or 11 into input
- 69 = used by word in input expected routine put 10 or 11 into output output
- 70 = web/internet URL address
- 71 = Neural Net code sets 7th bit 000000100
- 72 = Neural Net code
- 73 = Neural Net code put code into putininput to be reprocessed
- 74 = keywords for subject routine 'subject', 'about', 'saying', 'conversation'
- 75 = Keywords 'computer'
- 76 = Keywords find, locate from first input
- 77=ToDo word type 01 ALARM
- 78=ToDo word type 02
- 79=ToDo word type 03
- 80=music, sound, audio
- 81=image picture
- 82=contr
- 83=adjective and pronoun
- 84= attrib, adj
- 85= symbol,key,nmuonic
- 86 = weard un describable word
- 87= demons, pron
- 88=prefix assim
- 89=1st & 3rd sing past
- 90=organization 1
- 91=organization 2
- 92=emails used for search
- 93=business 1
- 94=business 2
- 95=business 3
- 96=RAS mind routines
- 97=name use when no more info required 23 has more specific commands
- 98=search keyword 'doc'
- 99=search keyword 'cd'
- 100=field 5 keyword like alwaysansweron or off etc...
- 101=1st and 3rd sing past of be
- 102=digital channel out on
- 103=digital channel out off

```
104 = RAS makes the word a RAS subject word highest priority in the program
```

105 = RAS makes the word a RAS reference word

106 = RAS makes the word a RAS action word

107 = int colloq

108 = digital in code

109 = cause to

110 = pv as in solar photovoltaic

111 = user defined

112 = user defined

113 = user defined

114 = noun colloq

115 = symb

116 = colloq adj

117 = interrog. pron

118 = Wheatstone Bridge generated letters

119 =

120 = WSB words

121 =

122 =

123 =

124 = user defined

125 = user defined

126 = user defined

127 = user defined

128 = colloq

129 = archaic

130 = suffix

131 = nne image scan code

132 = LM learning mode

133 = NNE codes important see note in NNE tags section

134 = Image Scan Functions generated code

135 = ST code if the codes are the same but not necessaraly in the same order then use the cell.

136 = ST Code if the codes are in the cell word then use the cell.

137 = ST Code if the codes are the same as the cell word and in the same order then use the cell.

138 = ST Code if an exact match of the cell word is found in the ST codes then use the cell.

Brain system cells are the cells 802 to 2000. These cells are used by various functions within the Ai program.

Cell number	Description				
----------------	-------------	--	--	--	--

 $oldsymbol{1}$  holds the length of BD and the positions of certain data within the BD

#### 801 unused

**802** location and name of today's log.

**803** computer name uses the present computer name set up in network.

**804** this entry is setup every time the Ai receives an input.

Field 7 = time, field 8 = date, file = file name, field 10 = days full name and field month.

**805** bad word response uses field 8 then puts the bad word at the end eg. i don't tword...

**806** start-path the folder the Ai program is in.

**807** help file location.

**808** train NN location of the train.txt file used to train the Neural Net.

**809** UDP information File field holds the codes that control what is sent and how input to output, i-i = input to input, o-i = output to input, o-o = output to output, Brain.

Field 11 is added to the sent packet before the END: command, this means extra eg.

ac:3000 activates Brain =Cell 3000 on receiving computer

in:110011111 9 bit code used for input of cell (overrides Brain input code)

if in: is in field 11 then the code following in: is used otherwise the input code is If in: is appearing in the receiving ai then set Brain Cell 809 Mode field to 0, also contents of

field 11

When sending if Mode = 1 nothing is sent if Mode = 0 then packets are sent.

The UDP packet is made up in the following way:

main text, password of the remote ai, how the text should be treated, ip number (packet eg.

hello world pw:1234do:i-oip:192.168.0.2en:

**QTF** query remote Brain this works in two ways you can either have the default qtf which

then sends any unknown words, words that have been added to the user dictional word type set to the remote ai. These word types are B A and # in the word is for Brain then the information is set back with the tag do:ret the Brain on the local nupdated or a new entry created. If either noun, verb, adjective, adverb is found ir then the word type is updated. For some reason LM on my server doesn't work puery another ai on another machine which then accesses information using the routines configured using Brain Cell 1602.

Make sure the mode field is set to 0 if not nothing will be sent.

**810** field 10 is the Browser string sent to the address bar on launch.

**811** EIIInt is used to control expected in input routine field 2 = number of times word, field 10 = eg. please answer question ... field 11 = give up phrase.

**812** ISC01 is fields 8 to 11 are image compare replies look at ISC02 entry 13 for the results are processed.

**813** ISC02 is fields 9 to 11 are image compare replies normally to output unless = 40 then readout error count. The input code is changed to sic error count divide **814** auto execute the word in field 8 is put into the input when the Ai starts. If the Tags field then the cell is activated at startup.

Field 7 is used to reset the BD to a new computer. If field 7 is set to "reset" then routine is run. reset is removed from the field after the reset which happens the nup.

If field 7 is set to "resetalways" then the full **CheckFix** routine is run every time keyword **CheckFix** for more detail. If there is something in the Tags field then tl startup

**815** output-failed field 10 = response field 8 = not sure?, field 11 = MoodC. Timer! Timer ^ Timer - Timer ~ Timer + if timer at top same then do confused r **816** used to control Ai font defaults, field 1=size 10 point, field 2=weight 300, fi Courier New, MS Sans Serif, Palace Script MT these values are set by the progra 10 can be changed. These fields alter the colors of the text and background field scrolling text field 8 = input boxes text colors field 10 = background color. The c value red green and blue. The value of each component can be anything between field 7 = rgb:0,255,0

field 8 = rgb:0,200,20

field 10 = rgb:0,0,0

**817** used to hold location of desktop in file field.

**818** controls maths functions, field 8 put in front of the maths answer field 10 is are more than 2 numbers in the input.

**819** STF search text file default settings, fields file and field 10 hold the filename holds the word to be put in input search-input, search-output, field  $11 = \text{the file } n_0$  holding more filenames field field 7 = name of the output file if 'htm' is found in

output file is wrapped in HTML code to make it into a web page. The file headcouthe HTML folder will be added to the output file in the head tags eg. <HEAD> (headcode.txt) </HEAD>. This is used to add extra code like Java script or style full location of the search results file overrides putting search results into readou 820, white space message

unused.

Reply Quality defaults field 11 holds a 23 figure number each number correstilers in settings.

d3in is digital in 3 do field 8 if it goes off field 10 if it is on.

d4in is digital in 4 do field 8 if it goes off field 10 if it is on.

d5in is digital in 5 do field 8 if it goes off field 10 if it is on.

bottom-right-button is the logo button at the bottom right of the screen f8 is when small display.

browser information field 10 holds the home page/default page sent to brow string is empty.

field 8 sent to output if ToPrint is the same as the bad word, this is useful if a bring the answer after the output has been sent.

unused.

repeatanswer repeat answer if the output is repeated and no other suitable an field 8 is put in front of the answer.

**831** Create HTML file field is the source file, field 10 is the output file. The sour tag is found then the Ai processes the tag then continues to read the file. At the n tags that can be inserted into web pages.

<aioutput> this tag is changed for the output from the Ai.

<cell:1234> activates the brain cell number following cell: the tag is replaced with a space added at the beginning and the end. These tags can be insert which isn't changed.

Auto Ai control used when Auto Ai is checked in settings. Also the mode se default mode so you can filter the input if mode field = 3 then Learning Mode (LLM is on entries 122 and 123 are also used. If the wheatstone bridge is connected that is used instead of random entries from the BD.

random-number-reset random number was below 40 so its bin reset to 33.

out-to-file used by Output to file Net OUT f9 put in front of output f10 put a input used to encode for html only works when create html checked in settings.

STM-noun used by short Term Memory STM if word is a noun sandwiched file field.

STM-verb used by short Term Memory STM if word is a verb sandwiched be file field.

STM-TF word found in STM and TF but its not a verb or noun sandwiched

file field.

STM-no-TF word found a few times in STM but not in TF ie. I don't know version NoSTM is words rejected from the Short Term Memory scan. Words in f11 the STM scan there must be a space on each side of the word for the word to be a space on subject set.

**841 to 899** subjects look in **Subject**.

ApplyMind-Referance used by ApplyMind routines.

ApplyMind-ActionToDo from ApplyMind routines.

**905** used when Ai in system tray used in menu puts cell name field into input pu **906** used when Ai in system tray used in menu puts cell name field into input pu **907** end\_of\_playlist when the play list ends anything in f8 it is put into the input has ended.

**908** Wheat stone bridge the keyword in field 5 is which output type, **wsbresistar** and 255 with wsb\_r\_ in front when a change in resistance is detected then the ne **wsbwave** an ever changing wave, the input signal is processed and enhanced by information produces letters, words and sentences the output can be fed into the to create an output by using the words already translated, **wsball** uses both wave output. If **putininput** in field 11 then WSB is put into input, **putinoutput** sends **swopinput** swops the analogue input channel, the default channel used for the W **909** output filter for QNV field 8 first bit "i think that" file field second bit "are" "with" field 11 fourth "do"

**910** used by Search Engine New (SEN). SEN sends the input to a web site like a extracts an output from the results sent by the search engine. Field 8 is the first p string, field 10 is the second part. The input is sandwiched in between. See SEN Algorithms for details.

**911** used by the CON process see CON in <u>Processing Algorithms</u> for details.

**918** sends a message to the name of the window in file field, used to send inform programs running on the computer. field 11 holds the what to do with the message output, i-i = input to input, o-i = output to input, o-o = output to output, qtf = que sends the input code of the cell. This function is tuned to work with the server. it LM download request or a QTF query, then the message is sent. On the test server

found then the qtf routine is use to initiate information download from web sites online dictionary, this process can take a long time on my server if it doesn't rece message then the php script fails to prevent this the message is sent on timer loop 30 seconds on the php script so if the information cant be found i set the timer loclocks before the 30 seconds is up.

**919** NNE 6p matrix files field 8 in front file file field is the file created by ai promatrix file. The in front and after files are matrix files containing nne entry numl **920** 6P Six Points fields 7 to 12 are the 6p responses.

**921** NNE image scan field 11 is the verification code 000010000 seems to work 00000000 into 000010000 file field is the matrix file used if image double clicke matrix file used if nnescanimage keyword is used to activate image scan 000000 000010000 because if all inputs are 0 then there is no positive value to set a "1" if the NN anomalies i've found whilst learning how to program NN. This means the image will be classed as a match to get round this field 8 holds the words to use. If field 4 = 69 and a match with the output code from the nnescanimage routine if then the words in the associated BD entry field 8 is put into the output.

If field 4 = 68 and a match with the output code from the nnescanimage routine i then the words in the associated BD entry field 8 is put in the input, if no match i output code from the routine is put into the input which can the be copied and pu manually. If NN learning is ticked in settings then the code is added automatical **922** used by LM if field 4 = 0 then field 8 is sent to the output, the following inposet the word in question field 4 to the number.

**923** LM prompts used to request information from the user field 8 is the word to in question, if there is one word in the next input then this word if in the BD is the word in question the cell is linked to the new word using an output node. file fiel more questions and uses the inputted words to set up more associations in the BI **924** Cell type exclude. Excludes the cell types from various outputs so sentence and other word types as defined in the cells type filed can be excluded from certain and NNE. Field 11 holds the word types to exclude default excluded word types: field 11 = 32 60 72 96 118 131 0 132 120 40 102 103 450

**925** special action the string in field 11 if found in input uses the special action retriggers the value to be increased by % in field 3 eg if field 3 = 25 amount = 25% **926** extension field 11 is the extension to use when looking for new image eg. fit **927** ftp (file transfer protocol) used for transferring files to a server using ftp propassword, field 8 user, file field is the filename to upload, field 10 is the address the path and filename of the uploaded file. If field 11 is empty nothing will be compared to finger used to change wsb words field 8 response if done.

**929** git (gap in time) if a gap in time is spotted like when the computer goes into

f8=output. If f4=68 then f8 sent to input, if f4=69 then f8 sent to output. The Air out the gap so if the Ai is running at midnight then the switch to the new day tright is the length of the gap it time to respond to in seconds.

**930** controls for the keyword scanimagefull field 10 is the color depth normally field 11 is the 2nd and 3rd color change if 10 then if the image has only 9 colors of the most found colors is changed. If there are more than 10 colors then the sec colors is changed to black the third filter is 2 x field 11 so if field 11 = 10 then 2x = 10 more than 20 colors then the third filter is used. normally set to 10 = 10 max around 1 colors to remove normal 3 max about 50.

**931** Learning Mode Auto LMA this routine finds information on the web and authe Brain. Its been tweaked to work with WorldWeb online dictionary if the wordinput code is set to 110000000 or 001100000 if not found. If a bad word is found settings then the search url is created this url is downloaded to the filename: ALI filtered using the Extract Words routine which creates the file: ALFile.txtx this for the focus word plus the contents of field 7. The line holding the word or a lin Operation field is used to set the contents of the detail field.

The coded tweak used for WorldWeb looks for the word noun, verb, adjective, a searching the ALFile.txtx file. if any of these words are found then the word type updated. There is some end of line filtering which removes the following 1. 2. nl adjective. The cell is checked for tags and the tag function is invoked if found, so work as expected though because the routine is invoked from an abnormal place another cell.

field 7 is added to the focus word after the focus word has been used in the searc field 8 is the first part of search string

file field is the output rejected if its the same as the contents on the field field 10 is added to search string after the focus word

operation = number of the line to use for the detail field after the line holding the field 7 has been found in the downloaded page

type = debug options 2 copies the text without filter 3 runs the ExtractWords profix the weard bug.

These are the settings for cell 931 to work with worldweb online dictionary.

operation = 1

field 7 = 1.

field 8 = http://wordweb.info/developer/SQL/search.pl?w=

field 11 = word to add to end of search string put a space here if having problem file

mode = 0 tells the routine what word to use

mode = 0 UDPBadWord variable used for online and results file search

mode = 1 field 8 used for DownLoadFile only local search string set to cell word

updated with new information if found

mode = 2 field 8 and field 11 used for DownLoadFile local search string uses fie ExtractedWords file. Word detail field will be updated with new information if for mode = 3 field 8 used for DownLoadFile only. Field 11 used as the web search search string when searching extracted files. Detail field will be updated with ne found

mode = 4 assume search string is already set so leave it as it is

mode = 5 leave web search string as is use field 7 only to search extracted text

mode = 6 use RAS code for web search sandwiched between fields 8 and 10, the word for local search

mode = 7 use RAS code for web search sandwiched between fields 8 and 10, the search

mode = 8 same as mode 2 but excludes lines that contain the contents, if any of f coded)

Some mode switches don't work on the default cell 931.

If initiated from the keyword AutoLearn then the word used in order is either the bad word or the cell word.

