

PC -1500/A

**16 x 16K Combo FLASH-RAM module
(CE163F)**

and

**8 x 16K RAM Module
(CE1638)**

User Manual

**Version 1.1
31 July 2023**

| | |
|--|-----------|
| REFERENCES | 4 |
| INTRODUCTION | 5 |
| HARDWARE INSTALLATION | 5 |
| CE163F MEMORY MODULE | 7 |
| Flash Memory Module Command Description | 7 |
| Switch to another bank | 8 |
| Check the bank number | 8 |
| Copy to another bank | 8 |
| Save BASIC RAM pointers to memory bank | 8 |
| Update BASIC RAM pointers from the memory bank | 9 |
| Memory Module Recovery | 9 |
| Install and format the module | 9 |
| Update the firmware image via a Flash programmer | 11 |
| Update the firmware image via the PC-1500/A | 11 |
| Update Loader image | 12 |
| Install keyboard interception | 12 |
| Files | 12 |
| CE163F optional hardware configurations | 13 |
| CE163F Command Summary | 13 |
| CE1638 MEMORY MODULE | 16 |
| RAM Memory Module Command Description | 16 |
| Switch to another bank | 16 |
| Check the bank number | 17 |
| Save BASIC RAM pointers to memory bank | 17 |
| Update PC-1500/A BASIC RAM pointers from memory bank | 17 |
| Memory Module Bank Restore | 17 |
| Install keyboard interception | 18 |
| Install and format the module | 18 |
| Files | 19 |
| CE1638 optional hardware configurations | 19 |
| CE1638 Command Summary | 19 |
| ANNEX A: CE1638 MEMORY MODULE SOFTWARE | 21 |
| ANNEX B: CE163F EXAMPLE - UPDATE | 22 |
| ANNEX C: CE163F EXAMPLE – UTIL | 24 |
| ANNEX D: CE163F FLASH MEMORY ORGANIZATION | 26 |
| Flash structure | 26 |

References

1. Microchip FLASH SST39SF010A
<https://ww1.microchip.com/downloads/en/DeviceDoc/20005022C.pdf>
2. Alliance RAM AS6C1008-55TIN
<https://www.alliancememory.com/wp-content/uploads/pdf/AS6C1008feb2007.pdf>

Welcome to the Sharp PC-1500/A Memory Expansion Modules User Manual

Introduction

The very popular Sharp PC-1500, and later PC-1500A, both shipped with limited user memory. Expansion memory modules were produced to increase the built-in memory size but 35 years later these original memory modules are increasingly difficult to find. Consequently, upgrading these machines is no easy task. The two Memory Expansion Modules described in this user manual offer unique memory expansion capabilities for the PC-1500/A*. They allow the memory to be expanded by 16Kbytes and provide the ability to switch between a number of 16K memory banks. In the case of the CE1638 memory module there are eight 16K RAM banks. For the CE163F there are 16 x 16K banks, eight banks of RAM and eight of non-volatile FLASH memory.

Both Memory Expansion Modules use a 128Kbyte static RAM chip that has battery back-up using either a CR2032 (CE1638) or CR2016 (CE163F) button cell. The CR2032 should give a few months of backup if the CE1638 is without power from the PC-1500/A. The CR2016 will provide around four weeks of backup on the CE163F. The backup batteries are intended to protect the memory while the main AA batteries are being swapped. They also allow the card to be moved to other machines while preserving the contents. Although, it is always good practice to backup any work on the Memory Expansion Modules before replacing the AA batteries or removing the card.

As both Memory Expansion Modules allow bank switching the useable memory size is limited to 16Kbytes (address range 0 - &3FFF - 16384 bytes[#]) although the PC1500/A will report a larger memory size. If any program/data in the 16K address range extends beyond that limit switching to another bank would destroy the program. If a program/data larger than 16K is required, then the bank switching feature can be ignored/removed and the full memory size utilized. The memory beyond the 16K boundary is used to store the utilities that perform the management of the memory banks in the case of the CE163F.

*Throughout the manual the term PC-1500/A is used to refer to both the PC-1500 and PC1500A when there is no distinction to the user in how the Memory Expansion Module behaves. Should a distinction need to be made then the specific machine will be highlighted either as the PC-1500 or PC-1500A.

#The “&” is used to represent a hexadecimal number in line with PC-1500/A nomenclature.

Hardware Installation

Installation of the Memory Expansion Module is very straightforward. If you are not comfortable working with the PC1500/A at the hardware level, then do not proceed further. Any damage to the PC-1500A, Memory Expansion Module or data loss from use of either Memory Expansion Module is the user's responsibility. Always take the necessary anti-static precautions when handling the module.

Before undertaking any insertion or removal of the Memory Expansion Module, it is essential that at least one of the AA batteries providing power to the PC1500/A is removed. This will prevent possible damage to the Memory Expansion Module and/or the PC-1500/A.

First, remove the cover from the memory expansion bay on the PC-1500/A. Next, ensuring the side of the Memory Expansion Module with the button cell is facing up, gently insert the memory module into the memory expansion connector. Again, be gentle as this should require only minimal force. If the force seems excessive, check that the card edge is lined up with the connector slot. The Memory Expansion Module should be pushed in until the edge connector is no longer visible.

Once the Memory Expansion Module is installed, replace the AA batteries and follow the instructions for installing the software for the Memory Expansion Module in the appropriate section below.

To remove a Memory Expansion Module, first make sure the AA battery power is disconnected. To help remove the Memory Expansion Module the button cell holder can be used to very gently pull the card out of the connector.

