

## How to use uBUG12

uBUG12 is a GUI to interface with Freescale's Serial Monitor that are pre-programmed into the NC12s and Adapt9S12E128 families. It has some similarities with Gordon Doughman's DBUG12.

uBUG12 can be downloaded from Technological Arts website  
<http://support.technologicalarts.ca/files/uBug12.zip>

For PCs with Windows98SE the .net framework must be installed in order for uBUG12 to run. WinXP, 2K the .net framework is (usually) already installed. The .net framework can be found at MS website

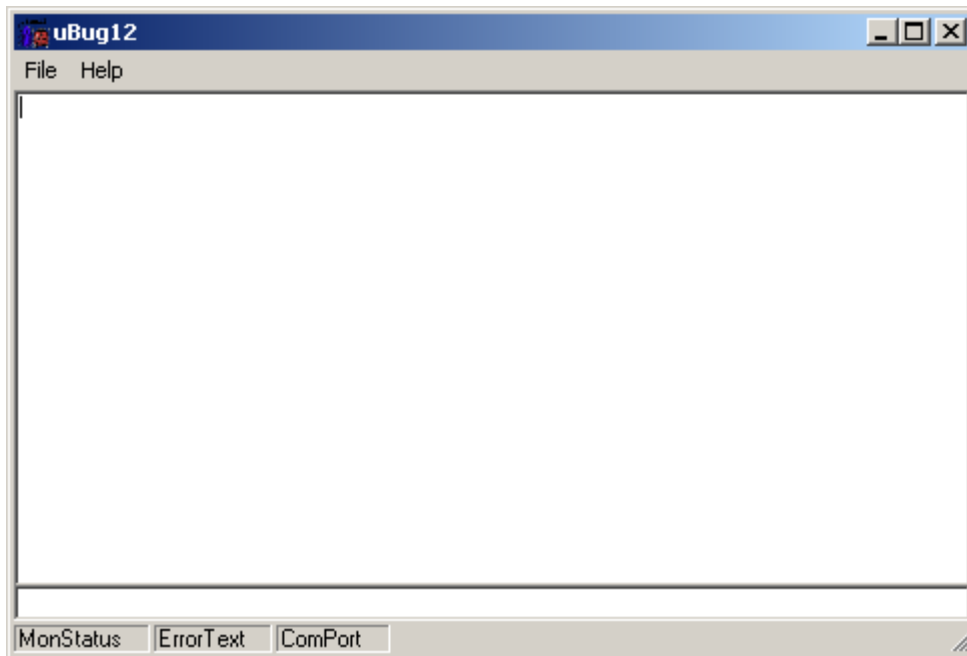
<http://www.microsoft.com/downloads/details.aspx?FamilyID=d7158dee-a83f-4e21-b05a-009d06457787&displaylang=en>

### Getting Started:

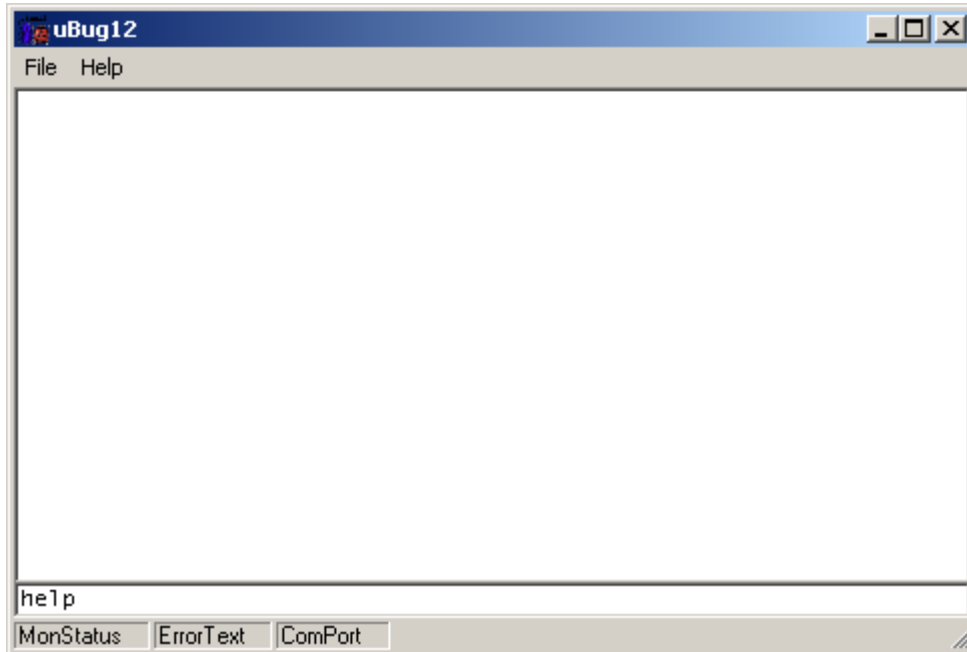
Double click on the uBUG12 icon to start GUI. Below is what uBUG12 started.



