Support Group Application Note
Number: 033
Issue: 1
Author: Alastair France

Using BAS128 on the Master 128 and Master Compact

Applicable Hardware :

Master 128
Master Compact

Related Application Notes:

Support Group
Acorn Computers Limited
Acorn House
Vision Park
Histon
Cambridge CB4 4AE

BAS128 is a version of BBC BASIC which uses Sideways RAM for storage of programs and data. Normal versions of BBC BASIC use main memory for storage of programs and data, meaning that only space between OSHWM and the top of RAM (&8000) is available for such storage. With the default value of OSHWM at &E00 this means that there is only 28.5k free for programs and data. Bas128 can use up to 4 banks of sideways RAM - the full amount in the Master 128 or Master Compact - giving 64k free.

The disadvantage of this is that because the RAM is 'paged', access to programs and data in Bas128 is much slower. Therefore it is recommended that the version of BBC BASIC that is installed in the machine is used if program and data space is not at a premium.

Bas128 differs from the version of BBC BASIC in the Master (BASIC IV) in the following respects:

* The assembler will not cope with the extra 65C12 instructions beyond normal 6502 instructions.

* Use of operands beginning with A will cause problems with instructions which will accept accumulator addressing. For example, LSR ANSWER will assemble as LSR A.

* The A in accumulator addressing must be upper case. 'lsr a' will not work as 'LSR A'.

* The interpreter will not strip trailing spaces from input line. Therefore entering lines such as:
  '10 PRINT"HELLO"   will waste space.

* Leading spaces will not be stripped when non-zero LISTO is set.

* LISTO will not always indent loops correctly.

* LIST IF is not available.

* Renumbering of lines containing the double height character (141) can caused 'Failed at ...' errors. This is because the token for line number is also 141.

* An open files length cannot be changed by using 'EXT#chan=length'.

* TIME$ is not available.

* AUTO puts a space after the line number.

* Problems can occur using functions as formal parameters for FOR/NEXT loops which themselves contain loops. For example:
  ```
  DEF FNQ:FORJ%=1TO10:PRINT J%:NEXT:=10
  FOR I=FNQ-9 TO FNQ STEP FNQ/10
  ```
  will not work as expected in Bas128.

* EDIT cannot be used from Bas128.

* The | character cannot be used to insert trailing zeros in a VDU command.

* ONx ..... will only work with GOTO and GOSUB. ON x PROCa,PROCb,PROCc will not work.

* The 64k of additional memory is treated as a continuous address space above the normal 64k address space of the computer. The addresses &000000 to &00FFFF refers to the normal memory, while addresses &010000 to &01FFFF refers to the 64k bytes of sideways RAM.

  In Bas128 PAGE defaults to &10000. HIMEM defaults to &20000. LOMEM and TOP are addresses in a similar manner. It should not be necessary to change the values of PAGE or HIMEM, except perhaps to allocate space for machine code. If they are adjusted remember to set them to sensible values within the extra address space - it is possible to overwrite the BASIC interpreter by injudicious use. Remember to be careful when using ?, $ and ! for similar reasons.

* Statements such as:

  ```
  DIM code% 100
  ```

  will assign a 17-bit value to code%. Indirection operators such as !, ? and $ will address normal memory if the eventual pointer is between &000000 and &00FFFF, and will address sideways RAM if the pointer is between &010000 and &01FFFF.

* In assembler, O% and P% are regarded as 17 bit pointers. As above, if the address used is less than &010000 then normal memory is referenced, if &010000 or higher then sideways RAM is used.

* As the microprocessor cannot arrange for switching between banks of RAM, code running in one bank cannot call code or refer to memory locations in another bank in normal assembler code. Such references will therefore give rise to a Bank error; eg:

  ```
  P%=&13F00:[JSR &1CF34
  ```

  will give this error. Beware of referencing sideways RAM memory using constructions such as:

  ```
  LDA #&80
  STA &71
  LDA #0
  STA &70
  TAY
  .loop
  LDA (&70),Y
  ADC #6
  STA (&70),Y
  INY
  BNE loop
  ```

  This may not produce the expected result. Altering the value of ROMSEL (&FE30) may cause unexpected results.

  Code generated that crosses a bank boundary is also trapped - producing a Wrap error. Execution cannot pass directly from one bank to another.

  ```
  P%=&13FFC:[LDA #0:TAY:SEC:SBC &70
  ```

  will give a Wrap error.

* Shadow mode is forced on - Bas128 resides in RAM between &3000 and &8000. Remaining memory between OSHWM and &3000 is used by Bas128 for workspace.

Transferring Bas128 from tape to disc

Connect your data recorder and disc drive and insert the Master Welcome tape into your data recorder and a formatted disc into your disc drive and type the following:

```
*SHADOW
MODE0
*TAPE
*LOAD BAS128 2800
*DISC
*SAVE BAS128 2800+24
*TAPE
*LOAD BASOBJ 3000
*DISC
*SAVE BASOBJ 3000+5000 3100 3000
```