⚡WARNING⚡

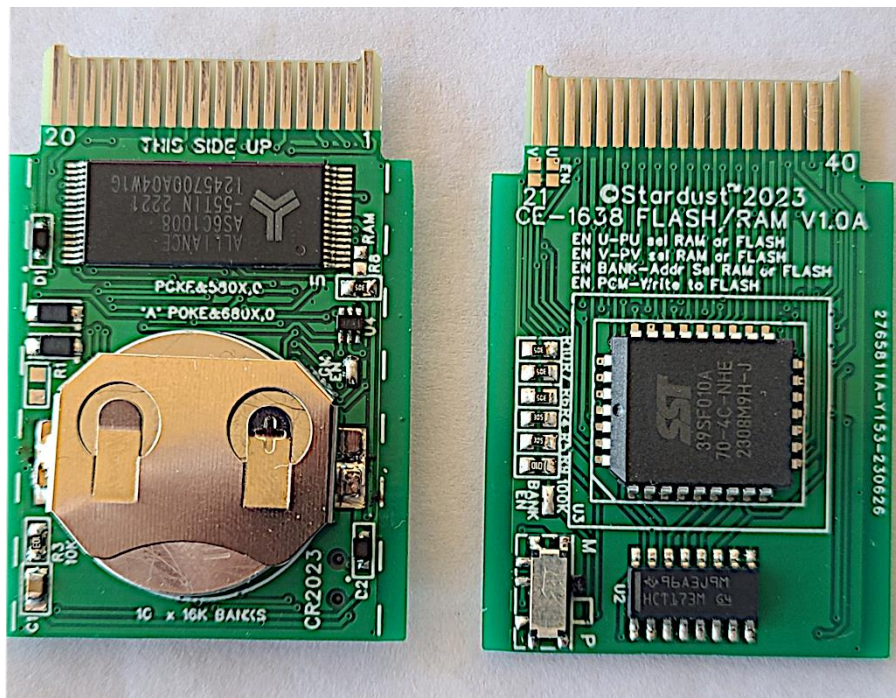
The Memory Expansion Module is static sensitive like any other piece of electronic equipment. Make sure to store any removed Memory Expansion Module in a static-proof bag until needed. When handling the memory module do so by the edges and refrain from touching the exposed edge connector or components.

CE163F Memory Module

The CE163F memory module for the PC-1500/A consists of two sections: a 128Kbyte Static RAM and 128Kbyte FLASH Memory. Each of the sections is subdivided into eight 16Kbyte banks, this gives 16 x 16K banks, eight banks of RAM and eight banks of FLASH, however only seven banks of the FLASH are useable. The banks are numbered 0 – 15 (0 - &F). The banks can be switched by writing to address range &5800-&580F (PC-1500) or &6800-&680F (PC-1500A). For example, bank 9 could be accessed by using POKE &5809,0 on a PC-1500 or POKE&6809,0 on a PC-1500A. However, this direct method of moving between banks will NOT work when program/data is stored in the bank as the PC1500/A keeps track of the memory contents and the tracking data will be lost. Therefore, to make switching banks transparent to the PC-1500/A a small piece of machine code/firmware is stored in each bank taking up about 70bytes of the bank's memory. As a consequence, when using the NEW command it should be NEW&112 instead of NEW0.

The firmware on each bank (RAM and FLASH) is stored between addresses &C5-&C112 and is used to switch between banks or to copy one bank to another. Copying to a FLASH bank (banks 9-15) will preserve the program/data even with total loss of power. To provide the copy command and a few maintenance features, bank 8 (first bank in the FLASH memory) is reserved for the exclusive use of the memory module firmware. No attempt should be made to copy or switch to this bank in normal usage. Bank 8 is accessed in a few cases outlined below in the description of the commands.

No attempt should be made to enter a program when the bank selected is in the FLASH memory. If this does occur it may produce confusing results. Immediately switch to another bank using the method described below.



CE163F FLASH/RAM memory module (modules may vary in construction)

Flash Memory Module Command Description

The following sections provide a description of each CALL that can be used with the CE163F expansion memory module.

Switch to another bank

To switch between banks simply set X to the destination bank number and then make the CALL. After the CALL the bank selected should be set.

```
X=bank [0-7,9-&F]
```

```
CALL&E3,X
```

For example, to switch to bank 9:

```
X=9
```

```
CALL&E3,X
```

Check the bank number

Occasionally switching banks can lead to some confusion as to which bank is current. To confirm the current bank the PEEK command can be used as follows:

```
PEEK &E2
```

This will return the current bank 0...15. If it does not return a bank number in that range or the expected bank number, then the bank maybe damaged. In which case disable the keyboard interception and then copy an undamaged bank into this bank to rebuild the bank. This will delete any data on the repaired bank.

Copy to another bank

To copy the current bank to another destination bank simply set X to the destination bank number and then make the CALL. After the CALL the contents of the current bank should have been copied to the destination bank. The Reserve Memory data is also copied along with any program/data.

```
X=Destination bank [0-7,9-&F]
```

```
CALL&C6,X
```

For example, to copy from the current bank to bank 12:

```
X=12 [or X=&C]
```

```
CALL&C6,X
```

Save BASIC RAM pointers to memory bank

Within a RAM bank this will save the BASIC pointers used by the PC-1500/A to delimit the BASIC program in memory to the RAM bank memory. Use this CALL if planning to remove the bank so the RAM bank pointers are updated. Set X = the current bank then uses the normal switch CALL. This will not work on the FLASH banks and is not required.

```
X= Current Bank [0-7]
```

```
CALL&E3,X
```


For example, to update the bank memory on bank 7.

```
X=7  
CALL&E3,X
```

Prior to removing the memory module disable the keyboard interception as follows:

```
POKE&79D4,0
```

Update BASIC RAM pointers from the memory bank

Within a memory module RAM bank this will copy the bank memory pointers to the PC-1500/A BASIC RAM pointers.

Use this if NEW0 was used instead of NEW&112 (NEW274), do not enter any program before doing so.

```
X=8  
CALL&E3,X
```

Memory Module Recovery

After moving a memory module or accidentally doing NEW0 instead of NEW&112 to restore a program use the following options. This option does a more extensive recovery operation. It is only required if the regular method fails.

For the PC-1500:

```
POKE&5808,0
```

For the PC-1500A:

```
POKE&6808,0
```

To issue the recovery command:

```
X=&FF  
CALL&C6,X
```

This will also restore the keyboard Intercept routines.

Once completed use the switch CALL to move back to a regular bank.

Install and format the module

Any damage to the PC-1500/A, Memory Expansion Module or data loss from use of either Memory Expansion Module is the user's responsibility. Always take the necessary anti-static precautions when handling the module.

