Great Cow BASIC

Index	Issues and Features in the release	Chip Type	Workaround
~~KI#1~~	~~Goto errors~~ a. ~~Avr - goto none existent_label… still compiles~~ b. ~~Avr - gosub unspecified sub …. Still compiles~~	~~AVR~~	~~With the full extent of the capabilities of Great Cow BASIC try not to use goto and/or be careful when coding.~~
KI#2	AVR pins may not be automatically setup to outputs or inputs as expected.	AVR	Set DIRection manually.
KI#4	Parsing of constants creates variables. It you accidently type a decimal number as 01 instead of 1 the compiler creates a variable "01" rather than treating the field as a numerical constant "1".	AVR & PIC	Always inspect your code for leading Zeros
~~KI#5~~	~~Cannot set function value with the result of function. An example~~ ~~function **myfunc**~~ ~~i2cstart '…. And other preamble~~ ~~i2crecieve (**myfunc**) ' does not set myfunc~~ ~~I2cend~~ ~~end function~~	~~AVR & PIC~~	~~Set results to a variable then return the value.~~ ~~function **myfunc**~~ ~~i2cstart '…. And other preamble~~ ~~i2crecieve (**myvar**) ' does not set myfunc~~ ~~I2cend~~ ~~**myfunc = myvar**~~ ~~end function~~

KI#6	Functions are not resolved on complex conditional statements but still compiles create bad ASM
	`#chip 16F1847, 32`
	`#config Osc = intOSC, MCLRE_ON, PLLEN_ON`
	
	`while ( !( ( f_one ) & ( f_two ) ) )`
	
	`end`
	
	`function f_one`
	`f_one = true`
	`end function`
	
	`function f_two`
	`f_two = true`
	`end function`
	
	As the second function is not defined… we should have an error.
		PIC, maybe AVR also	Expand all conditional tests to variables or one conditional test per line.

KI#7	A Sub and a Function of the same name may operate as expected	AVR and PIC	Please ensure you place the Function before the Sub
KI#8	Hex file generated may report incorrect file size. Internal structure of the generated hexfiles is good however.	AVR	Please no not worry.
KI#9	The compiler will only enforce things fitting on PIC 12F/16F, where each page must have subroutines carefully allocated to it to fit things in. On AVR or PIC 18F, the code that places subroutines onto program memory pages just dumps everything onto a single page. It doesn't complain about excessively sized programs, because some optimisation is done after that and often drops the size of the code. At the moment the compiler relies on the assembler or programmer to pick up anything that's still too big after optimisation.	PIC	Please watch file size generated.
~~KI#10~~	~~USB functionality not working~~	~~AVR and PIC~~	~~Supported~~
~~KI#11~~	~~SD Card functionality not working~~	~~AVR and PIC~~	~~Supported~~
KI#12	Two dimensional arrays do not work	AVR and PIC	Use your own methods to create multi-dimension arrays
KI#13	Enumerations do not work	AVR and PIC	Not supported
KI#24	Word search in GCGB does not work	AVR and PIC	Not supported
KI#15	Throwing compiler error condition from user code does not work	AVR and PIC	Not supported
~~KI#16~~	~~Microcontrollers with SFR bit IOSCFS will not set the chip frequency for the internal oscillator.~~ ~~Internal chip frequency (for these specific microcontrollers) can be set to 8mhz or 4mhz. You **MUST** set the configuration as follows for the differing~~	~~PIC~~	~~Include fix in future releases~~

	~~frequencies.~~ ~~4mhz = #chip <chipname>, 4~~ ~~#config IOSCFS _OFF~~ ~~8mhz = #chip <chipname>, 8~~ ~~#config IOSCFS _ON~~		
KI#17	When using GCASM you may an error - **Error: GCASM: Symbol RP0 has not been defined.** To resolve please use Great Cow BASIC to generate your ASM then use MPASM to generate your Hex file.	PIC	Include fix in future releases
KI#18	Chips of the type #chip 12F510. May issue warning **Warning: Current chip does not have enough common (non-banked) RAM.** This is warning and the code may operate as required however to resolve change the use of the wait or time related instructions to remove warning message. The NoBankRAM section in the chip data file refers to the area of microprocessor RAM that can be accessed regardless of which bank is currently selected. Example, the 16F59 does have RAM from 0x0A to 0x1F in bank 0, but only 0x0A to 0x0F can be accessed while another bank is selected. If for example bank 3 is selected, the PIC would be working with addresses between 0x60 and 0x7F. Accessing 0x6A to 0x6F would map back to 0x0A to 0x0F, but accessing 0x70 would access location 0x70 (not 0x10).	PIC	Include fix in future releases
KI#19	You cannot define a label within a macro. Defining a macro within a macro will cause the compiler to hang.	AVR and PIC	Do not do this. Very, very bad practice.