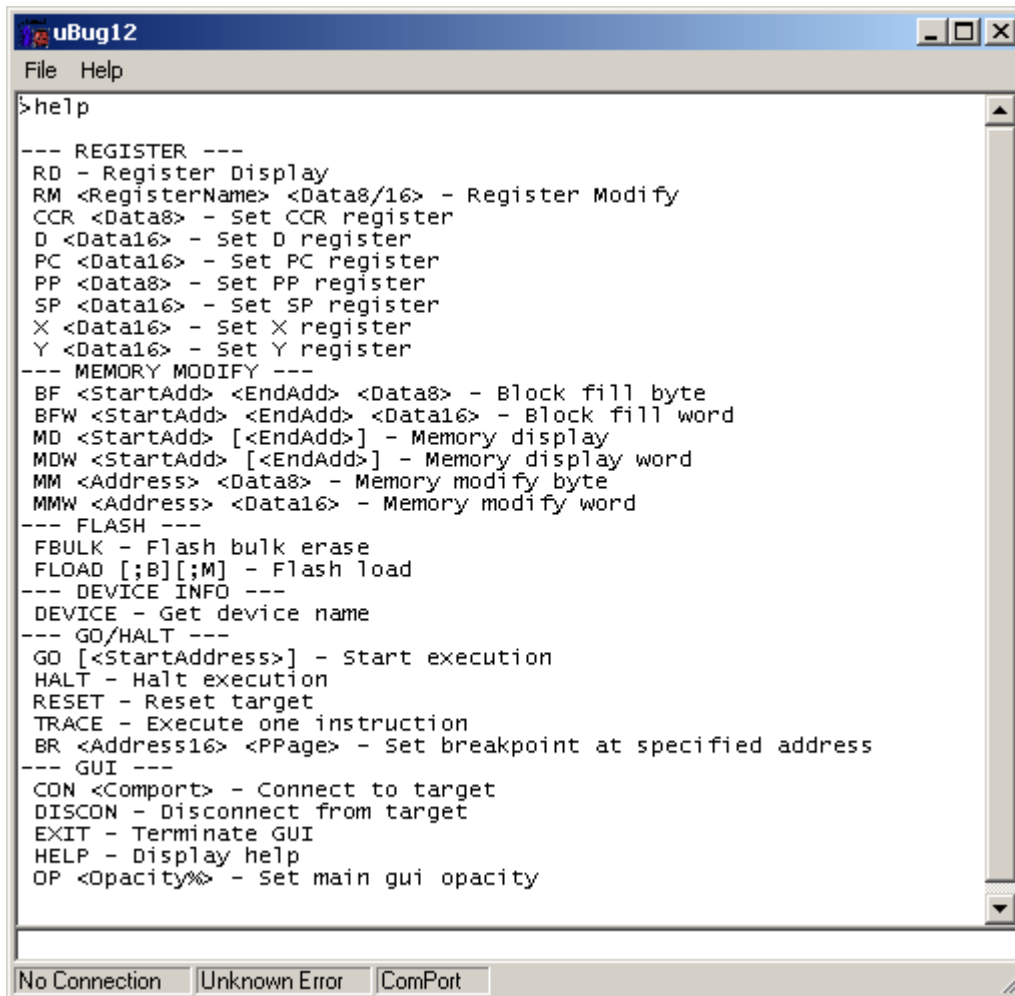
Here uBUG12 is waiting for commands. By typing **help** one can see different commands that can be use.



Type the **help** command



Once the help command is invoked, uBUG12 will list the different commands as shown.