**932** Learning Mode RAS LM\_RAS default settings. Uses the AutoLearn routine produces no output then cell 932 values are used for AutoLearn.

If field 8 is empty then the process is bypassed

**933** user defined window

mode is the left position, operation is the top position, order is the window width height.

If this cell is activated then the window is closed then reopened useful if you walkindow

field 7 edit box color rgb:255,0,0

field 8 edit box text color rgb:255,255,255

field 10 window background color rgb:0,0,255

**934**, activated when User Window is left clicked. The coordinates of the mouse a process which then reads the image under the mouse then sets up the cells weigh the cell. The input code is generated from the average color taken from 162 pixe around the mouse pointer.

If you set cell action to **cell IN to cell OUT** then the resolution is lowered and bl better. The color of the pixels is 16 million to fit this value into a 9 bit code the redecreased this is why some different colors produce the same output.

**935** used by emotion. node to cell fields hold the most common words when the most commonly found word is in the, to cell 01 field. the cell numbers move dow words are found. Cell numbers that drop off the list are added to field 11 in cell 8 New. When a cell word is added to the subject list then the words found count is

done by adding the tag <remove> to the emotion field of the cell. The tag is acte cell is found. The Mode field holds the number of times the most found word had this number exceeds 100 then its reset to 0 this means the list moves down. If a vacuum to the cell is activated as the list moves down. If a vacuum to the cell is activated as the moment the cell is activated as the moment the cell is activated. The input code is

changed to the number of times found before activation. At the moment i'm usin <nodewords-ifa> which uses some of the found words for the output.

**936** used by **imagescanvertical** keyword. The Order field is the top starting posi image scan if at 0 then the whole image is scanned the range is between 0 and 20 number the faster the scan but smaller image is scanned. Mode field if 1 then the lines are drawn. The input code is altered with 1 meaning an object very close to vertical line has penetrated the 200mm area in front of the robot. The position of the positions of the object i'm not sure yet how many 0 in the input code are required through safely. Field 10 is the tilt value used when adding the data from the inmap this value gives the software the value to control perspective.

**937** input code used by compass to store angle value.

**938** input code used by compass to store pitch value.

**939** input code used by compass to store roll value.

**940** used by the alarm keyword. In cell fields are the cell numbers of cells with these cells are checked every timer loop. There is a maximum of 9 alarms more found in the keywords section.

**941** used by defrag. If Defrag is pressed from this cell then special filtering is ap The type value is the type of word to exclude from the new database. Warning some reason the filter needs to be run a few times to get it to work.

**942** node to cell filter if 942 is activated then the clean in cell values are checked some reason there are lots of references to a cell that might be causing unwanted routines mostly spotted in NNE 6P. The **in cell** value in cell 942 is the cell numb change to 0. At the moment it filters cells 2000 to end of file. if set to -1 then the with -1 will be reset to 0.

**943** activated when the user window is selected / has focus and a keyboard key i code is set to the value of the key pressed before activation. The cell will only ac the key has been released

**944** activated when the user window is selected / has focus and a keyboard key i pressed. The input code is set to the value of the key pressed before activation.

**945** activated if the mouse is right clicked in a part of the user window that isn't control.

**946** Used by keyword **ImageScanVerticle**. If **ImageScanVerticle** returns 0 gree considered that if there are no green spots then there is something wrong. If the r wall or there is no image then the green count is 0 so the robot thinks the way ah

might not but in another situation it might be true. The input code is altered to 11 count otherwise set to 000111000

used by direct X windows, mode is left coordinate of window, operation is the standard by the image window, mode is left coordinate of window, operation is the standard by the image window, mode is left coordinate of window, operation is the standard by the image window, mode is left coordinate of window, operation is the standard by the sta

**949 to 1000** reserved for future use.

Cells 1000 to 2000 have their input code connected to various functions.

cells 1000 to 1175 are altered by the NIOtoNNE variable when these functions a input code is incremented.

These input nodes are also connected to the K8055 input/output card and to the I The event scanner scans cells 1000 to 2000.

Restart.

Startup.

Open settings.

Open BD.

Received messages.

field 8 is the RAS output.

Put in input.

Patterns in short term memory.

Sentence code in TF, field 8 is the Sentence code in TF and TF outputs.

Pattern output.

TempToPrint found in TF.

field 8 is the Parameters output.

System commands.

Adverbs.

Adjectives.

Verbs.

Nouns.

If f3=3.

If f3=2.

System commands.

File moves.

File deletes.

Show image.

Read file.

```
1025 Read file in field 10.
```

Read file in 10 into input.

Read file in field 10 full path.

Read file in readout directory.

field 8 is the NNE output.

1030 Know names.

field 8 is the QI output.

Question noun verb. Field 8 is the QNV output

Second part QNV.

Mode 5.

Long file read.

Read file to input OK.

Read file to input FAIL.

Read file OK.

Read file password OK.

Read file FAIL.

field 8 is the 6p output, field 7 is 6p2 output, field 11 is 6p2 ex output

Say print. Field 8 SayPrint output the main output inserted just before 1178 SayPrint routine

Mode 7.

Toprint buffer.

Toprint buffer bypass block.

Write to log OK.

Write to log FAIL.

Write input to file OK.

Write input to file FAIL.

Write output to file OK.

Write output to file FAIL.

Write HTML file OK.

Read top HTML file OK.

Read top HTML file FAIL.

1055 Read sandwich file OK.

Read sandwich file FAIL.

pass value, the pass number is recorded to the input code before the input is

Read bottom HTML file FAIL.

Read bottom HTML file OK.

Read bottom HTML file FAIL.

Write HTML file FAIL.

Keyword stopnetwork.

- Keyword startnetwork.
- Keyword putininput.
- Keyword put9ininput.
- Keyword writetofile OK.
- Keyword writetofile FAIL.
- Keyword aion.
- Keyword aioff.
- Keyword cleartoprint.
- Keyword doesfileexist YES.
- Keyword doesfileexist NO.
- Keyword iffileexist YES.
- Keyword iffileexist NO.
- Keyword autoit read OK.
- Keyword autoit read NO.
- Keyword readcontroltf.
- Keyword activateudp.
- Keyword sendscreen.
- Keyword sendtobrowser.
- Keyword filetoinput.
- Keyword filetooutput.
- Keyword expectedininput.
- read message data store.
- RQW Reply Quality Weight.
- 1085 Put In Data Store.
- Translation Weight String.
- PTTISTMOTPTOM (put TF translation in short term memory of the patter
- SubNameFound, student file open OK.
- SubNameFound, student file open NO.
- lesson subroutine.
- scan desktop and put all .lnk files into TF.
- lesson subroutine.
- Auto Ai..
- Auto Ai.
- Autoit check YES.
- Autoit check NO.
- Learning mode file read OK.
- Learning mode file read NO.
- best match in dictionary.
- word added to user dictionary.

```
1101 problem writing to user dictionary.
```

field 8 is the learning mode part 2

looking in user dictionary.

problem reading user dictionary.

UDP Initialize Winsock2 DLL.

UDP Create TX and RX sockets.

UDP bind and configure the RX socket.

UDP Data packet TX packets sent.

UDP Data packet TX packets sent.

UDP Invalid socket.

1111 UDP Data packet RX received packets, field 8 is UDP packet new message

UDP password ok.

UDP i-i or o-i.

UDP i-o or o-o.

UDP closedown routine.

send screen using udp.

receive screen using udp.

open web browser.

see if browser is open.

open browser program.

pattern recognition noun or pronoun found.

adverb or predictive found.

field 8 is the NNEB output.

field 8 is the LM output.

field 8 is the NNEC output.

field 8 is the DT output.

field 8 is the DID output.

field 8 is the sentence code.

memory scan rejected for the word.

memory scan word found in stm all words.

memory scan word found in stm words larger than 2 char.

memory scan word found in stm words larger than 2 char and more.

look in subject, field 8 is the Subject output.

not in BD.

noun found and word in BD.

verb found.

subject word found.

expected in input.

1138 expected in input something to do with the file field.

- expected in input 2.
- expected in input 2 else.
- number of bad words found.
- read log.
- problem reading log file.
- search text file.
- problem reading search file.
- download file routine activated DownLoadFile.
- used to stop graph drawing when idle set to zero on each input then increm to 0.
- **1148** Download file, error creating file.
- **1149** Download file, error downloading file.
- keyword found.
- playplaylist.
- Mp3Sound play mp3 number of tracks played.
- NNE Matrix file read FAIL.
- ReadCounter(1) USB card.
- ReadCounter(2) USB card.
- incoming window message.
- USB card analogue input ONE.
- USB card analogue input TWO.
- Analog 1 average.
- Analog 2 average.
- Chakra 1.
- 1162 Chakra 2.
- Chakra 3.
- Chakra 4.
- Chakra 5.
- Chakra 6.
- Chakra 7 Thought.
- Process time, time it takes to process an input, seconds converted to binary Gap In Time (GIT).
- Field 8 is the Pass value.
- changed inputs.
- changed outputs.
- day.
- month.
- year.
- minute.

**1175** hour.

**1176** digital channel in last 3 nodes used for inputs 3,4 and 5 eg. if button 3 is proposed on the proposed of the proposed o

**1177** activated if face is on used to control face drawing.

**1178** activated during say print routine so you can do more with the output. If the has been invoked from a brain cell activation then cell **1178** isn't activated in ord corruption in brain cell activation flow.

Activating this cell at the SayPrint routine enables much more flexibility with the new way of processing the output is due to replace the write output to file field in code records nothing at this moment but it still resets to 000000000.

**1179** Brain Cell Errors records the number of cell word entries that are empty. That track down bug that sometimes causes Brain Cell entries to disappear. This is used that are debugging. In cell value node 1 is the number of empty entries node 2 to number of entries with errors.

The other 7 numbers are the first 7 cell entry numbers that have errors.

**1180** OpenBrowser.

**1200** + word codes from field 4, eg 1200 + 2 = 1202 if field 4 of an entry is 2 the word is a noun. At this time there are 130 defined word types i am experimenting for document analysis.

**1500** input code = code created from Ai input set in sub routine: NNUseAiInput, n2=noun, n3=adjective, n4=verb, n5=adverb, n6=you yours, n7=question, n8=sc and other things!

**1501** nne default settings used when a word is added automatically to the nne. Weights file is read, if the file cant be read then a new file called defalt.nnw is created input code nodes 1 to 8 used to set digital channel outs 1 to 8 if keyword **d**efact.

activated.

**1505** cell activated every time activation if keyword nneontimeron is on, nneontiactivation.

**1506** cell activated when a new window or udp message is received. Only activa keyword has been used to turn cell activation on.

**1507** cell activated every timer loop if keyword nneonloopon is on, nneonloopof activation.

**1508** default cell if DT learning cell = 0 then its changed to 1508.

**1509** the input code is created from the value of the characters in the sentence co cell 1500 but the code is generated in a different way. 1500 is limited to certain vaffected by all types so has a much more varied value than 1500. Field 8 holds the **1510** USB to DMX activates cells to dmx scan, nne cells input code is used to sea cell between numbers 1511 to 1599 is activated then the scan is activated.

**1511** to 1599 DMX channels 1 to 88.

**1511** DMX channel 1 input code used to set value bits 1 to 8 used if bit 9 set then channels 1 to 88 are used.

**1599** DMX channel 88.

**1600** keyword searchfilein9 if string found input code set to 110000000 if no 00 or if string is found the input code is changed to the number of times found other 000000000. If cell entry field 9 has <log> in it then then today's log is used for s **1601** NNE event, the NNE is constantly receiving inputs this cell records the event code is the number of events. It is updated every time there is a change from the **nneeventon** turns on activation of the cell when an event change value is recording turns off cell 1601 activation.

**1602** activated on input if **Activate 1602 on inpu**t is ticked in settings. The cell is before the input is processed and before other functions that happen on input. Ce to the input before 1602 is activated.

**1603** analog output 1, the input code is used to set the analog output value on the **1604** analog output 2, the input code is used to set the analog output value on the **1605** Grammar filter in the Output filter. The Grammar filter sorts the outputted of correct english. Following grammatical rules the input code of cell 1605 contrare filtered.

If set to 1 the filter is used for the specified process.

wsb	6р	lm2	nnec	dt	did	lm	qnv	number
1	1	1	1	1	1	0	1	0

field 11 are words to exclude from the output. This filter is the last process in the repeat spaces and any words in field 11 from the output, Useful if web codes or j through all other filtering processes.

Operation field is the percentage value used by Output Filter Function 4 (OF FN OF FN4 is higher than the percentage value then the OF FN4 output is used othe from OF FN2 is used instead. If Operation field is 0 then it is assumed that it has percentage value is set to 50%.

**1606** Sentence Types, field 7 is the subject word, field 8 is the object, field 10 is The in cell fields hold the cell numbers of the previous 9 subjects and objects.

**1607** input code used by scan image functions for mode 18 dark side of gray scale **1608** input code used by scan image functions for mode 18 medium of gray scale **1609** input code used by scan image functions for mode 18 light side of gray scale **1610** input code used by scan image functions mode 19 left four bits represent versensitivity right four bits represent horizontal line detection sensitivity.

**1611** activated when a match is found by image scan functions keyword ISF. The

number of times the code has been found. The SwopVar variable is set to the ISF in the code then cell 1616 is activated instead.

**1612** input code holds the light dark count of the image, values over 255 are ligh **1613** input code holds the vertical code created by a ImageScanFunctions keywowhen the type value is set to 1 it shows where in the vertical plane something is  $\mathbf{1614}$  the input code is the emotion value, numbers higher than 5 are positive belowalue is set by the emotion tags  $\langle v:5 \rangle$  where the number is the associated emotion  $\mathbf{1614}$  in  $\mathbf{1614}$  then the Word Swopper (WS) output is used if no emotion is found, only the emotion is used if mode = 2 then only the WS output is used

**1615** used by the Word Swopper (WS) process. Field 8 is the WS output. The W verbs in the inputted sentence. The nouns and the verbs used are the next noun o BD.

EMS: is the id in scroll output.

**1616** similar to cell 1611. The cell is activated when a match is found by image subut is only activated if the object tracing processes completes giving a code 2w. sensitivity, the number here is multiplied by 10 then if the number of pixels used is higher than this number then the cell is activated. The more pixels the bigger to **1617** this cell is used for the default settings of added cells when added by <look SwopVar needs to be set before this will work.

**1618** swop cell used by the = process when changing a word to another word usi dog=cat will swop the two cells so the cell numbers point to the other word. If th found then the cell word of the first word is changed to the new word.

**1619** Decision this function works only when the tag <i=cell:1619> is used. Who decision process is activated. The process uses the analog input or the random numake a decision.

The signal is sampled 49 times (7x7) if the signal tends to rise then the answer is move down then the answer is NO

If the Mode field = 1 then only the random number generator is used.