Once the memory module has been installed and the AA batteries replaced press the ON key to power up the PC1500/A. The NEW0:Check? message should appear on the screen, proceed as normal and use:

NEW0

This should bring up the regular screen prompt '>'. At this point typing MEM this should indicate that the memory size has now increased by 16Kbytes, confirming the memory module is installed.

Formatting the memory will delete ALL the program/data stored in the memory banks. There are options to format either the whole memory module, the RAM section of the memory module or the FLASH portion of the memory module.

To install the firmware required to support the memory module move to memory bank 8 by typing (never use this method to switch to another bank otherwise program/data will be lost corrupting any BASIC programs stored in the bank):

For the PC-1500:

```
POKE&5808,0
```

For the PC-1500A:

```
POKE&6808,0
```

To format the whole memory module type:

```
X=0
```

To format just the RAM portion of the memory module type:

```
X=8
```

To format just the RAM portion of the memory module type:

```
X=15 (X=&F)
```

For the PC-1500:

```
CALL&290,X
```

For the PC-1500A:

```
CALL&29A,X
```

The screen will display the progress of the format process by showing:

```
FMT: RAM FLASH
```

Depending on the value of X either both RAM and FLASH will be displayed or just RAM or FLASH.

Once the format is complete bank 0 (for example) can be reached by typing:

```
X=0
```

then

```
CALL&E3,X
```

Update the firmware image via a Flash programmer

The CE163F is designed to be programmed using a normal FLASH programmer (e.g., T866IIplus) and a reversed PLCC adapter. The current flash used is the Microchip SST39SF010A. Before (re)programming the FLASH memory backup the Memory Expansion Module as all the data will be erased. The memory will need to be formatted again before use.

Before inserting the Memory Expansion Module into the PLCC adapter the slide switch should be set to P = Program. The slide switch should be set to M = Memory for use in the PC-1500/A. Forgetting to return the slide switch to M can cause odd behavior on the PC1500/A.

NOTE: When programming the FLASH memory via a FLASH programmer any data stored in the FLASH memory will be lost. Data in the RAM may also be corrupted.

Update the firmware image via the PC-1500/A

The utilities/firmware used to control the memory module is stored in the first 4Kbytes of bank 8. It is possible to update the utilities/firmware, if new features are added or performance improvements made or if users wish to add their own features, by using the PC-1500/A itself. Although this method is more tedious it will achieve the same result.

NOTE: When programming the FLASH memory data maybe lost on the module as a precaution it is good practice to backup any data on the module.

Updating the firmware requires the use of bank 0 RAM, therefore any program/data stored in that bank should be backed up before proceeding. The update process should not delete any program/data stored on the memory module, however as a precaution it would be wise to back up any program/data before continuing.

If not already on bank 0 switch to bank 0 (use the CALL outlined above if required) and load the provided firmware image. The image will be quite large. At the end of the set of BASIC DATA statements will be a checksum and number of lines make note of this data it will be used later in the update process.

To install the firmware update, move to memory bank 8 by typing (never use this method to switch to another bank otherwise program/data will be corrupted):

For the PC-1500:

```
POKE&5808,0
```

For the PC-1500A:

```
POKE&6808,0
```

Set X= No Lines in Image, provided with image file:

```
X=No lines
```

There is a loader built into the bank 8 utilities this needs to be called to install the new image:

For the PC-1500:

```
CALL&1500,X
```

For the PC-1500A:

```
CALL&150A,X
```

When complete type X<enter> the resulting value should match the checksum provided with the image file.

If this is the first use of the memory then proceed to the section ‘*Install and Format the module*’, this will set up the memory for use. Otherwise move to the desired bank to continue using the PC-1500/A.

Update Loader image

As with the utilities/firmware section it is also possible to update the loader routine used to load the new utilities/firmware. This process is a little more involved and should rarely be required.

NOTE: When programming the FLASH memory data may be lost on the module, as a precaution it is good practice to backup any data on the module.

Updating the loader firmware requires the use of bank 0 RAM therefore any program/data stored in that bank should be backed up before proceeding. The update process should not delete any program/data stored on the memory module, however as a precaution it would be wise to back up any program/data before proceeding.

If not already on bank 0, switch to bank 0 (use the CALL outlined above if required) and load the provided firmware image. The image should not be very large. At the end of the set of BASIC DATA statements will be a checksum and number of lines make a note of this information it will be used later in the update process.

The loaded program will consist of two distinct sections: a DATA section and a program section comprising a series of POKE statements. Once the program is loaded switch to RUN mode and then RUN the program. After the program is complete the updater routine will have been loaded into RAM at address &4100.

Set X= No Lines in Image, provided within the image file

```
X=No lines
```

Then

```
CALL&4100,X
```

This will update the Loader image.

When complete type X<enter> the resulting value should match the checksum provided with the image file.

Install keyboard interception

To simplify the use/maintenance of the Memory Expansion Module the OFF key is intercepted and used to update the BASIC memory pointers before entering Auto Power down mode. This helps maintain the integrity of the memory module banks and to prevent the PC-1500/A requesting NEW0 each time it is powered on when a FLASH bank is selected. The intercept is installed and enabled along with the formatting of the memory and can be re-enabled by the recovery command outlined above. The interception should be disabled any time the memory module is removed.

To enable keyboard interception. NOTE: the intercept routine is part of each memory banks support data:

```
POKE &785B,&00,&C6  
POKE &79D4,&55
```

To Disable keyboard interception

```
POKE &79D4,&00
```

Files

FULLUTIL.IMG: Flash utilities image used with flash programmer, includes utils and updater code. This file can be used to program the FLASH memory with a regular FLASH Programmer and PLCC adapter.

Usage of the following two files is described in the sections on updating the firmware and loader images.
 UTIL.BAS: Flash utils update image, load into PC1500/A and install with updater located at &290/&29A

UPDATE.BAS: Update flash updater, load into PC1500/A. RUN, then set X=no lines, CALL&4100,X

CE163F optional hardware configurations

As can be seen on the CE163F memory card there are several solder bridges that allow the card to be configured for different options by the user. The options are as follows:

PGM EN: When bridged this option allows writing to the FLASH memory using the Flash key access sequences. If this is not bridged then the Flash can only be used as Read Only Memory. The current flash used is the Microchip SST39SF010A, datasheet link: <https://ww1.microchip.com/downloads/en/DeviceDoc/20005022C.pdf>

The following bridge options are exclusive and cannot be combined doing so would render the PC-1500/A and the memory module inoperative.