The screenshot shows a window titled "uBug12" with a menu bar containing "File" and "Help". The main text area displays the output of the ">help" command, which lists various commands grouped into sections: REGISTER, MEMORY MODIFY, FLASH, DEVICE INFO, GO/HALT, and GUI. At the bottom of the window, there are three status buttons: "No Connection", "Unknown Error", and "ComPort".

```
>help
--- REGISTER ---
RD - Register Display
RM <RegisterName> <Data8/16> - Register Modify
CCR <Data8> - Set CCR register
D <Data16> - Set D register
PC <Data16> - Set PC register
PP <Data8> - Set PP register
SP <Data16> - Set SP register
X <Data16> - Set X register
Y <Data16> - Set Y register
--- MEMORY MODIFY ---
BF <StartAdd> <EndAdd> <Data8> - Block fill byte
BFW <StartAdd> <EndAdd> <Data16> - Block fill word
MD <StartAdd> [<EndAdd>] - Memory display
MDW <StartAdd> [<EndAdd>] - Memory display word
MM <Address> <Data8> - Memory modify byte
MMW <Address> <Data16> - Memory modify word
--- FLASH ---
FBULK - Flash bulk erase
FLOAD [<B>][<M>] - Flash load
--- DEVICE INFO ---
DEVICE - Get device name
--- GO/HALT ---
GO [<StartAddress>] - start execution
HALT - Halt execution
RESET - Reset target
TRACE - Execute one instruction
BR <Address16> <PPage> - Set breakpoint at specified address
--- GUI ---
CON <Comport> - Connect to target
DISCON - Disconnect from target
EXIT - Terminate GUI
HELP - Display help
OP <Opacity%> - Set main gui opacity
```

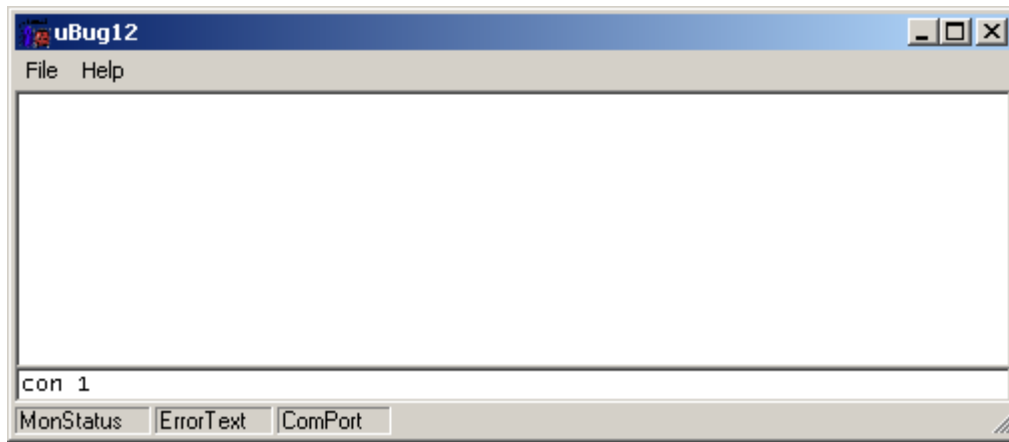
The different commands will be explained throughout this document. This document will use the NC12 module with a Docking module to help explain the command types.

## Connecting:

This document will use COM 1 of the PC to connect to the target as an example. For PC without serial port, a USB to COM can be purchase from any computer store.

Connect a Serial cable from COM 1 to the Docking Module. Slide the Run/Load switch to Load or Boot then power up the board. Make sure the power LED is on.

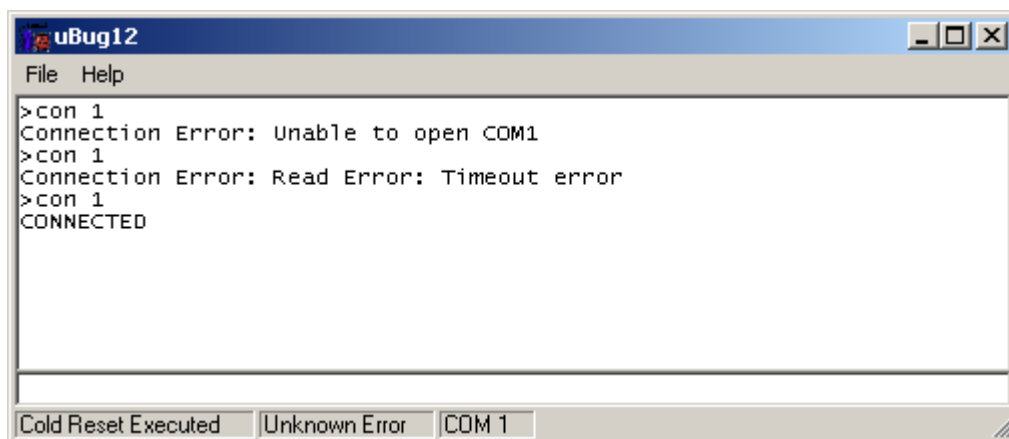
The command to connect is **CON 1** for COM 1 and **CON 2** for COM 2.