If the Operation field is 1 then channel one is used, if anything else then channel The input code is changed if

YES = 1100000000

NO = 001100000

Ambivalent or both same = 000011000

Both 0 or no signal = 000000110

**1620** used for default RAS cell when a new RAS word is added to BD some field process.

**1621** next available cell.

The following cells contain outputs and inputs created when an input is processed.

1006 field 8 is the RAS output.

1009 field 8 is the TF outputs.

1010 field 8 is pattern output.

1012 field 8 is the parameters output.

1029 field 8 is the NNE output.

1031 field 8 is the QI output.

1032 field 8 is the QNV output.

1041 field 8 is the 6p output.

1042 field 8 SayPrint output the main output inserted just before 1178 is activated in the SayPrint routine

1102 field 8 is the learning mode part 2 update help with this

1111 field 8 is UDP packet new message.

1123 field 8 is the NNEB output.

1124 field 8 is the LM output.

1125 field 8 is the NNEC output.

1126 field 8 is the DT output.

1127 field 8 is the DID output.

1128 field 8 is the sentence code.

1132 field 8 is the SUB subject output.

1168 field 8 is the Pass value.

1509 field 8 holds the input.

**Keywords** The Keyword field is used to associate a Keyword to a Cell Word. When the Cell Word is found in the input then the Keyword function is triggered.

To stop a cell word being used in the output then set the Type field to 40. If a cell with a keyword is activated then the keyword is only used if there is a keyword tag in the tags field.

If there is more than one keyword in the input then the fist keyword is the one used.

If the keyword call comes from a cell activation using the tag <keyword-if >then the keyword of the cell will be used even if the keyword variable is already set. The phrase UNDER DEVELOPMENT next to a keyword means the function may not work as expected or the way it works may change. It could also mean the function is out of date and other functions have superceded it.

**AiWindowHide** hides the Ai so its only visible as a process.

**AiWindowShow** restores the Ai window NOTE if you have no communication with the hidden Ai then you wont be able to access it. The only way to close it is to end the process in Task Manager.

**Alarm** field 7 = alarm time, format: HH:MM

The cell is activated if the time is found. Cell 940 in cell fields hold the cell numbers of cells with an Alarm Keyword only 9 alarms can be set. This is done like this to cut down on processor usage. Cell 940 is updated when the SaveBrain process is activated or when update is pressed.

**AlarmSet** This keyword like alarm is activated if the time is the same as the value in field 7. Field 7 is set from information in the input. If in the input there is a word with the keyword AlarmSet then the two numbers in the input are used to set the alarm. The first number is used to set the hour and the second sets the minutes, the two numbers are then used to set the alarm time in field 7. The words after the numbers are assumed to be the message and these words are inserted into the words-ifa tag which is then put into the tags field followed by the contents of field 11. When the alarm time in field 7 is met then the cell is activated. If there are no numbers in the input then field 7 and the tags field are cleared.

E.g. if the input is: alarmset 14 42 alarm time then the fields are set thus, note the keyword and field 11 aren't altered field 11 <sayprint> keyword AlarmSet

tags <words-ifa:alarm time><sayprint> field 7 14:42

**AllowRepeat** allows an answer to be repeated, works the same as f3 TF = 25, this gives another way of repeating the output.

**AlwaysAnswerOff** turns off Auto Ai on bad RQ.

**AlwaysAnswerOn** activates Auto Ai on bad RQ default is off.

**AOutNNEOff** turns off Brain to analog outs.

**AOutNNEOn** the input code of cell 1603 and 1604 is used to control the analog output value a1 and a2 on the k8055 card.

**AutoIt** autoit.exe must be in the path and .aut associated to it, then field 8 is the autoit script to run, field 7 is the reply if script is found. This performs a system command on field 8 and looks in the scripts folder for the AutoIt script to run. **AutoLearn** this keyword uses the DownloadFile and ExtractWords processes when a bad word is found. The file ALFile.txtx is created from the download and extract process. See System Cells cell 931 for more details on how this works.

The Order field is a secondary control if Order = 1 the contents of the ToPrint variable is used if it holds something this overrides other things. Here are settings for Wikipedia.

field	contents
tags	<keyword-ifa></keyword-ifa>
mode	4
operation	10
keyword	AutoLearn
field 8	http://en.wikipedia.org/wiki/Special:Search?search=

**Beep** file field is the frequency field 10 is the duration, processing stops whilst the sound is produced.

**BrightSpot** counts the number of bright spots on the image sends the answer to output, start of the find brightest area in a room when viewing with a web cam. **BrightSpotCompair** compares the number of brightspot's with the number generated by a BrighSpot keyword if there are more bright spots then field 10 is put

into the input if there are less brightspot's then field 11 is put into the input. The input code of the cell is changed to the results of the function 000001111 means there are exactly the same number of bright spots 000000011 means there are less bright spots

000001100 means there are more

**BulkWrite** writes cell data to a text file. Like pressing the TO TEXT button but this keyword bulk writes the data from cell number in field 7 to the cell number in field 10. The file field holds the name of the file to write to. A new file is created and the data added to it but if the filename is NewBrainData.txt which is the file used by the TO TEXT button then the cell data is added to the file without creating a new file.

**ChangeGetStartPath** converts < getstartpath > found in the BD to the proper address.

**ChangeToGetStartPath** converts address in the BD to <getstartpath> only address that contain the start path will be converted.

**CheckFix** activates the CheckAndFix routine that normally runs when the Ai builds a BD when no existing BD is present in the data folder. If run as a keyword only system cells not set will be updated. The cells updated are: 809 f9 to o-o

810 set to spud ai web site

815 set f7,f8,f10,f11

818 f8,f10

819 STF

822 RQ defaults

841 SJ Subject

860 Emotion

861 SJN Subject New

910 SEN

920 6p

922 LM

923 LM

924 NNE exclude

931 LM web add

**932 LM RAS** 

2000

2001 set to "is" for QI

2002 set to "what" for QI

2003 set to "indexfiles"

2004 UpdateBrain Keyword

indexfiles is activated to give RAS and PAR something to work with

**CleanNode** resets a in cell field to 0. Sometimes words start to appear that aren't wanted this is sometimes caused by repeat references to words CleanNode is a way of removing these words by removing the connections to them.

**CleanTrainFiles** this removes unused training files. Cells with <f: in the emotion field are assumed to be used so are left out, also cells with three or more lines of training data and files lower than 2001 are left out of the filtering process. Other files are deleted. Every time a new word is added to the BD then a new training file is created these can build up to large numbers and if unused aren't needed. a new file will be created if the cell is trained for any reason.

Clear All Digital clears all digital channels.

**ClearDigital1To4** this clears digital channels 1 to 4 at the end of the processing cycle this enables faster switching. If ClearDigital1To4 is in Keyword field of the word that turns the channel on then it will stay on for a shorter time.

**ClearDigital5to7** clears digital channels 5 to 7.

**ClearDigital1to4Immediatley** clears channels immediately as soon as this key word is found.

**ClearDigital5to7Immediatley** clears channels immediately as soon as this key word is found.

**ClearInputs** clears all input buffers including all PutInInput variables.

**ClearPutInInput** clears PutInInput variables.

**ClearTagOnce** clears the tag once variable.

**ClearToPrint** clears the ToPrint variable which is used to hold the output.

**CloseWindow** close the window name in field 9.

**ConnectToInternet** uses the InternetAttemptConnect command in wininet.dll. **CreateHTML** The File field is the source file, field 10 is the output file. This works in the same way as cell 831 but uses the information in the cell instead of cell 831. The tags are placed in the source file and are acted upon when read by the program.

<aioutput> this tag is changed for the output from the Ai

<cell=1234> activates the brain cell number following cell= the tag is replaced
with the output from the cell.

<file=filename> where filename is the full path of the file to insert, usually a text or html file. The tag is replaced with the contents of the file if it exists.

These tags can be inserted in other text which isn't changed. Best practice is to use only one tag in a line.

DateTimeToFile writes the date and time to the file name in file field.'

**DebugMini** sends the scroll output to the debugger only.

**DebugOff** turns off outputs to debugger.

**DebugOn** turns on outputs to debugger you need to have a debugger program running. Information sent to the debug program is more detailed and includes many messages that don't appear any ware else.

**DInActivateOff** de-activates cell 1176 activation on digital in.

**DInActivateOn** activates cells 1155 or 1154 or 1176 when K8055 digital channels 1,2,3,4 or 5 are activated. Input code of 1176 is altered according to the digital ins rightmost node = d5.

**DisconnectFromInternet** disconnects internet connection.

**DoesConnectionExist** does internet connection exist if yes field 10 if no field 11 if Operation or Type fields are 68 then the output is put into the input.

**DoesWindowExist** if the window name in file field exists then field 10 is used for the output, if not then field 11 is used. If Operation or Type field is set to 68 then the output is sent to the input.

**DOUTHNEOn** allows the input nodes 1 to 8 of cell 1504 to activate digital outputs 1 to 8.

**DOutNNEOff** turns off Brain to digital outs.

**DTLearn** uses a keyword to trigger the DT learning routine. The keyword is removed from the input then the input is put into the output box and the DT learning routine is activated. The Ai continues to process the input after the DT learning routine has finished. This is used so you can update using DT from the input box rather than typing directly into the output box useful when training the Ai remotely through a web page or other input. If there is an entry if field 7 then this is removed from the input as apposed to Cell Word, this is useful when activating using say a pattern code.

If "is the sentence code for" was in the previous output then field 8 of the sentence code or pattern is changed instead.

**DumpSTM** writes the short term memory to the file DumpSTM.txt found in the Ai directory.

**Encript** this changes the contents of the ToPrint output variable using the encripter function. Normal text will be encripted, encripted text will be decripted. Field 7 is the key. The function checks its output, if there is a problem then it attempts up to 5 times to encript if it fails then the scrole output will say error.

**EventActivate** activates the cell if an event occurs.

**ExpectedInInput** this tells the Ai to look for the words in field 8 or file field in the next input.

**ExportBrainToCSV** exports the BD to a spreadsheet. Entries 2000 upward are exported.

**ExtractWords** extracts the words and numbers from a document holding tags or scripts like a web page. It works as a simple parser for web pages so the document can be used as source material by the Ai. A lot of web pages these days are made up of scripts and objects that source data directly from databases, these type of pages have very little real text so the process sometimes produces a

blank page. Words are extracted from the file name in the file field. The new file is the same name but with x added to the extension.

If the Operation field is set higher than 0 then the extracted file is scanned and the number of lines down as specified by the Operation field is sent to the output. ExtractWords also sets up the variable used by the KnownWordsTwo keyword.

**F7ActivateF10Millisec** activates the cell number in field 7 in field 10 milliseconds. This function uses its own timer.

**F8ActivateF10Millisec** activates the cell number in field 8 in field 10 milliseconds. This function uses its own timer.

**F8ToInputF10Millisec** starts the timer to the value in field 10, in millisecond then puts contents of field 8 into input when the timer is up.

**Face** goes directly to face drawing routine using the cell number that activated the keyword. The output code is used to move the eyes and mouth. Each node operates an element of the face.

As your looking at the face the eyes pupil are moved to the position using the output code in the following way

node 1 pupils to bottom right

node 2 bottom middle

node 3 bottom left

node 4 middle right

node 5 middle middle looking straight at you

node 6 middle left

node 7 top right

node 8 top middle

node 9 top left

the mouth is made up of 4 elements which are either up or down

node 1 left top line left 0=down 1=up

node 2 left top line right 0=down 1=up

node 3 right top line left 0=down 1=up only if left top line right is in same place, stops the mouth looking weard

node 4 right top line right 0=down 1=up

node 5 left bottom line left 0=up 1=down

node 6 left bottom line right 0=up 1=down follows node 7

node 7 right bottom line left 0=up 1=down follows node 6

node 8 right bottom line right 0=up 1=down

The expressions and the associated code might be used as the emotion variable which is a new idea to give more information on words.

**FaceOn** turns on face drawing in image window.

**FaceOff** turns off face drawing.

**FastScript** turns off the main Ai timer and reads the script file name in file field into the Ai if file field is empty then the file name in field 8 is used. It is assumed that the script is in the scripts folder this is for compatibility with older Ai databases. Normal processing continues after the script has been read. If timer: followed by a number is in the script then the value is the time between each read of the line the default setting is 500ms.

**FileConnections** this uses the CON process this scans files for sentences with the best match to connections with the input. The File field is an index file, this file contains file names which are opened and read. The lines of text are then scanned against the words in the Connection variable and if a match is found then the count is increased. The sentence with the highest count is used for the output.

**FileCopyNewName** copies the file in the file field to the folder in field 11, a new filename is created using the number in field 7 which is incremented after the file has been created. If field 11 is empty then the file is copied to the Ai folder. If the file already exist then it is over written. The file extension remains the same.

**FileDownload** The external program dlf.exe is used to download the file from the server name in field 10 to the file in the file field. Using downloadfile allows the Ai to continue processing if you want processing to stop whilst the download is happening then use the keyword DownLoadFileWait. The file dlf\_output.txt holds the results of the download.

**FileDownloadWait** Field 10 is the full web address of the file to down load. If mode = 1 then the contents of the ToPrint variable are added to the address in field 10. The program waits for the download to finnish before continuing processing. Field 7 is the cell to activate when the download completes. dlf\_output.txt is deleted before the download starts. The process causes the Ai to freeze during the download which is useful if the AI then acts on the downloaded file but if you want the AI to continue processing use the FileDownload instead. This keyword uses the external program dlf.exe to download the file.

**FileExist** checks to see if the filename in the File field exits, if yes then the contents of field 7 is used, if the file doesn't exist then field 8 is used. If Mode = 1 then the output is sent to the Ai output otherwise the output is put into the input.

**FileGetSort** down loads the file in field 10 to the file name in File field then runs the FileSort process and then copies the resulting file back to the file in the File field. Field 11 is used for extra commands CheckWords and ExcludeWords

CheckWords only uses words that are in the BD

ExcludeWords exclude words in Field 7.

**FileIndex** sets up a search file that can be used by FileIndexSearch or as a play list. The filename in field 10 is created, this contains the filenames of the files in the folder. File field holds the folder to start the search, with the wildcard \*.\* instead of a filename eg. C:\temp\Ai\\*.\*

Field 11 holds the type of files to include eg. doc DOC txt TXT etc.

**FileIndexAppend** like FileIndex, but instead of creating a new file it adds indexes files to the existing filenames file in field 10. File field is the folder to start searching for files, field 11 is the file types to include.

**FileIndexSearch** The File field contains the filename of the Index file. An Index file is a file containing a list of file names which are searched. The contents of the ToPrint variable is searched for in the indexed files.

**FileRead** this process apples some filtering functions before reading the file to the input or output.

Mode 1 = read to output.

Mode 0 = read to input.