BANK EN: When bridged, this option uses the address range to switch between the RAM and FLASH memory.

U EN: When bridged this option uses the PU signal to switch between the RAM and FLASH memory.

V EN: When bridged this option uses the PV signal to switch between the RAM and FLASH memory.

RAM EN: For the curious, this is used for assembly testing.

NOTE: ONLY the PGM EN and BANK EN options are supported by the preloaded firmware. It is envisioned that Memory Expansion Module users might reprogram the module for their own requirements.

CE163F Command Summary

| Call Function | X Setting | Operation | Comment |
|--------------------------------|---------------------------|---|--|
| Bank Switching Commands | | | |
| CALL&E3,X | X=Dest Bank [0-7,9-&F] | Switch to Bank X | Save the PC1500/A BASIC pointers into the bank memory before switching to the new bank, after switching update the PC1500/A BASIC pointers with the destination BASIC pointers. |
| CALL&E3,X | X=Current Bank [0-7] | Update Bank X. | Update the PC1500/A BASIC pointers into bank memory. Use before removing the RAM Module. The FLASH memory cannot be updated this way. |
| CALL&E3,X | X=8 | Update PC1500/A memory. | Update bank memory BASIC pointers into PC1500/A memory. Use after accidental NEW0 command it, will restore the BASIC program use NEW&112. The FLASH memory cannot be updated this way |
| PEEK &E2 | N/A | Returns the number of the current bank. | Identifies the current bank, useful to navigate round the module. |
| Copy Command | | | |

| | | | |
|---|--|--|--|
| CALL&C6,X | X=Dest. Bank [0-7,9-&F] | Copy current bank of memory to Dest. X. | Copy the contents of the current bank to the Destination bank. The contents of the Destination bank will be overwritten. |
| Maintenance Related Commands | | | |
| 1500: POKE&5808,0 CALL&290,X 1500A: POKE&6808,0 CALL&29A,X | X=0 Or X=8 Or X=&F | Format the RAM & FLASH sections of the Module. Format the RAM section of the module. Format the FLASH section of the module. | Use to format the module on first use or to repair damaged banks. The contents of the RAM and/or FLASH will be overwritten. |
| POKE&5808,0 or POKE&6808,0 CALL&C6,X | X=&FF | Restore the module and <input type="checkbox"/> OFF key interception routines. | Use this command after moving a Module or if something stops working. |
| Update Related Commands | | | |
| 1500: POKE&5800,0 1500A: POKE&6800,0 1500: POKE&5808,0 1500A: POKE&6808,0 1500: CALL&1500,X 1500A: CALL&150A,X | X=No Lines X=Checksum | utilities image update. | Set bank to 0. Load the UTIL.BAS file into BANK 0. Note Checksum and value of X. Move to BANK 8. Call update routine. Screen will show FLASHING... while it updates the utilities. When complete type X the value should match the checksum |
| 1500: POKE&5800,0 1500A: POKE&6800,0 CALL&4100,X | X=No Lines X=Checksum | update image update. | Set bank to 0. Load the UPDATE.BAS file into BANK 0. Note Checksum and value of X. RUN the loaded program. CALL &4100 with X = set as shown in UPDATE.BAS When complete enter X the value should match the checksum |
| POKE&79D4,0 | | Disable the Keyboard interception. | This will disable the keyboard interception of the <input type="checkbox"/> OFF key and stop the automatic updating of the bank BASIC pointers. |

| | | | |
|-------------------------------------|--|-----------------------------------|---|
| | | | Use this command before removing the RAM module |
| POKE &785B,&00,&C6 POKE&79D4,&55 | | Enable the Keyboard interception. | This will enable the keyboard interception of the <input type="checkbox"/> key. |

CE1638 Memory Module

The CE1638 memory module for the PC-1500/A uses a 128Kbyte Static RAM subdivided into eight 16Kbyte banks, these banks are labeled 0 – 7. The banks can be switched by writing to address &5800-&5807 (PC-1500) or &6800-&6807 (PC-1500A). For example, bank 7 could be accessed by using POKE &5807,0 on a PC-1500 or POKE&6807,0 on a PC-1500A. However, this method of moving between banks will NOT work when program/data is stored in the bank as the PC1500/A keeps track of the memory contents and the tracking data will be lost, corrupting any program/data stored in the bank. Therefore, to make switching banks transparent to the PC-1500/A a small piece of machine code/firmware is stored in each bank taking up about 60bytes of the bank's memory. As a consequence, when using the NEW command it should be NEW&104 (NEW260) instead of NEW0.

The firmware on each bank is stored between addresses &C5-&C104 and is used to switch between banks.



CE1638 Memory Module

RAM Memory Module Command Description

Switch to another bank

To switch to another bank simply set X to the bank number and then make the CALL. Then after the CALL the bank selected should be set.

X=Bank [0-7]

CALL&D5,X

For example, to switch to bank 7:

X=7

CALL&D5,X

Check the bank number

Occasionally switching banks can lead to some confusion as to which bank is current. To confirm the current bank the PEEK command can be used as follows:

PEEK &D4

This will return the current bank 0 to 7. If it does not return a bank number in that range then the bank maybe damaged. In which case disable the keyboard interception and then copy an undamaged bank to this bank to rebuild the bank. This will delete any data on the bank.

Save BASIC RAM pointers to memory bank

Within a RAM Bank this will save the BASIC pointers used by the PC-1500/A to the bank memory. Use this if planning to remove the bank. Set X = the current bank then use the normal switch bank CALL.

X=Current Bank [0-7]

CALL&D5,X

For example, to update the bank memory when on bank 4.

X=4

CALL&D5,X

Prior to removing the memory module disable the keyboard interception as follows:

POKE&79D4,0

Update PC-1500/A BASIC RAM pointers from memory bank

Within a memory module RAM bank this will copy the bank memory pointers to the PC-1500/A BASIC RAM pointers.