2 possible errors can occur:

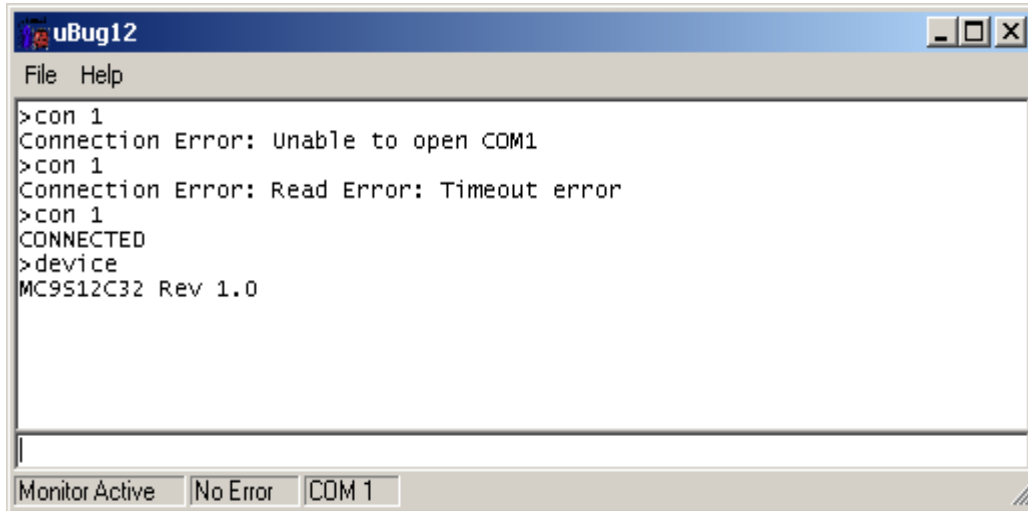
**Connection Error: Unable to open COM1** <- Another application is using the COM port

**Connection Error: Read Error: Timeout error** <- The MCU not currently in LOAD mode or the cable is disconnected from either PC or Docking Module, lastly the serial cable is connected on the wrong COM port.



A **CONNECTED** message will appear to show good connection between PC and the target.

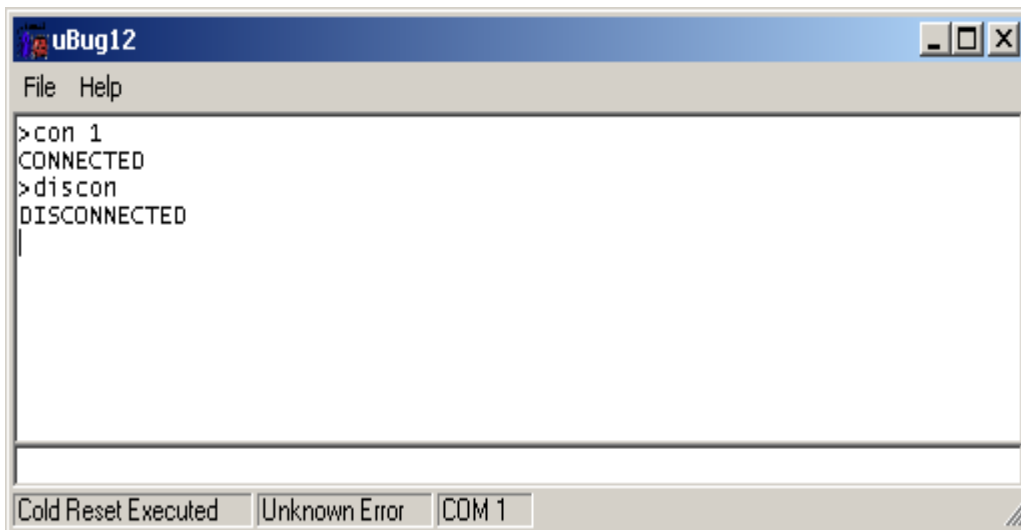
The **device** command will show the target type is as MC9S12C32 Rev 1.0.



```
uBug12
File Help
>con 1
Connection Error: Unable to open COM1
>con 1
Connection Error: Read Error: Timeout error
>con 1
CONNECTED
>device
MC9S12C32 Rev 1.0
Monitor Active No Error COM 1
```