Field 7 is the line in the file to start reading from if empty or 0 then nothing will happen.

if CheckWords is in field 11 then the words in the file are checked for in the BD. Words not found in the BD are rejected from the output.

Field 10 is the cell to activate after the file has been read the TimerLength variable is altered to give enough time for the readout script to finnish before the next line is read. The length of the output is multiplied by 70 to give the timerlength in milliseconds. TimerLength is returned to its original value once file read completes.

**FileReadLine** reads lines in a file sequentially until the end of the file. field 7 is incremented. Field 7 is set to one and the output set to field 8 if it reads past the end of the file. Each line is read every time the keyword is found. If Mode = 1 then it is assumed that the text is a number and the associated cell number is activated, the cell will only activate if the number is higher than 1 and no larger than the last entry in the BD.

**FileSafe** this function reads and adjusts a file making it safe to read by other processes. This checks line lengths which can overload variables. There is still a limit in line length this is 1000000 characters if more than this then a GPF will be caused. The text is converted to lower case and full stops comers and semi colons are removed.

**FilesSafe** this uses the index file in the file field to make safe multiple files. The index file contains the file names to make safe. This uses the FileConnections

function so the Connection variables are also set.

**FileScanForNew** Field 11 is the extension to look for eg. .txt. The file field is the filename to search for eg. C:\temp\ftp\\*.\*

**FileScanForNewJPG** looks in a folder to see whether an new .jpg (picture file) has been added, if it has it is copied to the image folder and called newimage.jpg all the jpg pictures in the folder are then deleted so the next image that appears will be the correct one to copy when monitoring a picture folder that is receiving images from say a motion detecting web cam. The file name in file field needs to have \*.\* at the end eg. C:\temp\ftp\\*.\*

If there is a new file then field 8 is put in the input if not then field 10 is put in the input.

**FileSearch** searches for the word in field 10, if field 10 is empty then the cell word is used.

File = filename of file to search the last found entry is used for the output. Type = 68 then the output is sent to the input.

Mode = 1 then the output is used for the search. The output is changed to the result if found. If multiple outputs are found then they are also added to the output up to a maximum length of 200 characters.

Field 7 = words to exclude from the output, used to filter out things like web code or trigger words.

**FileSearchF7** the word in field 7 is searched for in the filename in the file field. **FileSort** uses the FileSafe function before sorting the file. Using FileSort is another way of extracting information from say a web page. This is normally done using AutoLearn but this keyword uses a simple parser to process web pages, the parser is not capable of extracting text from complicated pages. Using FileSort extracts the known words from the file without parsing the file. The file in the file field is sorted into another file with the same name but with an x added to the filename extension. Words in the read file are joined together so new lines of text are of a similar length.

If CheckWords is in field 11 then the words are checked for in the BD and only know words are written to the file.

If ExcludeWords is in field 11 then words in field 7 are excluded from the output. This is useful if random bits of web code get through the filtering process.

**FileWrite** puts the contents of field 8 into the file in the file field.

**FileWriteInput** writes the input and the contents of field 10 to the file in the file field. Mode controls the file writing process.

Mode = 0 append text to the file.

Mode = 1 create a new file with the text.

**FileWriteOutput** writes the contents of the ToPrint variable and field 10 to the file in the file field. Mode controls the file writing process.

Mode = 0 append text to the file.

Mode = 1 create a new file with the text.

If there is a colon in the filename then it assumed it is a full path filename if there is no colon then it is assumed to be a filename only and the file is in the main ai folder.

**FindDouble** this finds double entries created by the evolution bug the file name in the ai folder called FDFile.txt holds the words with double entries in the BD. **FolderIndex** scans the folder and sub folders starting at the folder in file field and writes the folder names to text file 'DirNames.txt' which can be found in the data folder, use wildcard \*.\* at the end of name. E.g. c:\temp\\*.\* This means all files in the folder are read and directories are picked out and added to the list without the \*.\* the keyword wont work properly.

**FolderSearch** sets up a search file that can be used by FileIndexSearch. The file searchtemp.txt is created in the data folder, this contains the filenames of the files in the folder. Field 10 holds the type of files to include. File Field holds the folder to search \*.\* is required if all files in folder are to be included eg. c:\temp\\*.\* will search all files in temp any files with an extension present in field 10 will have there name added to the searchtemp.txt file. The extensions are case sensitive so if you are looking for say zip files make sure you include ZIP in field 10 as well. If nothing is in field 10 then all filenames and folder names are added to the list.

**FTP** File Transfer Protocol. Field 7 is the password, Field 8 is the user id and field 10 is the ftp address.

**HostToIP** field 10 is the host name to find the IP number of, field 8 is the IP number found. This function uses some of the UDP processes so Enable UDP needs to be ticked in settings. To Enable HostToIP but disable UDP un-tick RX and TX.

**If8Or9Do10Or11** if the word in field 8 is in the next input do field 10. If the word in field 9 is in the next input do field 11. Sometimes referred to as expected in input two. Field 4 can be used to control where the output goes if field 4 = 68 then field 10 or field 11 is sent to input, if field 4 = 69 field 10 or field 11 is sent to the output.

**ImageCompair** scans the display and compares the collected data with the data from a previous ImageScan.

**ImageCompairZone** similar to imagecompair but isolates the zone of most change and sends the name of the zone to the input. At present there are 3 zones zone0 = little or no change error count < 10, zone2 left half of image zone3 right

half of image. The code for 5 zones is embedded in the program but in order to make it work with the USB K8055 card moving a web cam the zone routine has been desensitized!

If this keyword is invoked from a cell activation then the timer may need to be stopped first. The timer can interrupt processes and cause things to go weard. If the keyword is invoked from an inputted word then the timer is automatically stopped before the input is processed.

**ImageInstructions** this keyword is used to create instructions from an image. Its used by the robot to draw an image. The process uses information created from ImageScanFunctions.

field 11 is the ISF functions to use note they need to have also been used by the ImageScanFunctions keyword to work.

Mode = left start position

Operation = top starting position

Order = number of pixels down

Type = number of pixels along

File = filename of the created text file which consists of instructions that can be read by say the ReadLine keyword.

Field 7 = draw a dot instruction

Field 8 = leave blank

Field 10 = end of column

detail field = step number, due to the lack of fields the detail field is used. The step number moves the read pointer along by the value, in effect it reduces the number of pixels read and thus the number of instructions, a bit like zoom.

**ImageLoad** this keyword loads an image into the DX Window. Direct X (DX) is a way for programs to use the resources of the graphics card. The Ai uses the DX window to process images doing it this way speeds up processing time. The keywords ImageLoadScan and ImageScanVertical use the DX functions if the DX window is open if not then they use the normal image window. The robot uses ImageLoadScan and ImageScanVertical using the DX window reduces image processing time on my machine from 4 seconds to under a second.

**ImageLoadScan** file field is the filename of the image to load if empty then the existing image is used. The image is scanned and loads the image variable which can then be used by other functions like ImageScanFunctions. If the DX window is open then DX functions are used to scan the image, if closed then the scanning process is done in a separate thread which scans the normal image window. Using the DX window is much faster.

Normally the image is displayed in the image window and the DX window the Mode field changes the way the image is displayed.

Mode = 1 load image into DX window only.

**ImageScan** scans the display and retains information on the image.

**ImageScanFunctions** this is a collection of image processing functions. They use the DX window which needs to be ticked in settings to work.

The Operation field is the intersection value, this is the maximum number of times the intersection can happen before an intersection is counted this effects the input code.

Order field is the sensitivity control for the function.

The Type field is the intersection line, the point in the image that the intersection occurs.

Field 11 is used by auto calibrate, if there is a number here then this is used as the auto calibrate count, if empty then the count is set to 100. This only applies to the keyword ImageScanFunctionsCalibrate

The Mode field is used to control which function to use:

mode = 0 color swopper, field 7 represents the red color, field 8 the green and field 10 blue. The name of the color in the field will be used instead of its normal colors eg if field 8 says "red" then the green element of the image will be changed to the red value. If clear is in one of the fields then that color element is removed from the image. This function affects the main image variable.

mode = 1 all the colors of the pixel are set to the average, makes the image black and white.

mode = 2 any pixel with any red in it is set to red.

mode = 3 any pixel with any green in it is set to green if operation is set to 1 then any color with no green is set to white.

mode = 4 any pixel with any blue in it is set to blue.

mode = 5 if red is higher than the average of all 3 colors plus the value in the order field then highlight the red.

mode = 6 if green is higher than the average of all 3 colors plus the value in the order field then highlight the green.

mode = 7 if blue is higher than the average of all 3 colors plus the value in the order field then highlight the blue.

mode = 11 sets a light dark split point. The number in field 11 is the split point this is a number between 1 and 255.

Pixels are split into there red green blue parts then averaged, if the average of the colors is higher than the split point then the colors are increased by the value in the order field. If the average is lower than the split point then the colors are decreased by the value in the order field. This is used to increase the image definition or contrast which means other ISF functions work better. The change affects the main image variable.

mode = 12 this function reduces the number of colors in the image. The order field controls the level of effect. The main image variable is altered. This function is useful to decrease the resolution before using another function helps remove scatter, another way of putting it is it alters the contrast! mode = 13 compares the image with the previous image and highlights the changes, sensitivity is controlled by order where 1 is very sensitive. Operation is the clumping threshold, this highlights clumps of pixel movement the red box is drawn round the pixels that pass the threashold. This function is being designed to highlight objects moving in the field of view where the camera is static. mode = 14 same as mode 13 but shows the speckle, this is pixels that have changed but not reached the threshold

mode = 18 this uses the gray scale to make decisions the input codes of cells 1607 1608 1609 are used to control which area of the gray scale to highlight with 0 meaning don't highlight and 1 meaning do. The grayscale is 255 divided by 27 with each bit representing 9 degrees of grayscale. The color of the pixel is changed from red to blue as the pixel gets brighter.

mode = 19 using the gray scale this function looks for edges of objects similar to scan vertical but done in a different way. The input code of cell 1610 controls the density, the first 4 bits control the vertical density and the last four control the horizontal density. The order field of the cell that activated the function controls the sensitivity.

mode = 20 using the RGB code in field 7 this highlights the pixels matching the rgb information

e.g. field 7 = rgb:170,0,0 this will highlight pixels that have red at 170 green at 0 and blue at 0 the number in the order field is added to increase the range so if order is at 7 then red pixels whose red is between 170 and 177 will be highlighted.

mode = 21 using the RGB code in field 7 this highlights the pixels matching the rgb information similar to mode 21 but this works on the average of the pixels colors and is set to highlight if any of the conditions are set, as apposed to all three needing to be set before highlighting.

mode = 22 using the rgb code in field 7 the red green and blue are increased by the amount in the order field. If the rgb codes are all the same then the overall brightness of the image is increased. The effect is written to the main image variable so repeat activations of the function will increase the brightness by degrees as set by the value of the order field.

mode = 23 same as mode 22 but in reverse so pixels colors are reduced by the amount in the order field.

Note a color may be highlighted but not necessarily with the actual color.

mode = 24 using the rgb value in field 7 this function takes the proportions of red green and blue and highlights pixels matching the proportions. This is an attempt to compensate for different brightness.

mode = 25 this function allows a number of functions to be used simultaneously. The numbers in field 11 are the functions to use. Other settings taken from the order, type and operation fields are used for each function.

A code is generated from the function. If the function has registered something in the image then this is compared to the previous 10000 codes generated, if a match is found then cell 1611 is activated. The input code of cell 1611 is changed to the number of times the code has been found in memory.

The output code is made up in the following way:

ISF code: isf5m32o2w12759

isf 320 5m 2w 12759
id mode order CBStop unique number generated using some math

id is the identifier telling anything using the code that its an isf code mode the number of the function that generated the code order the sensitivity or order value

CBStop this is a code generated by the outline process 1=no where to else to go 2=coordinates are the same as the start so the outline is complete. 3=width value has moved out of range. 4=height value is out of range. 5=outline draw process maxed out at 6000 6=outline process bypassed.

The code created is put in the SwopVar variable, use the tag <toprint=swopvar> to use the code. The input code is created from the red lines that appear in the image when objects have been detected highlighted and also pass the threshold value set by the Operation field. The yellow lines are created when an object has passed the intersection line that is set by the Type field. To use the input code use another cell with the tag <i=cell:n> where n is the cell number of the cell with ImageScanFunctions keyword. You can then use the information to do other things like controlling robots.

**ImageScanVertical** The process scans the image variable for vertical lines after it has been set up by ImageLoadScan. This is used by the robot to locate objects that have sharp edges. What information is displayed and returned is controlled and held in cell 936.

**ImageScanVerticalCalibrate** calibrates the image.

ImportFromCSV WARNING this will replace the entries in the BD. Entries

2000 upward are replaced.

**InputFilter** activates the input filter function.

**IPAutoRXOff** turns off auto IP if settings is opened then the original value is set.

**IPAutoRXOn** turns on auto RX IP. If settings is opened then the original value is set.

**IPRXAuto** closes down UDP and restarts it setting auto IP RX in settings, finds local IP number.

**IPRXFromFile** reads file field and uses the found IP number for sending to then restarts UDP routine. If there is a word in field 10 then the HtmlExtractor routine is used. The IP number is assumed to be after the keyword until < is found. This tells the routine that a html tag has been found. HtmlExtractor is used when getting the IP number from an online source i use http://checkip.dyndns.com or http://checkip.dyndns.com:8245/

these create a web page that can be used when downloaded using download file. **IPRXToFile** writes IP number of local machine to file in file field.

**IPTXFromFile** same as IPRXFromFile but changes the TX IP number.

**KnownWords** extracts words from the file in file field and creates a results file with the same name and extension except with an x on the end of the filename extension. The words are not in the same order as the words from the original file. This routine is useful if still getting gobbledygook words after the ExtractWords process.

**KnownWordsTwo** extracts the known words from the data created by ExtractWords and writes the results to the file: KWNFile.txt in the main Ai folder. The first 200 extracted words are sent to the output. To exclude words that are known but not wanted in the output, set the words Mode field to 1. Only words in the highest defrag area of the BD are used.

**LastCellNumber** sets the variable NNELastEntry to the number of the last entry in the BD. This is normally done automatically but could be useful if adding data using cell activations.

**LookInLog** look in today's log for the word/s, 40 must be in field 11 and a pattern must be used to trigger the search.

**LowPower** turns of WSB and SE processing so saving on system resources. **MapCreate** clears existing data and rebuilds the map database UNDER DEVELOPMENT.

**MapPlace** field 7 is the direction the robot is facing. This is a number between 1 and 60 like a clock. This works out what the robot can see from information from the map. Map data is held in the LocationFile.dat UNDER DEVELOPMENT.