Use this if NEW0 was used instead of NEW&104 do not enter any program before doing so.

X=8

CALL&D5,X

Memory Module Bank Restore

To restore a damaged bank first move to a known good memory module bank. All the data on the damaged bank will be lost. First disable the keyboard interception if it has been enabled.

POKE &79D4,0

Set X to the damaged bank number

X=Damaged bank [0-7]

If the PC-1500/A has not been reset or otherwise cleared the routine originally poked in to format the memory should still be present. If not then it will need to be restored, see section on formatting the memory. Then make a CALL to the routine.

```
CALL&410E,X
```

The CALL will restore the bank and then move to that bank. Then:

```
POKE&D4,Current Bank [0-7]
```

Restore the keyboard interception if required.

```
POKE&79D4,&55
```

For example, if memory module bank 1 is damaged then from an undamaged bank and assuming that the firmware is still resident and intact at address &4100 perform the following steps:

```
X=1
```

```
CALL&410E,X
```

```
POKE&D4,1
```

The keyboard intercept routine can be restored if desired.

Install keyboard interception

To simplify the use of the memory module the **OFF** key is intercepted and used to update the BASIC memory pointers before entering Auto Power down mode. This helps maintain the integrity of the memory module banks. The intercept is installed along with the formatting of the memory but must be enabled post format. The interception should be disabled any time the memory module is removed.

To enable keyboard interception, the intercept routine is part of each memory banks firmware:

```
POKE&785B,&00,&C6  
POKE&79D4,&55
```

To Disable keyboard interception

```
POKE&79D4,&00
```

Install and format the module

Any damage to the PC-1500A, Memory Expansion Module or data loss from use of either Memory Expansion Module is the user's responsibility. Always take the necessary anti-static precautions when handling the module.

Once the memory module has been installed and the AA batteries replaced press the ON key to power up the PC1500/A. The NEW0:Check? message should appear on the screen, proceed as normal and use:

```
NEW0
```

Now enter the program show in Annex A. Once entered change mode to RUN and run the program. This will store a small machine code program in memory starting at address &4100. Once installed then enter the following CALL:

CALL&4100

This CALL will format the memory module and make it ready for use. Once the ">" prompt returns type:

PEEK &D4

This command should return "7" the final bank and shows the CALL has worked.

Once the memory module has been formatted use NEW&104 (NEW260) and not NEW0 for clearing memory. If required enable the keyboard interception routine as described above.

To simplify the switching of banks it is possible to set up the Reserve Memory to perform the CALLs. For example:

Set the '!' Reserve Key to: CALL&D5,X@

Set the "" Reserve Key to: PRINT"BANK ";PEEK&D4@

Then to switch banks simply enter X=dest bank and use '!' key. To show the bank then press the "" key.

The reserve keys will need to be set for every bank.

Files

The bank switching program file and as shown in Annex A
BANKSW.BAS

CE1638 optional hardware configurations

As can be seen on the CE163F memory card there is one solder bridge and an option for a slide switch. The slide switch when installed can be used to disable writes to the RAM memory module protecting any contents. The memory module can then be used as ROM for storing programs that will not change. The solder bridge 'R/W-' allows for a permanent setting of write access. Either the solder bridge can be closed, or the switch can be installed and used, both cannot be used together.

CE1638 Command Summary

| Call Function | X Setting | Operation | Comment |
|--------------------------------|-------------------------|---|---|
| Bank Switching Commands | | | |
| CALL&D5,X | X=Dest Bank [0-7] | Switch to Bank X | Save the PC1500/A BASIC pointers into the bank memory before switching to the new bank, after switching update the PC1500/A BASIC pointers with the destination BASIC pointers. |
| CALL&D5,X | X=Current Bank [0-7] | Update Bank X. | Update the PC1500/A BASIC pointers into bank memory. Use before removing the Memory Expansion Module. |
| CALL&D5,X | X=8 | Update PC1500/A memory. | Update Bank memory BASIC pointers into PC1500/A memory. Use after accidental NEW0 command it, will restore the BASIC program use NEW&104. |
| PEEK&D4 | N/A | Returns the number of the current Bank. | Identifies the current bank, useful to navigate round the Module. |

| | | | |
|------------------------------------|--|------------------------------------|--|
| POKE&79D4,0 | | Disable the Keyboard interception. | This will disable the keyboard interception of the OFF key and stop the automatic updating of the bank BASIC pointers. Use this command before removing the Memory Expansion module |
| POKE&785B,&00,&C6 POKE&79D4,&55 | | Enable the Keyboard interception. | This will enable the keyboard interception of the OFF key. Only enable the key interception routine once the banks have been formatted. |

Annex A: CE1638 Memory Module Software

Bank firmware formatting routine. See installation and formatting instructions for the CE1638 module.
Bank switching software not including Reserve Memory set up.

BANKSW.BAS

```

10 POKE &4100, &B5, &00, &BE, &41, &0F, &DD, &B7, &08, &99, &08
20 POKE &410A, &BE, &00, &D6, &9A, &04, &2A, &FD, &C8, &68, &58
30 POKE &4114, &2E, &68, &68, &2E, &B5, &00, &1A, &18, &6A, &C4
40 POKE &411E, &51, &88, &03, &4A, &30, &48, &41, &6A, &3F, &F5
50 POKE &4128, &88, &03, &FD, &8A, &AE, &00, &D4, &9A, &FF, &BE
60 POKE &4132, &E2, &4A, &B7, &0F, &89, &30, &BE, &00, &D3, &BA
70 POKE &413C, &E3, &3F, &4A, &00, &04, &5A, &FE, &58, &00, &4A
80 POKE &4146, &65, &48, &78, &FD, &88, &FD, &98, &B7, &08, &8B
90 POKE &4150, &0A, &BE, &00, &F8, &2A, &68, &58, &2E, &68, &68
100 POKE &415A, &2E, &FD, &0A, &FD, &1A, &E9, &78, &63, &00, &6A
110 POKE &4164, &05, &F5, &88, &03, &9A, &01, &04, &01, &04, &01
120 POKE &416E, &04, &FF
130 REM POKE &758B, &0, &C6 : POKE &794D, &55

```

Bank switching software including Reserve Memory initialization. The ! and “ keys in Reserve Memory II can be used to switch bank (set X=Bank first) and print out the Bank number respectively.