### Disconnecting:

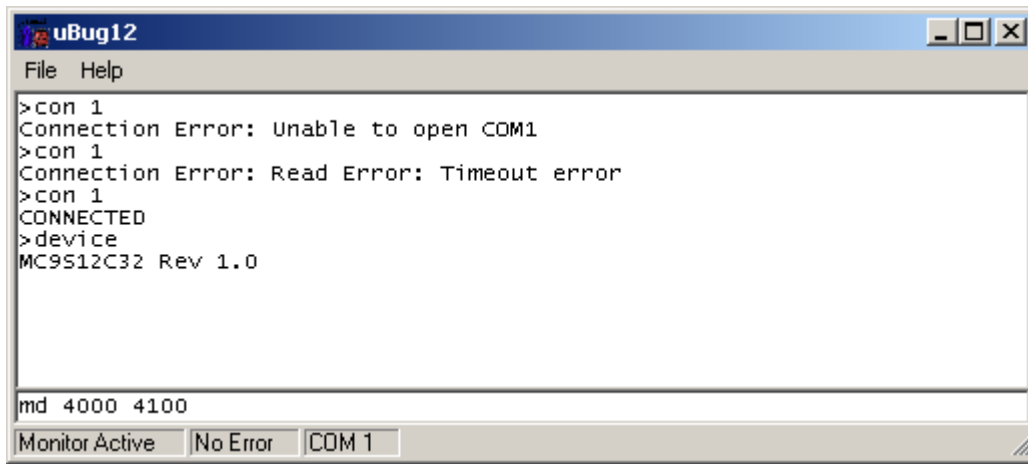
To disconnect uBUG12 from the serial port, the command **discon**. This would allow other application to use the COM 1 like HyperTerm or Tera Term.



```
uBug12
File Help
>con 1
CONNECTED
>discon
DISCONNECTED
Cold Reset Executed Unknown Error COM 1
```

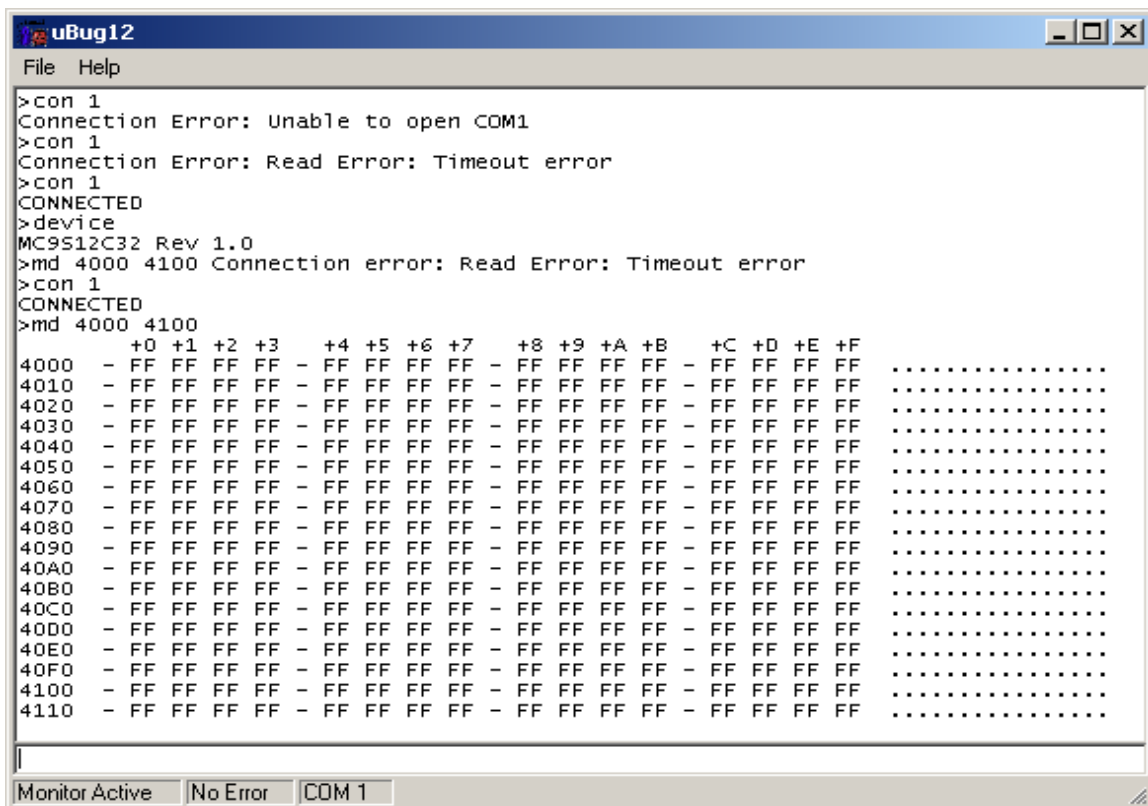
### Memory Dump:

The commands are **MD** and **MDW**. For example, type **md 4000 4100** to dump an address block.