**MapShow** shows the map in the image window UNDER DEVELOPMENT.

**MessageBox** field 8 are the words in the message box, field 11 is the caption.

**MessageBoxYesNo** field 8 are the words in the message box, field 11 is the caption UNDER DEVELOPMENT.

**NameFound** activates the name process UNDER DEVELOPMENT.

**NewSum** resets maths variable to 0 same as the cancel button on a calculator see also ShowSum.

**NextSong** moves to the next track on the play list.

NN=WSBR UNDER DEVELOPMENT.

**NNA1In** reads the USB analogue 1 input, converts it to a NN code (9 bit binary) and sets the NN input code.

**NNA2In** reads the USB analogue 1 input, converts it to a NN code (9 bit binary) and sets the NN input code.

**NNActivateCell** activates the Brain cell number in field 11.

**NNChangeCellInput** change the input code of the cell in file field to the input code of the cell in field 10.

**NNChangetoCodeIn7** changes the input code to the code in field 7.

**NNEEventOff** turns off Brain cell 1601 activation on Brain event.

**NNEEventOn** turns on activation of cell 1601 when a Brain event change is noticed to a cells input.

The file NNECCNewFile.txt is created in the program folder. This file records the nne entries that have changed.

To stop a cell's change being recognized as an event use the tag <eventno> in the cell, to exclude it from the event scan.

If the Tag <eventactivate> is in a cell that records an event then the cell is activated.

System cells 1000 to 2000 are scanned for changes.

**NNEInToBase10** converts the Input code of the cell number in Field 11 to a base 10 number. Sets the ToPrint variable with the output.

**NNEOutToBase10** converts the Output code of the cell number in Field 11 to a base 10 number. Sets the ToPrint variable with the output.

**NNEOnLoopOff** turns off activation of cell 1507 on timer loop.

**NNEOnLoopOn** turns on cell 1507 activation on timer loop.

**NNEOnTimerOff** turns off activation of cell 1505 on timer.

**NNEOnTimerOn** turns on cell 1505 activation on timer.

**NNEOnUDPOff** turns off activation of cell 1506 on new udp or window message.

NNEOnUDPOn turns on cell 1506 activation on new udp or window message.

**NNEScanImage** Brain cell 921 holds information for this keyword look at

system cells cell 921 for more information.

**NNESetInputs** sets the Brain inputs, changes the input code to that of field 7 does entries from number in file field to number in field 10.

**NNTrain** trains the NN using the training file in the file field.

**NNTrainInputF7** this uses the input code and the contents of field 7 for the training data which is added to the cells training data. The code in field 7 is the output code.

If the Order field = 1 then the cells neural network is trained after the data has been written to the training data file.

**NNWeightsSave** saves the NN weights to filename in the file field.

NNWeightsLoad loads the weights into NN filename in file field.

**NNWorkMatrix** activates the matrix file in the file field.

**OutputToClipboardOff** turns off the output to the clipboard.

**OutputToClipboardOn** turns on sending output to clipboard.

**PauseSong** flip flop function, pauses the song that's playing or starts the song playing if its been paused.

**PlayPlaylist** the file field holds file name of the file that holds the file names that are played by the mp3 sound routine. The Play list number in settings is the position in the file to start reading the names from the play list will play constantly till the end of the list unless stopped. Play lists can be created using indexfiles. The default play list is called TempPlaylist.txt and can be found in the Data folder.

**PlaySong** plays the song in the file field full path recognized types .mp3, .wav, .kar, .mid.

**PlayWAVSound** plays the WAV file in the file field, the wav file must be in the 'sounds' folder, mostly used for system sounds.

**PutInInput** puts the contents of field 8 into the Ai input, Operation must be 0 **Put7InInput** this is a field 11 keyword it puts the contents of field 7 into the input. UNDER DEVELOPMENT.

**Put8InInput** this is a field 11 keyword it puts the contents of field 8 into the input. UNDER DEVELOPMENT.

**Put9InInput** puts the contents of the file field into the input.

**R2D2Off** turns off r2d2 vocalization.

**R2D2On** turns on r2d2 vocalization makes a noise a bit like r2d2 from star wars.

ReadControlTF UNDER DEVELOPMENT.

**RestartUDP** restarts UDP

**ScanDownload** file field is file name of file to search for word/s in field 8.

**ScanImageFull** image scanning and analysis routine. An attempt to get more information from an image, work out what the image is and what objects in the

image are.

**SCFromText** reads a text file containing correctly structure sentences, these sentences are converted into sentence codes (SC) and if not already present added to the BD. If a word in the sentence is not present or the type value for the word isn't set then the sentence is rejected. This is a fast way of adding SC's which are used by the Output Filter function 4.

**Scroll** sends the contents of ToPrint variable to the scroll output screen. This is normally used from a cell activation using the keyword <keyword-ifn> where n is the function eg: <keyword-ifa> note repeat outputs to the scroll process are ignored.

**ScrolOutput** turns on the scrole output to the USB K8055 card.

**SendMessageToWindow** sends field 8 to the window name in the file field. To get the Ai to receive messages it need extra code on the end eg.pw:1234 o-i pw:=password and the end of message marker, 1234=Ai password o-i=where to put the message

If the tag <toprint> is in field 8 then the contents of the ToPrint variable is used if the tag <swopvar> is in field 8 then the contents of the SwopVar variable is used.

If using the character AiChar.exe program add pw: to the end of the message in field 8.

example to send a message to AiChar.exe

field	contents
file	AiChar

keyword sendmessagetowindow field 8 this is a message pw:

Cell 918 is the default settings used when send message to window is ticked in settings

If for some totally inexplicable reason this keyword fails to work try MessageToWindow this peculiar bug was spotted on an XP machine running a robot.

**SendOrderToWindow** sends an order to the window name field 10 is the window to send to field 8 is the order at the moment the only order is close which closes the window.

**SendToBrowser** sends commands to the browser program, sends contents of field 10 to URL bar.

<refresh> sends the instruction to refresh the page. some mode switches can also

be used used.

If mode = 0 just send field 10 to browser.

If mode = 1 send field 10 add the output.

If mode = 2 send field 10 add the input.

**SendToIP** changes the send to IP number in settings to the number in field 10 and then resets UDP.

**SendToTray** send the Ai program to the system tray.

**SetFocus** sets the Ai as focus and brings the Ai window to top using the script SetFocus.aut.

**SetFontsColours** see cell 816 for details.

**SetRQWeights** used to set the RQ's. The numbers in field 11 sets the RQ's in order, look in <u>settings</u> for more.

**ShowBadWords** bad words from input are sent to the output.

**ShowSongName** outputs the name of the song currently playing.

**ShowSongNameFull** outputs the full path filename of the song currently playing.

**ShowStatsOff** turns of statistics display in Image window.

**ShowStatsOn** turns on statistics display in Image window.

**ShowSubject** sends the subject description to the output.

**ShowSum** show the maths result cell 818 field 8 is put in front of the answer.

**ShowVersion** sends the ai version to the output.

**SixPClean** resets in cell fields 5 and 6 to 0. These are the fields that 6p learning sets up.

**SixPLearnOff** turns off 6P learning each timer loop.

**SixPLearnOn** turns on 6P learning each timer loop, this means learning can be done when the Ai is idle and means inputs are processed at the normal speed.

**SoundToOff** turns off sound to light.

**SoundToOn** turn on sound to light, uses the analogue input from the k8055 card then sends signals to DMX channels 1 to 3, this causes the Ai to run at about 50% processing power. inputting something into the Ai stops the process.

**SRFOff** turns off SR filter.

**SRFOn** turns on SR filter.

**StartTimer** starts the main timer.

**StopDownload** stops file download.

**StopSong** stops a song started with the PlaySong or PlayPlaylist keyword.

**StopSongAtNext** stops at the end of the track playing.

**StopTimer** stops main program timer.

**STTags** sets the ToPrint variable to the results of the STTags filter which translates the contents of field 11. The ST codes need to be set. If the tag is in the

code then the tag is replaced with the associated word.

**STTags-Activat**e same as STTags but the cell is then activated.

**SuffixCopy** this creates a file called SFCFile.txt this file holds the words that have a ST code in there emotion field. If mode = 1 then all found codes are added to the file. If mode = 0 then only words with codes added manually are added.

**SuffixFind** looks for suffix on the end of words then if found adds the associated ST code to the emotion code.

**System** file field is the program to run, field 10 is the extra command to send to the program being run.

**TimeInputCode** this converts time elements from the computers clock into the cells input code. Mode field controls which element to use.

Mode = 0 convert hours to input code.

Mode = 1 convert minutes to input code.

Mode = 2 convert seconds into input code.

**ToDoReminder** UNDER DEVELOPMENT.

**UDPF8F9Send** sends via UDP field 8 using the code in the file field as opposed to using entry 809 BD which is the default UDP entry.

This is also used when sending remote BC activation, eg. field 8 = in:000001111 ac:2376 f9=pw:1234do:o-o

where 1234 is the password of the remote ai and o-o tells the receiving Ai what to do with the information. The IP number and en: is added to the end of he packet by the program . The receiving Ai also looks for the keywords ac: and in:, ac:=activate cell, in:=input code. If there is no input code then the cells input code is used.

**UDPF8F9SendForce** if f8 is empty then the output is used

**UDPRestart** restarts the UDP connection.

**UpdateSettings** reloads the settings file.

**UpDateBrain** updates the Brain from a text file. The file in the file field is used, if empty then the file used is NewBrainData.txt in the data folder. The button New Data adds the cells data to the NewBrainData.txt this is used for moving data from one Ai to another.

USBOff turns off USB K8055 card.

**USBOn** turns on USB.

**USBStopDInF10ms** turns off digital after F10 milliseconds.

**UWSave** this saves cells with user window tags to a text file which can then be used by the UpdateBrain keyword. The file name in the file field is created, if empty then the tag details will be added to the default new data file.

**WipeEmotion** clears the emotion field of cells in the last part of the BD.

wsbShow show WSB window.

## **General Information**

The main Ai window consists of 3 text boxes and 3 buttons. Starting from the top the first text box shows various information that scrolls up as inputs are processed. The middle box is a edit box and shows the output. The bottom box is an edit box and is the input box, words typed in here will be grabbed by the Ai processed then the output sent to the middle output box

#### Autoexec

When the program starts the Ai reads cell 814. It is usually called autoexec. Field 8 is put into the input. The Mode field controls how subsequent inputs are translated this should bet set at 0 unless you want all inputs translated in a different mode. Mode switches can still occur during translation.

#### Search

The search routine is designed to do different jobs depending and the algorithm that is using it.

**RAS** uses the routine most so it is advisable to have the number of files searched limited to an acceptable amount depending on how fast the response is required. The file in cell 819 field 11 is the file name of the file that holds the file names of the files to search each time by the **RAS** routine.

If you want to have a large number of files searched and don't want the **RAS** routine to search all files each translation then it is best to use a different search file. If you use the keyword **searchinfofile** for a pattern then the file field is the file holding the file names that will searched. This allows for different search criteria depending on what has been activated from the translation. You could have a search file full of say html file names that is used when the pattern for say 'search html for fish' is entered into the input.

# **Effect on system performance**

Some of the routines in the program can put a heavy strain on the computer system resources. The effect on resources is controlled by how you have it set to process eg. if you have hundreds of files used as a resource for the RAS routine then hadrdrive access speed will have a major impact on the amount of time to process. If the RQ weight of any routine is set to 0 then that process is ignored during processing of inputs. Here is a brief summary of the routines and the effect on system resources only routines that have any noticeable effect are included. As the Brain Database gets bigger it will take longer to search but no

effect has been noticed on a Brain with over 7000 entries

**QNV** this uses the TF but also does allot of deep scanning and if set to its highest scan level can start to use allot of CPU

**RAS** there is no limit on the number of files that can be used to search by the RAS routine these files are held on the hard drive hard drive speed effects performance use fewer files to process inputs faster.

**P**, Pattern this routine uses some of the same search functions as RAS **WSB** when on this puts a heavy load on the CPU the WSB routine works as a separate process and attempts to run as fast as possible, the faster the better as this increases the sample rate, the routine uses the analogue input channel from the USB card, the better the sample rate the better the word association. **SE** search engine this sends off a search request and slows down the internet connection it has no effect on the Ai because it runs as a separate thread **NNE** this routine loads weights from text files the bigger and more complex the matrix of cells the the bigger effect on hard drive and CPU

**NNE6P or 6P** this digs deep into the NNE data base following words connected to other words looking for connections the more words and the more connections the longer it takes. If 6P learning is ticked the learning routine compares every word in the NNE with every description of words in the TF this means millions of requests are made to the TF it takes about 16 seconds to process 1.4 million requests NNE6P words start at entry 2000 in the NNE database. The whole database is loaded into memory so speed of memory and CPU are affected. The keyword **lowpower** turns off the **WSB** and **SE** routines

#### **DMX** control

The dll K8062D.dll is used to operate the USB to DMX controller. The Velleman K8062 is available from Maplins.

NNE cells 1511 to 1765 correspond to DMX output channels 1 to 254

## **Function Check**

The following are ways of checking each algorithm to see if they are set up correctly and working properly

# QI Question and Is

input: what is a dog

answer from QI: Dog is noun four legged flesh eating animal akin to the fox and wolf and of many breeds

# **QNV Question Noun Verb**

input: how do fish swim

answer from QNV: I think that fish is animal swim with fins with living with

water.

## PR Pattern Recognition and SC Sentence Code

Tick SC learning in settings, set STM Short Term Memory to 4 then input: nothing happened

put the phrase in around 4 times until the STM adds the pattern to the end of the BD Brain Database.

When added to the BD the Pattern is a Sentence Code and sends an output if there is an exact match to change the SC to a PC change the cells type from 32 to 60

put into field 11: did <%> happen now?

press UPDATE

now input: nothing happened yet

output should be: Did nothing happen now?.

## **DT Deeper Thought**

Type something into the input box, when processing has finished change the answer in the output box then press OK or ENTER. The routine will create new training data, re-train the relative cells neural networks and create new connections between cells.

Typing the same sentence in should produce the re-trained response.

# **RQW** Reply Quality Weight

The value of the ReplyQualityWeight is generated by how well the Ai thinks its understanding the inputs, the value increases if the outputs have on average a good reply quality and in reverse decreases if the Ai produces bad outputs. The color of the word TIMER in the top left hand corner is changed according to the ReplyQualityWeight value

# **De fragmenting**

This process organizes the data into groups so each process only needs to access data that it uses. This process runs at startup if new data has been added. This speeds up searches. Training files updated and node connections are reassigned to the new cell entries. Tag references to cells are also updated. There are some situations where tags cell calls become unstable a fix for this is to tick disable defrag in settings. New data will be added to the end of the database which is also searched by all processes as well as searching there own relevant part in the database. After a few thousand new entries you may start to notice processing speed slow down more than normally would happen if the defrag function is enabled.