BANKSWRM.BAS

```

10 POKE &4100, &B5, &00, &BE, &41, &0F, &DD, &B7, &08, &99, &08
20 POKE &410A, &BE, &00, &D6, &9A, &04, &2A, &FD, &C8, &68, &58
30 POKE &4114, &2E, &68, &68, &2E, &B5, &00, &1A, &18, &6A, &C4
40 POKE &411E, &51, &88, &03, &4A, &39, &48, &41, &6A, &3F, &F5
50 POKE &4128, &88, &03, &FD, &8A, &AE, &00, &D4, &58, &00, &5A
60 POKE &4132, &56, &6A, &19, &F5, &88, &03, &9A, &FF, &BE, &E2
70 POKE &413C, &4A, &B7, &0F, &89, &30, &BE, &00, &D3, &BA, &E3
80 POKE &4146, &3F, &4A, &00, &04, &5A, &FE, &58, &00, &4A, &65
90 POKE &4150, &48, &78, &FD, &88, &FD, &98, &B7, &08, &8B, &0A
100 POKE &415A, &BE, &00, &F8, &2A, &68, &58, &2E, &68, &68, &2E
110 POKE &4164, &FD, &0A, &FD, &1A, &E9, &78, &63, &00, &6A, &05
120 POKE &416E, &F5, &88, &03, &9A, &01, &04, &01, &04, &11, &04
130 POKE &4178, &FF, &11, &F1, &8A, &26, &44, &35, &2C, &58, &40
140 POKE &4182, &12, &F0, &97, &22, &42, &41, &4E, &4B, &20, &22
150 POKE &418C, &3B, &F1, &6F, &26, &44, &34, &40
160 REM POKE &758B, &0, &C6 : POKE &794D, &55

```

Annex B: CE163F Example - Update

Example to show the structure of the UPDATE.BAS program.

UPDATE.BAS

```

10 DATA 48008E08FFFFFFFFFFFF4801FD88BEEE716A696815B50BBEED004A7448155A00
20 DATA 58416A8FBE1565FD0A4204AE411C4C018B14B558AE414AAE4153AE416FAE4175
30 DATA AE4182AE418ABE4100B5000A085657FDCA5CFF99075EFF990BFD88BE00E1BEEE
40 DATA 71FD0AFB9AF588039A464C415348494E472E2E2EBE4144A5786508A578660AB5
50 DATA 05FDCAB5001A186AC4BE41635488066A006E008B22FDA86A1FAE5800AE6800BE
60 DATA ED95AE5808AE6808BE41635488156A05448803FD2A88229ABE417FB580AE6809
70 DATA AE1555BE417FAE6808B530AE0000A5000038B9809B089AB7FF8B17FDC8BE417F
80 DATA B5A0AE6809AE1555AE6808FD8A1E381799049AB5AAAE6809AE1555B555AE6808
90 DATA AE2AAA9AFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
100 REM CS: -29379 (0x8d3d) set X=9
105 POKE &4100, &04, &8B, &52, &DF, &FD, &C8, &BE, &41, &57, &A5
110 POKE &410A, &78, &65, &08, &A5, &78, &66, &0A, &B5, &05, &FD
120 POKE &4114, &CA, &B5, &00, &1A, &58, &15, &FD, &8A, &2A, &FD
130 POKE &411E, &A8, &6A, &1F, &AE, &58, &00, &AE, &68, &00, &BE
140 POKE &4128, &ED, &95, &AE, &58, &08, &AE, &68, &08, &BE, &41
150 POKE &4132, &7C, &54, &88, &15, &6A, &05, &44, &88, &03, &FD
160 POKE &413C, &2A, &88, &22, &B5, &00, &0A, &08, &56, &57, &FD
170 POKE &4146, &CA, &5C, &14, &99, &07, &5E, &FF, &99, &0B, &AE
180 POKE &4150, &58, &00, &AE, &68, &00, &FB, &9A, &BE, &41, &9E
190 POKE &415A, &B5, &80, &AE, &68, &09, &AE, &58, &09, &AE, &15
200 POKE &4164, &55, &BE, &41, &9E, &AE, &68, &08, &AE, &58, &08
210 POKE &416E, &B5, &30, &AE, &10, &00, &A5, &10, &00, &38, &B9
220 POKE &4178, &80, &9B, &08, &9A, &B7, &FF, &8B, &1D, &FD, &C8
230 POKE &4182, &BE, &41, &9E, &B5, &A0, &AE, &68, &09, &AE, &58
240 POKE &418C, &09, &AE, &15, &55, &AE, &68, &08, &AE, &58, &08
250 POKE &4196, &FD, &8A, &1E, &38, &17, &99, &04, &9A, &B5, &AA
260 POKE &41A0, &AE, &68, &09, &AE, &58, &09, &AE, &15, &55, &B5
270 POKE &41AA, &55, &AE, &68, &08, &AE, &58, &08, &AE, &2A, &AA
280 POKE &41B4, &9A

```

Flashing utility including Reserve Memory initialization. The !, “ and # keys in Reserve Memory II can be used to switch bank (set X=Bank first), print out the Bank number and print out version in bank 8 respectively.