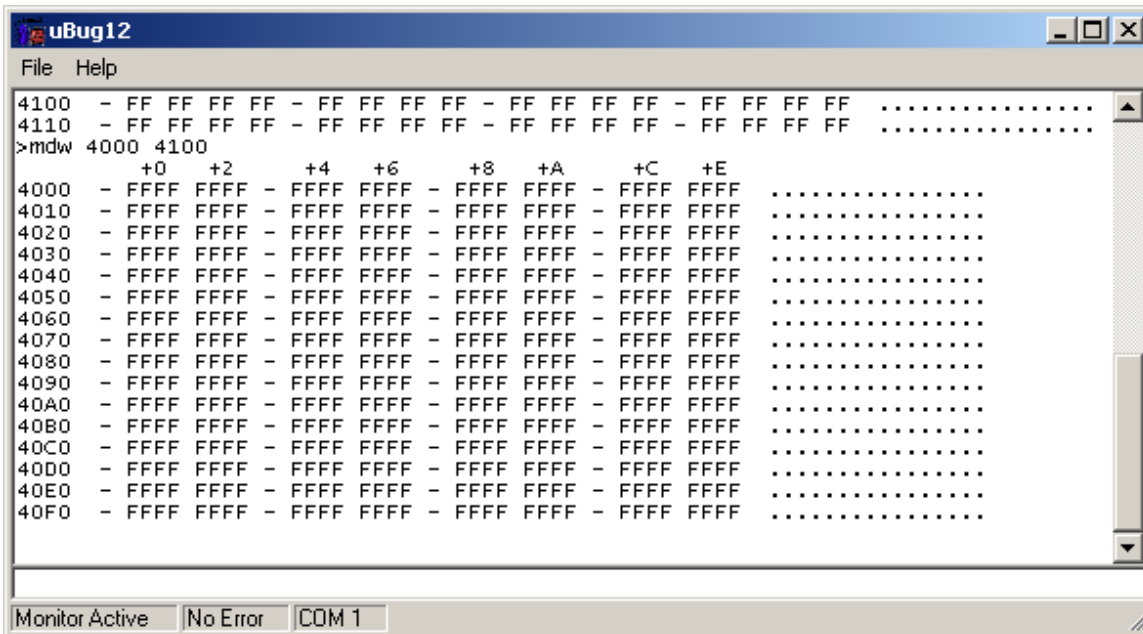


If for whatever reason the cable is disconnected, power disruption, MCU is reset the error may appear "**>md 4000 4100 Connection error: Read Error: Timeout error**". Reconnect again by typing **CON 1**.

The **MD** command is byte dump.

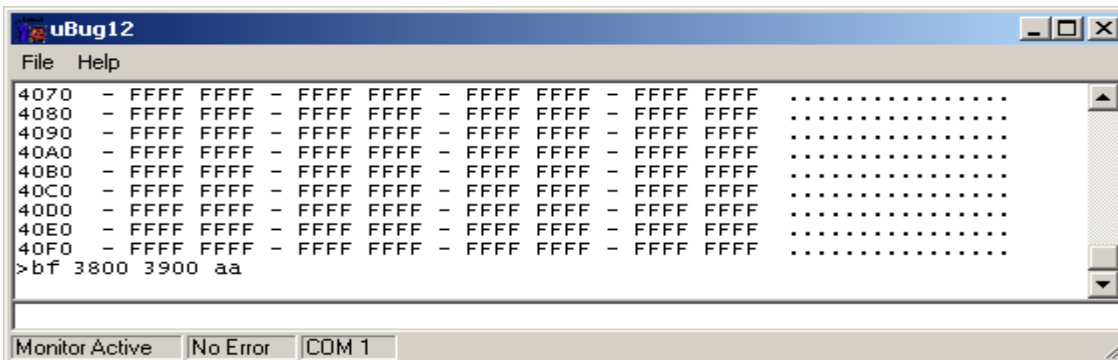
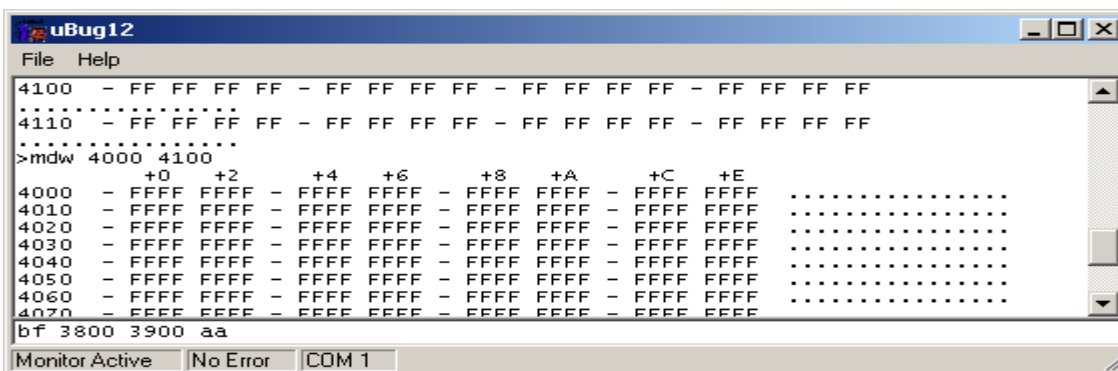


The **MDW** command is word dump.