# **Process order**

this list is the order of processes starting from main timer activation

l-		
timer 1 activates		
marker put scrole output ****		
Act On d inputs		
read analog inputs		
timer 2		
stuff on timer		
	update brain with number in outs	ĺ
	check entries 1000 to 2000 for changes	
	Play list Kick start	
	check digital counters and activate cells	
	accordingly	
	check Read File To Input	
	timer loop	
	unier 100p	
		S
		d
		[p]
		sł
		de
		[u]
		cl
		cl
		a
	read message	
	do udp	
	activate cells that do so on timer	
grab text from input box		Ī
input filter		
update date time		
some learning		
visual stuff		
do udp		
	process input most controlled by rain	╬
	process input most controlled by rq in	
	settings Dissort input and cotup wariables	
	Dissect input and setup variables	
II .	II	11

Sentence type emotion 3 learning processes special sentence code function look for patterns translate using brain QNV RAS search engine expected in input both types short term memory more pattern recognition subject and subject new Parameters wheatstone bridge neural network neural network backward neural network deeper thought neural network cell numbers at one neural network dig in deeper auto learning face check output choose answer output filter say print send the output to various places check for brain changes do udp add to short term memory gap in time r2 d2 vocalization reset variables and restart main timer

# Other programs used by the Ai Ai Server

AiServer.exe is a tcp/ip server that works as a standard web server. Its a very simple server and only serves pages with the extension .html it wont serve images or any other type of file. Its designed to send information to the Ai via a web page form. The server binds to the ip address if 0.0.0.0 then it binds to any incoming ip address. The port number is normally set to port 80 the usual for a web server. Incoming packets are sent to the window name usually set to the Ai's name this then processes the string and sends it back to the server which then sends it to the client.

Only files in the server programs folder with the extension .html are served. The server first sends the page index.html then the main.html

To access the server enter the machine the server is running on ip number or host name DONT add the file name this will override the Ai output and you wont get the Ai response even though you are served the page eg. on my machine start the server

open web browser

type in the address http://zip

zip is the name of my machine change the name to the name of your computer you should see the index page the address bar should read http://zip/press the button

the page will Wait until the message has been sent to the Ai processed then sent back. If there is no response after 30 seconds the server will timeout and the index page will be sent. The timeout is adjusted once communication is established.

If aicell is in the input then the words "cell activated" is sent immediately in reply the server sends a the reply just after sending the message to the ai the reply from the ai is ignored. This is used so form button can send a message quickly but also it stops the need to create an answer from a cell activation. The code for the web page is: aicell1234pw where 1234 is the cell to activate. This code was created for the robot remote control.

To see DeBug outputs add /debug to the web address this will cause the server to send more information to a debug program running on the same machine as the server.

If the window message from the ai to the server has the words send: in it then the words after is the web page file name to send eg. send:results.html will cause the server to send the results page rather than the usual main.html web page.

aicode123 is a way of sending a code to the ai keeping the code in the message. Its used for buttons e.g.

<input type="submit" name="aicode001" value=" TEST CODE ">

this will send the contents of the edit box followed by the code in the same way as SEARCH FOR does. Normally a

button press will only send the contents of the edit box. If the edit box is empty then the button name and label

are sent instead

Tags are used in the served page to insert information

<aireply> inserts the reply from the Ai before serving the page. The tag must be on a line of its own to stop weard things being added, the whole line containing <aireply> is replaced with the output from the Ai.

<file=filename> This tag inserts the contents of a file into the page. Filename is the full path location of the file to insert into the page.

### Ai K8055 Interface

AiK8055InterfaceV002.exe is used to read inputs from the K8055 card and transmit the data to cells in the Ai. Card address is a number 1 to 4 and is the card number to connect to. 4 cards can be connected to a PC a single card can be connected to by many instances of the program.

Timer length is the time in millisecond between card reads.

TX switches on transmission to the Ai

To cell values are the cell numbers in the Ai to set input codes are updated.

Activate if ticked the cell is activated after the input code has been set.

On change the data is only transmitted if there has been a change in the value since the last read.

Window name is the name of the window to transmit data to usually the name of the Ai running on the same machine.

CARD when pressed the program attempts to connect to the card numbered in the Card address box.

START starts the program timer sampling data at a rate according to the value in Timer length

Auto start if ticked then the card is opened and the timer started automatically at program start up

Notifications at the bottom of the screen give information. Count is the number of times the timer has triggered it resets at 1000000

Click is the data being sent Click 5 is D3 4 5 to cell etc.

When the program closes the data is saved in a text file called AiK8055.txt this is read when the program starts and the values are loaded into the fields.

The rate in which the Ai can receive information is dependant on the speed of

the Ai timer as well as the Interface timer speed

## **Download file**

dlf.exe is used to download a file from a server.

# Ai Encript

AiEncript02.exe this program encript's text files. Make sure the decript is correct before deleting the original as some files can corrupt the encript process. The Encript process can be activated if it receives a Window message

## Ai Capture

AiCapture03.exe This program is used to capture images from a web cam. When the program receives a window message then it saves a snapshot image called 00001.bmp in the image folder this can then be used by the Ai usually using the keyword ImageLoadScan then ImageScanFunctions to process the image. The keyword CopyFileNewName can be used to copy and rename the captured image. Useful if you want to keep all the images. Each new image has a number that is incremented for its filename.

# Processing Algorithms, routines and functions. These are listed in the order they are called in the program with other abbreviations at the end.

## **CON Connections**

This function uses the words connected to the input words to choose the best sentence from a number of sentences. Words in the BD are connected to each other using the In Cell fields. 9 words can be connected to a word, the number in the In Cell field is the cell number of the connected word. When an input is processed these connections are used to set the Connections variable. This variable is then used when scanning multiple sentences. The number of times a connected word is found in a sentence is used to determine the best sentence, where the sentence with the highest count is used. It is hoped that this will extract the best sentence for the output.

The Mode field and field 7 of cell 911 control which words are used.

Mode = 0 words connected to the input words and words whose SC code is in field 7 e.g if field 7 has the SC codes B% then bad words and nouns in the input are also used.

Mode = 1 words whose SC code is in field 7.

If T is in field 7 then this tells the process to add the ST subject and object words, these are the contents of field 7 and 8 of cell 1606.

Processes that use CON are:

SEN The SEN process uses the CON output if the Order field of cell 910 is set to 0

KEY The keyword FileConnections scans files for connections to the input.

#### **SC Sentence Code**

The code is created from the words type value. If the word isn't in the Brain Database then the code associated with the word is set to B.

# **ST Sentence Type**

When the input is processed the Sentence Type ST routine creates codes referred to as ST codes. These codes represent meaning extracted from the input. a ST code is two letters and colon eg fq:

words can have codes associated to them to do this put the codes you want associated with the word in the Emotion field. Eg. the emotion field of the word the could look like this: <f:38><ms:> . The tag <ms:> is used by the ST process,

its meaning is detailed below. The tag<f:38> is used by the emotion process the number after f is the number of times the word has been found in the input. If SC and ST learning is ticked in settings then the code is added to the BD if not already there. when codes are added they are given the type number 137

The codes found and created are then searched for in the BD. Cells whose type is set between 135 to 138 are ST codes.

The Type field number controls how the code is matched when searched for in the BD

Type = 135 ST code if the codes are the same but not necessarily in the same order then use the cell

Type = 136 ST Code if the code is in the cell word then use the cell NEEDS CHECKING NOT SURE

Type = 137 ST Code if the codes are the same as the cell word and in the same order then use the cell

Type = 138 ST Code if an exact match of the cell word is found in the ST codes then use the cell

When a match is found in the BD then field 11 is checked and processed. Any ST tags found in field 11 will be converted to there associated word. Eg. if the input is: what is fish and field 11 is: are you talking about a <su:> then the output will be: are you talking about a fish. After processing field 11 then the cell is activated. This means you can set the ToPrint variable and process it using the normal Tags in the Tags field. This could be used to get the details of a word eg. if the inputted word is fish and there is a cell set to su: then field 11 could be: <su:>

then the tags field could be: <lookup-cell><clear><f12-cell:0> the output would be the detail field of the cell associated to fish.

In the tag <f12-cell:0> when the cell is set to 0 then the cell number created by the <lookup-cell> is used.

The first output created is used for the ST process output.

ST codes can be created automatically using the keyword SuffixFind or manually added to the emotion field.

ST codes are defined so as to fit in with the experimental formula of language. ST codes are in lower case when in the BD.

\*\*\*\* ACTION perform some task

AI: instruction: command or keyword

AF: fact : update brain

AP: preaching or dictating : defensive mode no religion

AA: answer: update brain

AQ: question: reply with an answer

\*\*\*\* CONNECTED TO THE AI LAST OUTPUT subject is set or object

ongoing chat could be connected to other things

CR: reply : find something to say CE: repeat : rephrase the last output

CS: same as reply but said differently, agreeing with the ai's output or yes

CP: rephrase, not quite sure about the last output or no. Rephrase output to double check

\*\*\*\* DISCRIPTOR a word that describes something usually the subject

DI: descriptor describes what the subject is

DD: describes what the subject does or is doing words with ing on the end

DL: location of thing

DP: place in something else

DS: size of the thing

DC: condition of the thing

\*\*\*\* EXTRA means extra switch used by functions for various things.

EX: <ex:1> used by image scan functions to switch variable CPDevide to be sensitive to top part of image

\*\*\*\* F cant use F due to interference with Emotion code

\*\*\*\* GENERAL CHAT reply with interesting and relevant

GA: anecdote or story: try and find similar references relevant to ai and the story

GR: rambling conversation: tell a joke or story

GJ: joke: laugh

GQ: quote: maybe update brain, or ask a question about the quote

\*\*\*\* LEARNING update data base with new information

LI: information : update brain

LO: observation : question the observation maybe update brain

LS: suggestion : question the maybe update brain

\*\*\*\* MODIFY modifies things like subject

MS: modify subject change an RO: to RF:

MO: modify object change RO: to RF:

MA: modify all or any

\*\*\*\* REFERANCE

RY: you referring to the Ai RM: me referring the in putter

RO: referring to the subject or object

RF: referring to the subjects or objects function

RT: referring to the task action or process

RP: referring to more than one plural

\*\*\*\* SUBJECT

SU: subject first noun found SO: object second noun found

\*\*\*\* TENCE

TP: past words with ed on the end so words like confused get through these are a state of doing

TN: now, present

TF: future

\*\*\*\* OTHER special function tags

OW: hidden tag that gets set by words that don't have a ST code words are added to the end of the created output only one word is used.

\*\*\*\* USER DEFINED Sentence Codes with a U are user defined there are 9 available. These are useful for singling out a specific word or group of words.

**UA**: User Defined

**UB:** User Defined

**UC:** User Defined

**UD:** User Defined

UE: User Defined

UF: User Defined

UG: User Defined

UH: User Defined UI: User Defined

\*\*\*\* V cant use V due to interference with Emotion code

1606 this holds information on the conversation

filed 7 = subject

field 8 = object

field 10 = sentence type codes

in cell fields hold the subject and object nouns cell 1 is the subject cell 2 the object as new words are added the list moves down until dropping of the list

# QI Question and Is

If there is a question, word type 34 and the word is, word type 47 in the input then the following word is looked up and the detail field is sent to the output

## **LM Learning Mode**

Creates associations between cells using the prompts in system cell 923

#### **BD** Brain Database

The Brain Database is the database used by the Ai for all things.

## **QNV Question Noun Verb**

A question about a noun and its associated verb cause the QNV routine to scan through the various words detail fields to create associations. The resulting words are built into a sentence using the information from cell 909 the output filter for QNV field 8=first bit "i think that" file field=second bit "are" field 10=third bit "with" field 11=fourth part "do"

## **EI Expected in Input**

The routine looks for words in the input that it is waiting for an answer to, set from a previous input. Field 8 the file field hold the two words. Depending on which words are found then the response is in field 10 and field 11. 10 is the response if word in 8 is found and 11 is the response if the word in file field is found. If the type field is set to 68 then the response is put into the input, if type field is 69 then the response is sent to the output. The operation field in entry 811 is the number of times to keep asking for the word before giving up, field 11 is the "I give up phrase".

# K Keyword

Keywords are predefined words in the keyword field that perform various functions. Keywords are described in more detail here

The Keywords exitai and restartai are pre defined within the program, all other keywords are activated by any user defined cell word. Some Keywords can only be activated with a pattern.

# PR Pattern Recognition, PC Pattern code and SC Sentence Code

The Sentence Code SC is created by using the word type number of the input words. This number corresponds to a character in the ASCII character set. The characters are added together to build the Sentence Code.

If SC learning is ticked in settings then sentence codes created from the input are added to the Brain db.

The type field for a sentence code is 32, the cell is only used if an exact match is found with the inputs sentence code. To turn the SC into a Pattern Code PC the cell type field needs to be changed to 60. Patterns cells are used when the Pattern

is found anywhere in the inputs sentence code

Tags in field 11 are used to build an output

orsendtobrowser>

cprsendtosearchengine>

**rsetname**>sets the name variable and if SC learning is ticked in settings then adds the name to the Brain if its not already there. If the name is larger than one word then a dash - is put in between the names. If there is a number in field 10 then that number is used when adding to the Brain type field instead of the usual number 51 which is the type denoting a name. The names with spaces are added to the detail field . The name is that of the person talking to the Ai.

reads out the name variable puts the contents of field 8 in front
of the name. If the name is in the brain then the contents of the detail field of the
cell holding the name are added to the output

ordafter>

<activate> activates the cells neural network, sometimes other cells are activated instead of the pattern cell depending on other tags used

<use7> adds contents of field 7 to the output variable

<use8> adds contents of field 8 to the output variable

<use10> adds contents of field 10 to the output variable

<{any word type}> any pattern code eg pattern code for a noun is % this is created by f4 of the fish entry in the tf being 2 so if you use the tag <%> then the noun in the input is used. Because the sentence code is scanned words that aren't in the pattern but are still in the sentence code can be used for the output. F11 is used for output with words and tags. eg what is<B>, would use the bad word in the input for the output instead of the tag <B>

f9 is used when there is nothing in f10 this is legacy code and due to change.

#### **EM Emotion**

The Emotion field in settings holds tags. These are codes within square brackets eg. <f:12><v:3><tp:>. Processes that use these tags are the ST and EM processes. For ST see above.

EM uses the f: and v: tags. The v: tag holds the Vibe Number (VN) associated to the word, pattern, ST code or RAS code. The number is used by the processes to tell the Ai if the cell has a positive or negative vibe where 1 is very negative, 9 very positive and 5 is ambivalent. The numbers are used to set the Vibe Count (VC). The VN's are added together then divided by the number of v: tags found. The result is converted to binary which is used for the input code of cell 1614. The Emotion function then activates cell 1614. The number with the f: tag is the number of times the word has been found in an input.