UPDATERM.BAS

```

10 DATA 48008E08FFFFFFFFFFFF4801FD88BEEE716A696815B50BBEED004A7448155A00
20 DATA 58416A98BE1565FD0A4204AE411C4C018B14B558AE4153AE415CAE4178AE417E
30 DATA AE418BAE4193BE4100B5000A085657FDCA5CFF99075EFF990BFD88BE00E1BEEE
40 DATA 71FD0AFB9AF588039A464C415348494E472E2E2EBE414DA5786508A578660AB5
50 DATA 05FDCAB5001A186A55BE416C5488066A006E008B2BFDA86A1FAE5800AE6800BE
60 DATA ED95AE5808AE6808BE416C545C0089065E7989025AC5881FB506FDCAFD2A882B
70 DATA 9ABE4188B580AE6809AE1555BE4188AE6808B530AE0000A5000038B9809B089A
80 DATA B7FF8B17FDC8BE4188B5A0AE6809AE1555AE6808FD8A1E381799049AB5AAAE68
90 DATA 09AE1555B555AE6808AE2AAA9AFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
100 REM CS: -30440 (0x8918) set X=9
105 POKE &4100, &04, &8B, &52, &DF, &FD, &C8, &BE, &41, &57, &A5
110 POKE &410A, &78, &65, &08, &A5, &78, &66, &0A, &B5, &05, &FD
120 POKE &4114, &CA, &B5, &00, &1A, &58, &15, &FD, &8A, &2A, &FD
130 POKE &411E, &A8, &6A, &1F, &AE, &58, &00, &AE, &68, &00, &BE
140 POKE &4128, &ED, &95, &AE, &58, &08, &AE, &68, &08, &BE, &41
150 POKE &4132, &7C, &54, &88, &15, &6A, &05, &44, &88, &03, &FD
160 POKE &413C, &2A, &88, &22, &B5, &00, &0A, &08, &56, &57, &FD
170 POKE &4146, &CA, &5C, &14, &99, &07, &5E, &FF, &99, &0B, &AE

```

180 POKE &4150, &58, &00, &AE, &68, &00, &FB, &9A, &BE, &41, &9E
190 POKE &415A, &B5, &80, &AE, &68, &09, &AE, &58, &09, &AE, &15
200 POKE &4164, &55, &BE, &41, &9E, &AE, &68, &08, &AE, &58, &08
210 POKE &416E, &B5, &30, &AE, &10, &00, &A5, &10, &00, &38, &B9
220 POKE &4178, &80, &9B, &08, &9A, &B7, &FF, &8B, &1D, &FD, &C8
230 POKE &4182, &BE, &41, &9E, &B5, &A0, &AE, &68, &09, &AE, &58
240 POKE &418C, &09, &AE, &15, &55, &AE, &68, &08, &AE, &58, &08
250 POKE &4196, &FD, &8A, &1E, &38, &17, &99, &04, &9A, &B5, &AA
260 POKE &41A0, &AE, &68, &09, &AE, &58, &09, &AE, &15, &55, &B5
270 POKE &41AA, &55, &AE, &68, &08, &AE, &58, &08, &AE, &2A, &AA
280 POKE &41B4, &9A

Annex C: CE163F Example – UTIL

Example to show the structure of the UTIL.BAS program.

UTIL.BAS

```

10 DATA FFFD88BE00E1AE58086A0038BA0135BEE24AB70F8930BE00E1BAE33F4A08044A
20 DATA 6548785A0C5801FD88FD98B7088B0ABE01062A68582E68682EFD0AFD1AE97863
30 DATA 006A05F588039A0112012A0112000A1AF0972242414E4B382D56312E3020464C
40 DATA 415348220DFFFFFFFFFFFFFFFFFFFFFFFFFD0A0824284EFF891DE9786300E9785B
50 DATA 00B5D4AE785CEB79D4554A085AE35800FD98FD888E415A0B5801FD98FD884E08
60 DATA 8B314C088B2D84068B294E0883084AC048016A1F8E064AE048016AA75A005841
70 DATA F58803A48B1958686868FD0A041A842ABE4100FBEE6696A034ADC48019E2358
80 DATA 58685894AE4138AE4141AE4166AE417CAE4184AE4190AE41969E311EB5000A08
90 DATA FD88414C409105FD0A4A082E051E414C40910814AE00E2FD1A2E9ABE41252E4A
100 DATA 0048004C0089074EE28903148E0105BE415EBEE45189062E444C40911AFD0A9A
110 DATA FD88FDA804286A03BE41378805FD2AFD0A9A4868A40ABE4179B580AE6809AE15
120 DATA 55BE41790E24F1084A00B5300EBEE45189060538B9809B0B9AFDC8BE4179B5A0
130 DATA AE6809AE15551EFD8A0E38BEE45189030799099AB5AAAE6809AE1555B555AE68
140 DATA 08AE2AAA9ABE4179B590AE6809AE1555AE680838A5000008A500010AB5F0AE00
150 DATA 00FB9AFFFFFFFFFFFFFFFFFFFF48008E082052414DFFFF4801FD88BEEE716A246803
160 DATA B504BEED004A2E48035A0058416A7FBE0312FD0A4E0F8B14FD886A946802B504
170 DATA BEED00BE414EFD0A4E088B3784FDC86A286803B506BEED004AAE48035A4E5841
180 DATA 6AB3BE0312FD8A8917B558AE41ADAE41B4AE41DDAE41F3AE41FBAE4157AE418B
190 DATA BE414EBEEE71BE0316BE00E19AF588039AE9785B00B5D4AE785CEB79D4559A46
200 DATA 4D543A20464C415348FFFFD88BE00E1AE5808AE6808BA0135BEE24AB70F8930BE
210 DATA 00E1BAE33F4A00044A6548785A0C5801FD88FD98B7088B0ABE01062A68582E68
220 DATA 682EFD0AFD1AE97863006A05F588039A011201120112FF5A006A0068582E6868
230 DATA 2EB5000A08414C409105144A0048415AC558006A4DF58803AE00E2DD1A2AB708
240 DATA 9927AE5808AE68089ABE4198B509FDC82A68682E4A004800B500BE418E4EC599
250 DATA 095A0058414AC54C00890A4EE28906FD8AFDC88E0115BE418E545E4E9917FD8A
260 DATA DDB7109937AE68089ABE41D1BEE451990C449A680958076A03A40ABE41AC8807
270 DATA FD60FD52990F9A4868BE41F0B580AE6809AE1555BE41F00E24F1084A00B5300E
280 DATA BEE45189060538B9809B0B9AB7FF8B1AFDC8BE41F0B5A0AE6809AE15552EFD8A
290 DATA 0E38BEE45189030799099AB5AAAE6809AE1555B555AE6808AE2AAA9AFFFFFFFFF
300 REM CS: -30863 (0x8771) set X=29

```

Flashing utility including Reserve Memory initialization. The !, “ and # keys in Reserve Memory II can be used to switch bank (set X=Bank first), print out the Bank number and copy a bank (set X=Dest Bank first) respectively.