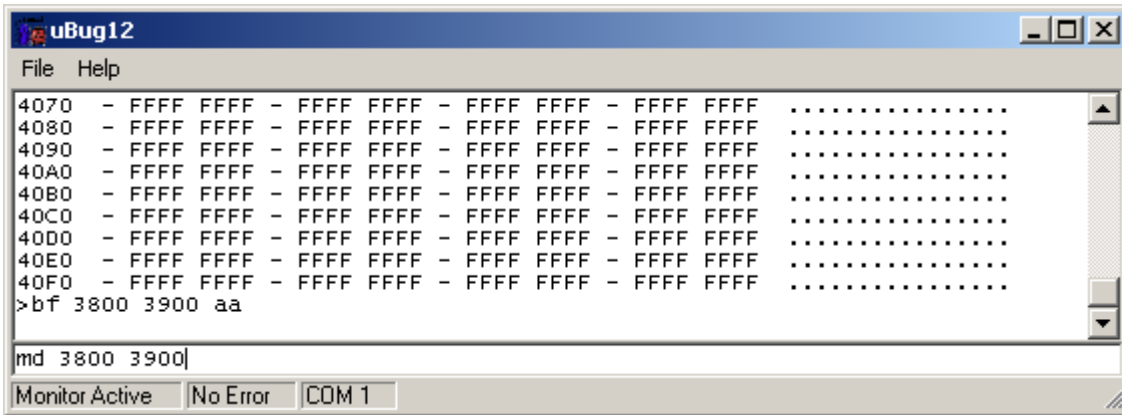


### Memory Fill:

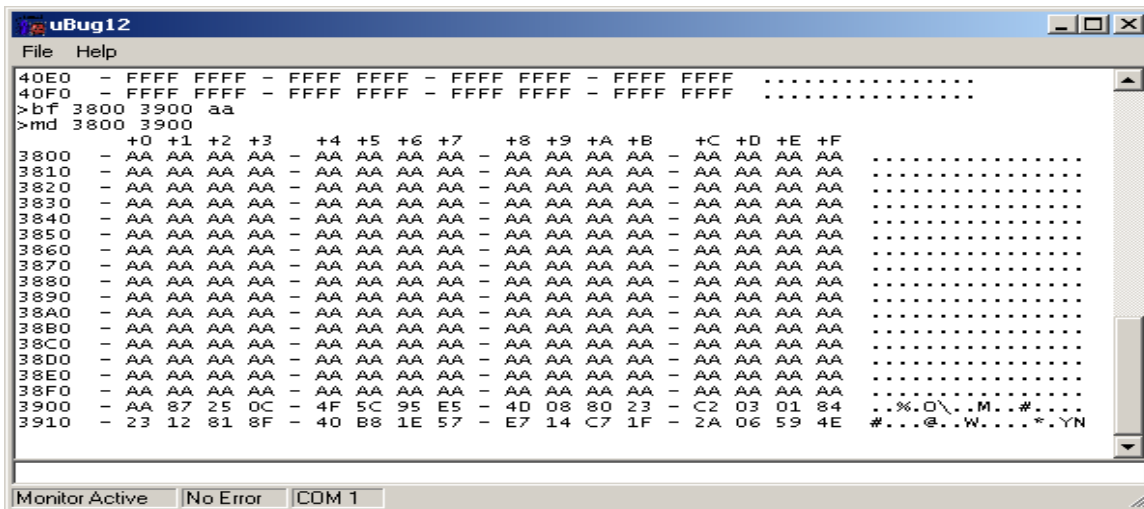
The commands are **BF** and **BFW**. Example of byte fill, type **bf 3800 3900 AA** to fill an address block with the value \$AA