Cells used by the emotion processes

cell 935 to cell fields hold the most common words see System cells 935 for more details.

cell 860 field 11 is words moved from cell 935

cell 1614 the input code is the binary representation of the VC.

## NN Neural Network, NNE Neural Network Extended, BD Brain Database

The Brain Database is two databases merged together. These databases were the Translation File TF and the Neural Network Extended NNE databases. These were used on previous versions of the Ai program. all references to the TF or NNE now refer to the Brain Database BD.

Each entry in the Brain Database is called a cell each cell can be connected to other cells and grouped together in a matrix of cells. The processing of cells is done using a matrix file, this file holds the numbers of the cells to be activated, each line holds a number and the cells are processed until it reaches the end of the file. Cells can also be activated using a keyword, a Brain event or directly from the database window.

The file names in entry 919 fields 8, file field and field 10 are the file names of the matrix files used. Field 8 is the first to be read, then file field and then field 10. If there is nothing in the field then nothing happens. The file field is usually ...\weights\6PMatrix.txt this file is created by the NNE6P routine. When the Matrix file is created it uses the numbers in the files in the fields 8, file and 10 field 11 is put into the box marked **cell action** this means the cells can have a default output set when the words are added

Brain cells 800 to 2000 are used for various things by the program look in <a href="System Cells">System Cells</a> for more details

When a cell is activated the cells Neural Network is fed the input code. This is processed through the network and produces the output code. The network is trained using the training data which controls the output produced by the input. The network will always produce an output even if it hasn't been trained with that particular input code.

The input is a 9 bit code consisting of 0 and 1 the output is also a 9 bit code of 0 and 1.

Weights are created when the network is trained, these are stored within the Brain Database in a hidden field in the cell data

The field Cell Action performs actions before Tags are interpreted **cell IN to cell OUT** = send input code straight to output code, bypasses processing by the NN

**cell OUT to cell IN** = feeds the output back into the input

The box's on the right are used to connect the output nodes of the NN to other input nodes in other cells. The input code in the cell is altered and changed to the output node eg. node 6 to node '4' in cell '2' if the output of the cell is 000001000 then counting from the left we can see node 6 is 1 so then if the input code of cell 2 was 010000000 it will now be 010001000 non of the other nodes are changed. If an input code is created by other cells then its important to have the final cell at the end of the list of cells in the matrix file which processes the cells in sequence.

**O to TRAIN** this button puts the input code and the output code into the training file, the weights filename with .nnt extension

The Tags field is used for function tags which do various things <u>click here for details of Brain Tags</u>

## **RAS Reference Action Subject**

The RAS routine breaks down the input into 3 distinct types. This routine is designed to look at individual words in more depth. The routine utilizes the search routine first looking for the subject word then rescanning for the reference and action words. The assigning of words to different types is dependant on the types of words in the input ie the subject word may be the noun in the sentence and the reference may be the verb in the sentence but if there is no verb and there are 2 nouns then the reference word will be one of the nouns. This routine is particularly useful in finding information that could be in many files. The test, search file, contains over 60 different files from text files to web pages that are searched. The search takes less than a second on 1296mhz laptop. Because the RAS routine breaks down the sentence to 3 words only you would have thought that the meaning in the sentence would be lost but most times this isn't the case consider the question 'please tell me what the capital of england is'

RAS=england-tell-capital this quite clearly describes what is wanted, this also means that different ways of talking or asking for something can still produce the correct answer.

If a file that is searched contains 2 or more of the RAS words then the sentence containing the words is sent to the output.

The Brain Database is also searched for the 3 words joined eg. england-tell-capital if found in the database the sentence in field 8 is sent to the output. The file name in field 11 of entry 819 is the file name of the text file holding file names of files to search when using RAS

# SJ Subject

The Subject routine attempts to work out what subject the conversation is about.

The most frequently found word in the Short Term Memory STM is compared against key words that define subjects. Cells 840 to 900 are used by the Subject routine. Fields 8, 9 and 10 contain the keywords that the most occurring word in the input is compared with. The sentence in field 8 of the subject match is sent to the output. Words in field 7 are sometimes fed back into the input. This happens when there is no other output or the output is repeated.

Cell 40 is the 'no subject set' response.

## **SJN Subject New**

The new subject routine is designed to better extract meaning from the input. Using a number of words in the input to do this eg. do you like talking, would you like to talk to me, what do you like talking about. The key operators to distinguish between are:

do you like

would you like

what do you like

Subject new uses cells 860 to 900 with the more complicated meanings toward the end, so if there is say 'you like' then this meaning is used if not overwritten by another meaning in following cells if say 'do you like' is a match then this meaning is used. Cell 860 is used by the Emotion routine so the fields order,11 and 8 are sometimes changed. Cells 861 to 900 are user defined. Field 11 is used for the subject words. The cell with the best match is activated. The Subject Reply variable, RQ number 14 in settings is set with the output from cell activation. eg.

field	contents
order	3
field 8	if you <'> so
field 11	do what

using order set to 3 uses one of the Brain translation routines which also uses PR Tags. The word type code between brackets is replaced by the word of that type in the input. If a word of that type isn't found in the input then the tag is replaced with a space.

# **STM Short Term Memory**

The STM is some times refereed to as the Data Store. The length of the STM is set in the box called Learning file length in settings. The number is the number

of previous inputs to remember and refer to when looking for repeat occurrences of words when trying to work out what we are talking about. The word is first looked for using the Subject routine, if no match is found then the database is searched.

Brain Database cells 35 to 38 are the various default outputs if nothing is in field 11 of the word found most in STM. If SKIP is in 11 then the word is ignored, if NoSTM is in 11 then the previous word is used. 35=STM noun 36=STM verb 37=Word found in STM and TF not a noun or verb 38=Word in STM but not in database

## QN rq slider in settings

This is the setting for the depth that the QNV algorithm goes to find an answer, can be heavy on processing power

## **SM Sum Maths algorithms**

#### P and PAR Parameters

The PAR algorithm attempts to get as much information as it can from the inputs received. The first thing it does is scans the input for nouns the found nouns are compared with nouns found in previous inputs and applies weights to the nouns, the noun that appears the most is treated as the main subject noun. RAS keywords words with 104 in the type field are treated with highest priority. Verbs and Adjectives are then scanned and the verbs and adjectives occurring the most are given highest priority. The words in the priority words field 10 are words associated to the word the main word should also be in this list. The words in this list are the words used when search the information files. The file in the file field is a text file holding the filenames of the files to search for the keywords in field 10.

The first results file created is the results from the main subject word, the first file is searched using the adjective key words or the verb keywords creating the second file which is then researched with a third file being created holding more narrowed down answers.

If the third file is created then full reply quality is assigned RQ weight no.19 in settings, if only 2 results files then RQ = RQ weight / 2 if only one file created RQ = RQ weight / 3.

This routine is very good at pinning down information i use it mainly to find the prices from the price list as it also picks out numbers and you can associate many things to the words. The routine uses the short term memory and will remember the words from previous inputs until replaced by another word this

means it will remember what you are talking about.

Because the routine uses information generated by the RAS routine then RAS rq slider 11 in settings must be higher than 0

## **WSB Wheat Stone Bridge**

This processes uses the analogue input from the k8055 card and turns the signal into words. If there is no k8055 card connected then the random number generator is used instead.

In settings the WSB window, DX window and User window, need to be ticked and RQ slider ws needs to be higher than 0. If WSB learning is ticked then new words are added to the BD. The new word is associated to the next available English word in the BD. If in the output a word seems wrong then to change words use the = sign between the word and what it should become e.g. dog=cat will change the word dog to cat when an associated WSB code is found in the incoming signal.

The process starts just before the input is processed and ends a short time after processing. The process can be kept going but it uses a lot of processor power.

I use a circuit called a Wheat Stone Bridge this circuit has the two probes inserted into the soil of a plant pot holding a plant. The WSB circuit is what early lie detectors use. The signal from this circuit is connected to the analogue input of the K8055 card.

There are 3 pieces of information extracted from the signal these are:

**Resistance** (wsbresistance) The resistance value changes slowly so this value is used to associate to whole sentences, if the plant is moist i.e. just been watered this value is high so the sentence associated is 'I'm feeling damp'.

**Resistance variance (wsbvarience)** The resistance variance produces two letters the first is the number of times the signal has raised over time the second is the number of times the resistance signal has dropped over time.

**Frequency and frequency variance over time (wsbwave)** This uses a number of variables that have been set up from information extracted from the signal to create letters and word lengths. These created words are then checked for in the BD and the associated English word is used in the output.

Cell 908 controls how the function works. The input code controls how the data is presented. From left to right:

## red green blue purple resistance variance numbers keep\_going audio

eg. If the input code of cell 908 is: 010010001 this will show in the DX window the green wave signal, the resistance and make a noise.

The line color represents the following:

red = number of times the signal rises

green = wsbwave

blue = number of times the signal falls

purple = red and blue signals combined

resistance = wsbresistance

variance = wsbvariance

numbers = the raw data, the signal is digitally processed this produces 10

different values

keep\_going = keeps the process going

audio = makes a noise, the frequency is altered by the value of the signal

Mode field controls the output filters

mode = 0 wave output

mode = 1 resistance output

mode = 2 resistance variance

Order field controls the maximum length of the words. Normally this is controlled automatically but when using the random number generator word lengths can become long. The smaller the words the quicker the learning process. If set to 2 the random number generated codes will resolve to normal words after about five minutes of training.

Operation field, if 0 the analogue channel 2 is used, if 1 then channel one is used.

Type field controls the number of words in a sentence. The output length is limited to 160 characters.

type = 0 automatic mode, sentence length controlled from the resistance value.

type = 1 upward number of words in sentence.

# **SE Search Engine and SEN Search Engine New** UNDER DEVELOPMENT Cell 910 controls how this process works

If there is a file name in field 8 then the CON process is used.

If there is a web address in field 8 then the web based resource is used. The switch is http if this is in field 8 then it is assumed to be a web address. This is

normally a search engine.

This process sends a search request to a search engine, then uses the search engine results for the output. The process uses the DownLoadFile. The downloaded file called SENew.txt is then processed depending on the control words in field 11.

Field 11 is used for control words:

CheckWords only use words found in the DB in the output file.

ExcludeWords the words in field 7 are excluded from the output.

ExtractWords is a simple html parser, only works on simple web pages.

Words are extracted from the downloaded file and a new file called SENew.txtx is created.

System cell 910 holds the information for the process

Field 7 contains the words to exclude from the output file.

Order field is the line number to use for the output. If set to 0 then the

Connections (CON) output is used.

Field 8 is the first part of web address.

Field f10 is the code added after the search string.

The Mode field controls the search string

Mode = 0 then the search string is built up with the input sandwiched between the contents of field 8 and 10.

Mode = 1 then the search string placed in between fields 8 and 10 is built up in one of the 4 ways:

the noun in the input

the noun and the adjective

the noun and the adverb

the noun and the verb

Mode = 2 the RAS code is used the - in the RAS code is replaced with a + If there is no search word then the process produces no output.

# **DT Deeper Thought**

The Brain Cells of the input words are activated the output code is used to select cells to use for the answer. This means you can use the neural network and train it to use certain words/cells in the output. The input code is the sentence code character numbers added together then converted to the 9 bit binary code used for the input. The thinking behind this routine is to use the full functionality of the neural network and using all the input to work the neural network. This should also help facilitate a feedback look to build better responses by training the network.

If you click in the output box and the answer has been produced by the DT

routine then you can edit the output. When you have edited the output press ok. The entries of the original input words have there output nodes connected to cells changed to the new words from the edited output. New training codes are created and the training file for the cell is updated. The training file and weights file if different from, filename created by the cell number then new training and weights files are created using the format <cell number>.txt

The training file is then loaded into the neural network training algorithm and new weights are created and stored in the new weights file associated to the cell. The DT routines are designed to provide a feed back loop and utilizes the full functionality of the neural network.

The DT routine attempts to incorporate Chaos Theory. The training algorithm uses random numbers for the weights and then tests those weights though the neural network keeps the best outputs and starts the process again. This system is used to choose the best words and by editing the output and changing the words the connected to words are updated with better words and the best words are kept so the principle is the same as the generic algorithm that trains the neural network but its done by the person updating the output rather than randomly. If the same words are changed allot then the training data builds up leading to longer training sessions as more conditions need to be set. The number of generations is set in the settings dialog. The more generations the better chance of achieving 100% accuracy, if this is reached then the training algorithm is halted and the results written to the weights field used by the cell. The smaller the amount of training data there is then fewer number of conditions need to be set and thus training is quicker. There might be instances where all conditions of the training data cant be set or would require considerable more generations of training to reach 100%. Future development will probably have a sleep option so at times of inactivity the Ai uses the time to train the network.

#### Created file

the text file NNECCFile.txt is created by the changed cell inputs routine. As the Ai is processing inputs certain functions and algorithms are used these create information, this info is added to the Brain Database. err this needs looking at

#### **6P NNE 6 Points**

This routine uses the NNE database. Entries 2000 upwards are used by 6P. Words in the TF are added to the NNE, each new word is linked to the TF, the entry number of the word in the NNE is put into field 1 of the words entry in the TF. The words added are words in the input but they will only be added if the word exists in the TF. When 6P learning is ticked in settings then when an input is being processed the 6P words are scanned and associations are made by

adding NNE entry numbers to output node cell numbers 5 and 6. Field 12 TF is scanned for the word being examined and the word in field 6 is added if it exists in the NNE. The keywords **sixplearnon** and **sixplearnoff** turn on and of 6P learning each timer loop, this means learning can be done when the Ai is idle and means inputs are processed at the normal speed.

This routine works by making connections to words in the input following associations with other words. Each word has an entry in the NNE the output nodes cell number is used to associate to other words cell number nodes 1 to 6 are used. The word being examined in the input is checked for in the TF if there then it looks to see if it has an entry in the NNE if field 1 is higher than 10 then this tells it to go to the number in the NNE. It then looks at the numbers in the output nodes cell numbers starting at node 1 and follows the entry numbers checking for connections to the other words of the input, if a direct connection is found then the words that took it to the word are used for the output. If no direct route is found then it chooses one of the other answer routes, a bit of filtering is done using grammar and crude associations.

The last 3 bits (righter most) of the input code can be used to force a route \*\*\*\*\*\*000 = work your way through all nodes

```
******100 = follow node 1

******010 = follow node 2

******110 = follow node 3

******001 = follow node 4

******101 = follow node 5

******011 = follow node 6
```

#### EB and NNEB Neural Network Extended Backward

This routine uses the keyword from the PAR routine then using the NNE looks for words that connect to the word. Its works in reverse when compared to the NNE6P routine.