UTILRM.BAS

```

10 DATA 11F18A2645332C584012F0972242414E4B20223BF16F2645324013F097225631
20 DATA 312240FFFFD88BE00E1AE58086A0038BA0135BEE24AB70F8930BE00E1BAE33F4A
30 DATA 08044A6548785A0C5801FD88FD98B7088B0ABE01062A68582E68682EFD0AFD1A
40 DATA E97863006A05F588039A0112012A0112000A1AF0972242414E4B382D56312E31
50 DATA 20464C415348220DFFFFFFFFFFFFFFFFFFFFFFFFFD0A0824284EFF891DE9786300
60 DATA E9785B00B5D4AE785CEB79D4554A085AE35800FD98FD888E415A0B5801FD98FD
70 DATA 884E088B314C088B2D84068B294E0883084AC048016A1F8E064AE048016AA75A
80 DATA 005841F58803A48B1958686868FD0A041A842ABE4100FBEE6696A034ADC4801
90 DATA 9E235858685894AE4138AE4141AE4166AE417CAE4184AE4190AE41969E311EB5
100 DATA 000A08FD88414C409105FD0A4A082E051E414C40910814AE00E2FD1A2E9ABE41
110 DATA 252E4A0048004C0089074EE28903148E0105BE415EBEE45189062E444C40911A
120 DATA FD0A9AFD88FDA804286A03BE41378805FD2AFD0A9A4868A40ABE4179B580AE68
130 DATA 09AE1555BE41790E24F1084A00B5300EBEE45189060538B9809B0B9AFDC8BE41
140 DATA 79B5A0AE6809AE15551EFD8A0E38BEE45189030799099AB5AAAE6809AE1555B5
150 DATA 55AE6808AE2AAA9ABE4179B590AE6809AE1555AE680838A5000008A500010AB5
160 DATA F0AE0000FB9AFFFFFFFFFFFFFFFFFFFF48008E082052414DFFFF4801FD88BEEE716A
170 DATA 246803B504BEED004A2E48035A0058416AADBE0312FD0A4E0F8B14FD886A9468

```



```
180 DATA 02B504BEED00BE4171FD0A4E088B3784FDC86A286803B506BEED004ADC48035A
190 DATA 7158416AB9BE0312FD8A8917B558AE41D6AE41DDAE4206AE421CAE4224AE417A
200 DATA AE41B4BE4171BEEE71BE0316BE00E19AF588039AE9785B00B5D4AE785CEB79D4
210 DATA 559A464D543A20464C41534811F18A2645332C584012F0972242414E4B20223B
220 DATA F16F2645324013F18A2643362C5840FFFD88BE00E1AE5808AE6808BA0135BEE2
230 DATA 4AB70F8930BE00E1BAE33F4A00044A6548785A0C5801FD88FD98B7088B0ABE01
240 DATA 062A68582E68682EFD0AFD1AE97863006A05F588039A011201120112FF5A006A
250 DATA 0068582E68682EB5000A08414C409105144A0048415A5658006A22BE41AA5AC5
260 DATA 6A4DBE41AAAE00E2DD1A2AB708992EAE5808AE68089AF588039ABE41C1B509FD
270 DATA C82A68682E4A004800B500BE41B74E5699095A0058414A564C00890A4EE28906
280 DATA FD8AFDC88E0115BE41B74E7989024AC5545E71991DFD8ADDB710993DAE68089A
290 DATA BE41FABEE451990C449A680958076A03A40ABE41D58807FD60FD52990F9A4868
300 DATA BE4219B580AE6809AE1555BE42190E24F1084A00B5300EBEE45189060538B980
310 DATA 9B0B9AB7FF8B1AFDC8BE4219B5A0AE6809AE15552EFD8A0E38BEE45189030799
320 DATA 099AB5AAAE6809AE1555B555AE6808AE2AAA9AFFFFFFFFFFFFFFFFFFFFFFFFFFFF
330 REM CS: -2315 (0xF6F5) set X=32
```

Annex D: CE163F FLASH Memory Organization

Flash structure

The FLASH memory is divided into 32 x 4K sectors. Each sector can be individually erased. The FLASH memory on the module is arranged as shown below. The first four sectors contain the firmware used to manage the module. The remaining sectors are used as the seven banks.

Bank 0 - 000 \$00000 - \$03FFF = Sectors 0,1,2,3
16K Utilities, see table below

Usable BANKs on the Flash

Bank 1 (9)*- 001 \$04000 - \$07FFF = Sectors 4,5,6,7
Bank 2 (10)- 010 \$08000 - \$0BFFF = Sectors 8,9,10,11
Bank 3 (11)- 011 \$0C000 - \$0FFFF = Sectors 12,13,14,15
Bank 4 (12)- 100 \$10000 - \$13FFF = Sectors 16,17,18,19
Bank 5 (13)- 101 \$14000 - \$17FFF = Sectors 20,21,22,23
Bank 6 (14)- 110 \$18000 - \$1BFFF = Sectors 24,25,26,27
Bank 7 (15)- 111 \$1C000 - \$1FFFF = Sectors 28,29,30,31

*() = real bank ID on Memory card.

Bank 0 Organization

| FLASH Sector | Address | Contents | Arrangement |
|--------------|---------------------|--|---|
| SECTOR 0 | &0 | | Keys: !, ", # in Reserve Memory II |
| | &0056 | Start Reserve Mem | |
| | &0079 | End Reserve Mem | |
| | &00C5 | Start of Firmware | |
| | &00C6 | Copy Command | |
| | &00E3 | Switch Command | |
| | &0112 | Start of Program space in a regular Bank | Copy Loader + RAM Copy + FLASH Copy |
| | &0135 | Start of Copy Command | |
| | &0287 | End of Copy Command | Init Loader + RAM Format + FLASH Format |
| | &0290 | Start of Init Command | |
| &0494 | End of Init Command | | |
| &0FFF | | | |
| SECTOR 1 | &1000 | | |
| | &1500 | Flashing command | Flashing Loader + Flash Writer |
| | &160F | End of Flashing command | |
| &1FFF | | | |