#### EC and NNEC Neural Network Extended Cell numbers at 1

This routine uses the **Word Find** routine to find a keyword from the input, the WordFind routine uses different criteria than the PAR routine so the word is often different. The Cell in the NNE associated with the word is activated. The output nodes that are set to 1 after activation, if connected to another cell then the TF entry word associated with that cell is added to the output. The input code is the input code generated from the user input.

At the moment this routine only uses one of the words from the input but this routine is proving very good at producing accurate answers and because it uses

the full functionality of the NNE it is extremely flexible and also opens up more self learning possibilities.

## **DID Dig In Deeper**

This routine is similar to the 6p routine but follows cell connections to much greater depth. The routine attempts to find connections between inputted words by following the cells that the inputted words are connected to. The 9 Cell node fields contain the cell numbers to follow these numbers are either added manually or automatically by the LM, DT and 6P learning routines. The output generated is the cell words in between or the route taken between the two words. This routine checks all the words in the input for connections to each other.

## 6P2 second 6p algorithm experimental

This is an experimental routine and produces no output. Results are displayed on the green screen. This algorithm needs a lot of monitoring to see if the idea is any good and worth using as an output.

Similar to six p but used a different method of getting to linked words. Maps data like cd's laid out on floor with each cd representing an entry in the Brain the route is north east west south and words used are those it passes through

#### **BA Best Answer**

The Choose Answer routine makes the decision about which output to use. It does this by comparing the reply qualities and previous answers. Aiming to find an output that hasn't been used in the last 9 outputs and with the highest reply quality.

# **CR Create Reply or Keyword**

Look at Keywords for details on the functions

# **GIT Gap In Time**

The amount of time the program takes to process an input is monitored, if there has been a long gap. The operation field in cell 929 is the amount of time in seconds to Waite before triggering the GIT response which is in field 8

# HTML web page created

If create html is ticked in settings then a web page is created using the information in cell 832. <u>Look in system cell details for more</u>

# **IN input from input box**

The input from the input box

## MSG message

Indicates an incoming message from another program or from the network

## **OF Output Filter**

Performs various filters on the output.

There are 2 main functions used by the filter these are function 2 (FN2) and function 4 (FN4) functions 1&3 are either in experimental mode or have been superceded by another function.

FN2 and FN4 use the SC code of the outputted sentence to re organize the words in the sentence.

FN2 uses a logic algorithm to processes the output

FN4 uses the structure of SC's already in the BD. SC's in the BD are added when SC learning is on or by the keyword SCFromText. SC codes are created from inputs so it is assumed that these inputs are structurally correct as they usually come from a person. The function compares the SC created from the output with found SC's in the BD. A percentage value is created which reflects the accuracy of the output. The threshold value is set by the Order field in cell 1605, if the percentage value is lower then FN2 output is used instead. If the Order value in cell 1605 is 0 then it is assumed that the cell is not set up so the threshold value is set to 50%

## OU output sent to output buffer

Outputs sent to the output buffer

# **OUT Output**

Main output from the Ai

# Pass number of processes or algorithms used to process the input

The pass number is incremented as it passes through various routines, it gives an idea of how much processing various inputs use.

# PLY play mp3 music

Indicates the playing of a mp3 audio file

# **REP Reprocess Input**

If no output is found then sometimes the input is reprocessed

#### **TF Translation File**

the name of one of the data bases used by Ai versions prior to the rebuild version  ${\rm AiV}016a$ 

#### Pre set words

Pre set words are words embedded in the program, **exitai** shuts down the Ai program and runs the StartAi.exe program **inputkeyword** pops up a messagebox with the words after input keyword in it

#### Search

The search routine is designed to do different jobs depending and the algorithm that is using it.

**RAS** uses the routine most so it is advisable to have the number of files searched limited to an acceptable amount depending on how fast the response is required. The file in cell 819 field 11 is the file name of the file that holds the file names of the files to search each time by the **RAS** routine.

If you want to have a large number of files searched and don't want the **RAS** routine to search all files each translation then it is best to use a different search file. If you use the keyword **searchinfofile** for a pattern then the file in 9 is the file holding the file names that will searched. This allows for different search criteria depending on what has been activated from the translation. You could have a search file full of say html file names that is used when the pattern for say 'search html for fish' is entered into the input.

#### **DLF DownLoad File**

Used by the Keywords DownLoadFile and DownLoadFileWait, also used by the separate download file program. Used to download files from the internet.

DLF error codes

error 1, unable to read file: dlf info.txt

error 2, no local filename

error 3, no online filename

error 4, problem downloading file

error 5, problem creating file

# PTL Print To Log

Writes to the log

**ISF Image Scan Function** 

# LD Light Dark

value set by ISF

# TRD Thread

## Settings

The Settings dialog box controls various functions within the Ai. Some settings can be controlled by keywords.

**Show Brain Database** opens the BD dialog box.

**Small display** Shows small display, see <a href="mailto:small.html">small.html</a>

**Image window** opens or closes the image window. The Image window is embedded in the User Window (UW). Cell 948 controls its size and position within the UW

**Web browser** Browser window used for displaying internet pages and web sites. **Stats** Statistics graph is drawn on the image window. The Stats are information gathered from the processing of the inputs, different routines and algorithms set the states of the stats.

**Analog inputs** Shows the analogue inputs from the USB card, the inputs are only fed into the input or output if this is ticked.

**WSB window** if ticked shows the Wheatstone Bridge window.

**User Window** visible if ticked. The user window is controlled by cell 933 UW Tags are used to set up elements like edit boxes etc... The DX and Image windows are integrated in the UW

**Focus on Net msg** sets the focus on the Ai when a new network message is received.

**Direct X window** opens or closes the DX window. The DX window is embedded in the User Window (UW). Cell 947 controls its size and position within the UW.

Direct X is a way of controlling hardware like graphics cards and using there processing resources. The Ai uses the DX window for image scanning.

**Information** general program information.

**Image** this is the name of an image file in ..\Ai\image\ the name of the current image is displayed and the name is saved when the program is closed.

**Ai name** the name of the Ai.

**Password** the password required before a network order will be carried out. **Window X Y** the coordinates on the screen of the Ai program top left of the Ai window.

# Audio player

Scroll bar shows the position in the track and also enables the track to be moved

to a position.

Edit box is the track number playing or the track to go to if changed and go to is pressed.

goto go to the track number

< go back a track

**S** stop play back

**P** play the play list default being TempPlayList.txt in the data folder.

> move to next track.

Timer and Timer events things that happen with the timer

**Timer Millisec** the main system timer controls how long to wait before putting text into input.

**Short term memory** some times referred to as **Learning file length** if set at 10 then the last 10 inputs are remembered and if a repeat pattern found then the pattern is put into the TF. Also used by subject.

**T (Timer) Loop** this is the extra loop for things like clearing the screen the variable is increased by one each time the timer activates when the timer loop value is reached then the extra commands are activated

**Max NN Generations** the maximum number of Neural Network generations before stopping the breading cycle

# **RQ** Weights

The Ai translates the inputs with different algorithms and routines, the routine asses the quality of the reply produced and sets the Reply Quality (**RQ**). The highest RQ that the routine can set is determined by its weight. The the output from the routine that produces the highest RQ is sent to the Ai output. Pressing the RQ defaults restore the RQ's to the original settings that were used before the option to change was created. The keyword **setrqweights** can also be used to set the RQ's. The numbers in field 11 correspond to the info below the maximum number is 9 all 16 RQ's are set in order if the number is short then the remaining RQ's wont be set eg. if 11=123 then SSC LM and TF RQ's will be set.

default as a number is: 444444441234444440043334

1 RQ SSC or QI default 9 (Special Sentence Code, functions things like the 'is' algorithm also)

2 RQ LM default 3 (Learning Mode)

3 RQ TF SC,3=3 or 3=2 default 7 (translate input using Translation File data base also Sentence Code)

4 RQ TF command, function, and some modes. The weights of certain functions is changed in the program by some code that use this weight w=weight. mode 5=w, mode 6=w+1, mode 7=w, function=w, system=w\* 4, 1 st pass TF=w, image=w -3, read file in 10=w, read file in readout dir=w, if name=w -1

5 RQ TF QNV default 9 (Question Noun Verb TF routine)

6 RQ EI default 6 (check to see if anything is Expected in the Input)

7 RQ CR default 3 (any special Create Reply's needed)

8 RQ PR1 default 8 (Pattern Recognition part 1)

9 RQ PR2 default 3 (Pattern Recognition part 2)

10 RQ NN default 5 (Neural Network)

11 RQ RAS1 default 5 (Reference Action Subject part 1)

12 RQ RAS2 default 7 (Reference Action Subject part 2)

13 RQ RAS3 default 8 (Reference Action Subject part 3)

14 RQ SJ1 default 8 (Subject part 1)

15 RQ SJ2 default 6 (Subject part 2)

16 RQ STM default 4 (Short Term Memory sometimes refereed to as data store)

The button **RQ Defaults** uses the information in cell 822 to set the weights.

**Save to Brain** saves the RQ's to a new entry in the Brain.

Random RQ generates a random number for the values

#### USB to DMX K8062

Tick this to use the K8062 DMX controller used to control DMX devices like stage lights.

Sometimes it can be difficult to get DMX to work, if you keep getting error 600 when running DMX check routine then close the Ai run the program DMX\_demo.exe in the main Ai folder. This is the demo software that comes with the K8062 DMX controller check the DMX is working then keep this program running in the background and then run the Ai. A fix for this problem is being worked out.

Sometimes the hardware freezes this can happen if the computer goes into hibernation or sleep. To remove the freeze unplug the usb cable then plug it back in.

#### **USB** interface K8055

The Velleman K8055 card has 5 digital inputs, 8 digital outputs, 2 analog inputs and 2 analog outputs.

**Clear d1 to d4 next Loop** this control is to be re-assigned as the function can be done in other ways.

**Clear d1 to d4 T Loop** this control is to be re-assigned as the function can be done in other ways.

**Check after first T Loop** checks the digital inputs after the next timer loop. **Check when d OFF** checks the digital inputs when digital outs are off.

#### In/Out

**Activate 1602** on input the cell is activated when an input is received. **Activate 1178** on output the cell is activated when the SayPrint routine sends outputs to various places.

**Output to FTP** sends the output to the FTP server. Cell 927 controls the FTP **Output to clipboard** sends the output to the clipboard so it can be pasted into other places

**Send msg to window** sends window messages to other programs running on the computer. Cell 918 is used to control how it works.

**Receive window msg** receives incoming messages from other windows. Used mostly by AiServer and AiCompass for passing information to the Ai Example code to place in a html document if you want to activate a cell from a web page button

<input type="submit" name="aicell505pw " value="FORWARD">
505 is the cell number to activate **aicell** is the code used by the server to activate a cell the server program extracts the information received then formats it into a string to send to the ai. The code **ac:** is used to instruct the Ai to activate a cell, **aicell** can be used if you want the form to be processed directly by the Ai without going through the server, used if the web page sends information to the Ai via a text file or a UDP packet.

The compass program sends a message to the Ai which sets cell inputs. To stop the cell being activated the string sent to the Ai contains **na**: which stands for no activation the string must contain **ac**: in order to activate a cell from a window message **na**: stops activation and just sets the cells input code.

#### **UDP**

UDP is a protocol similar to TCP and works on ip networks. UDP is a cruder way of transmitting data over a network and doesn't have the same error checking as TCP it is a lot faster and is used for things like streaming media. The Ai uses UDP in order to maximize the functionality of the network communication. System cell 809 is used to control how UDP works.

Enable UDP turns on UDP functions

Firewall's can block access to the network if you get a pop up message saying do you want to let Ai access the network then this is the firewall asking if its ok for

the program to access the net click yes to enable communication click no to stop the process. \* Make Rx socket will appear for as long as messages cant be received a firewall blocking access will do this. Receiving is passive so you may need to type in something for the send routine to open the socket again more applicable with a firewall.

If TX: <Ai output> appears and the UDP status and code don't change then there is a problem sending the message, a firewall blocking the message will do this. Cell 809 file field is used to add extra codes.

**RX** the ip number of your computer the local machine in-coming communications are sent to this ip number

**Port** All Ai's must use the same port number. Use a port number that no other program on the computer is using. Avoid common ports e.g. web browsers use port 80 and many other programs use ports. I use a port number higher than 10000.

**Auto RX** sets the local network IP number automatically. To gloom all IP numbers set RX to 0.0.0.0 and un-tick Auto RX

**TX** the ip number of the remote computer running the other Ai you are communicating with. If you type in a host name then the program will look up the IP number of the host name.

**Pass** the password of the Ai you are trying to communicate with, messages still get through but any extra commands are rejected

**Auto TX** this changes the TX IP number to the IP number of an incoming packet if an IP number is found.

**Read text file IN** reads a file from disk or network. The first line of the file must contain the password found in the Password box if the same then the second line in the file is put into the Ai input. If the input is the same as the previous one then it won't be carried out unless Allow repeat is ticked.

**To output** reads the file to the output.

**To input** reads the file to the input.

**Allow repeat** allows repeated sentences to be read.

**Del aft read** deletes the file after reading.

Edit box is the full path file name of the file to read.

**Write text file OUT** writes to a text file onto the disk or network.

**Input to file** writes the input to the file.

**Output to file** writes the output to the file.

**Create HTML 831 DB** creates a HTML file using the information in cell 831. Edit box is the full path file name of the file to write to.

**General** various functions including learning modes.

**Log activity** This switches on logging a new log is created each day with the file name being the date stored in ..\Ai\logs\ the name of this log is put into the translation file under 'log' when the program starts. The input, output, date, time and reply quality are put in the log. Reply quality is a value between 0 and 15. This is what the Ai thinks is the quality / how good the reply is.

SC: = sentence code

RQ: = reply quality

**Log stats** logs the stats numbers.

**Disable Defrag** disables the defrag process which runs at startup or when the Defrag button is pressed in the BD dialog.

**Set Input focus** sets the input focus to the Ai window.

**Auto Ai** sends the output to the input.

**Send to tray** sends the Ai program to the system tray.

Learning boxes if ticked switch on the associated learning processes.

LM learning

**6P learning** 

**WSB** learning

**RAS** learning

**SC** learning

NN learning

**Filters** filters the input or output

**Auto input OFF** disables automatic input, stops text written into the input being put automatically into the input.

**Input filter** filters the inputs, removes full stops and other things.

**Output filter** filters the output, capitalizes the first letter and adds a full stop. The output is processed through the grammar filter, this attempts to reorganize the words into proper english. The input code of cell 1605 controls which outputs are filter

wsb 6p lm2 nnec dt did lm qnv

1 1 1 1 1 1 0 1 0

lm is set at 0 because some of these outputs are pre programmed statements **Firewall** stops system commands.

**No output** suppress the Ai output Prevents the Ai's output from getting to the speech and print buffer but all other Ai routines still function.

Help opens the help file.Update updates the settings.OK updates the settings and closes the settings dialog box.