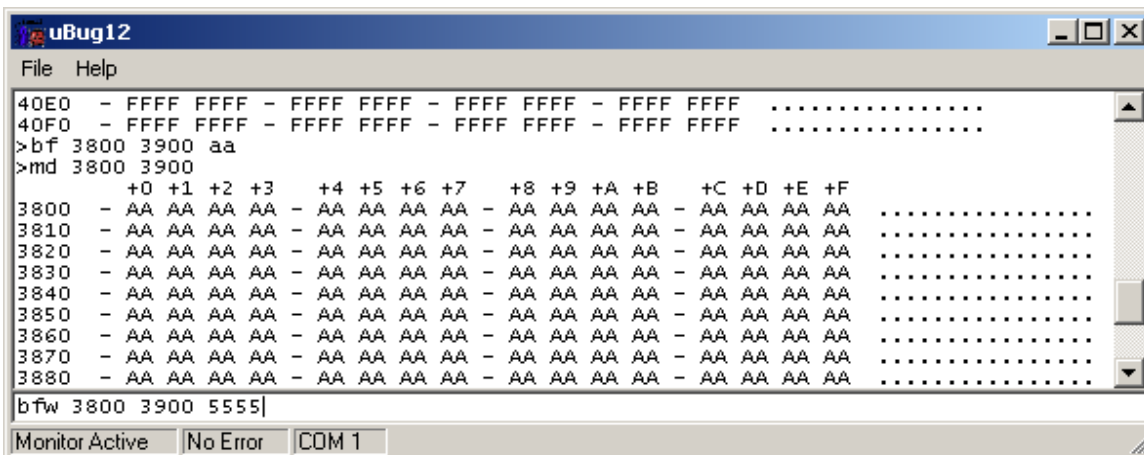
### Verifying the block fill



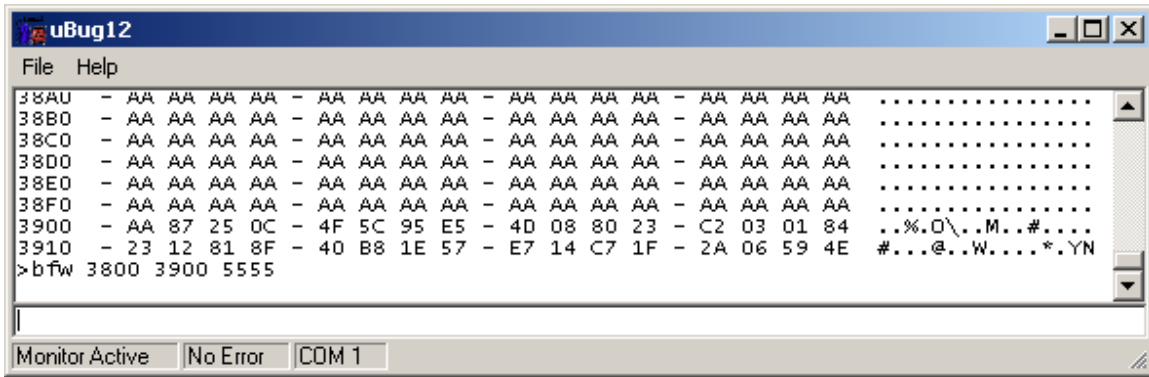
Memory dump of the address block \$3800 to \$3900



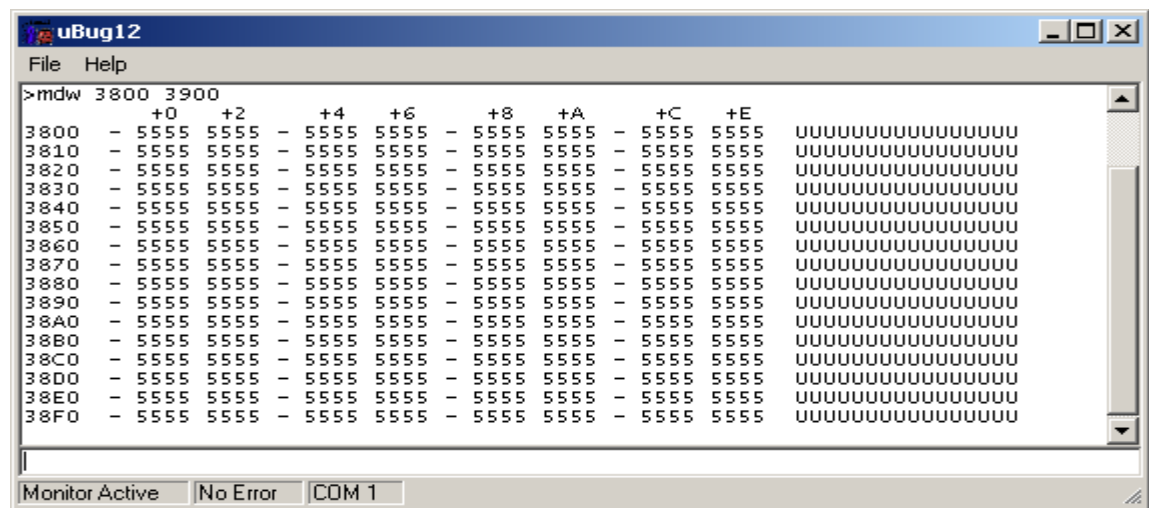
Example of word fill, type **bfw 3800 3900 5555** to fill an address block with the value \$5555





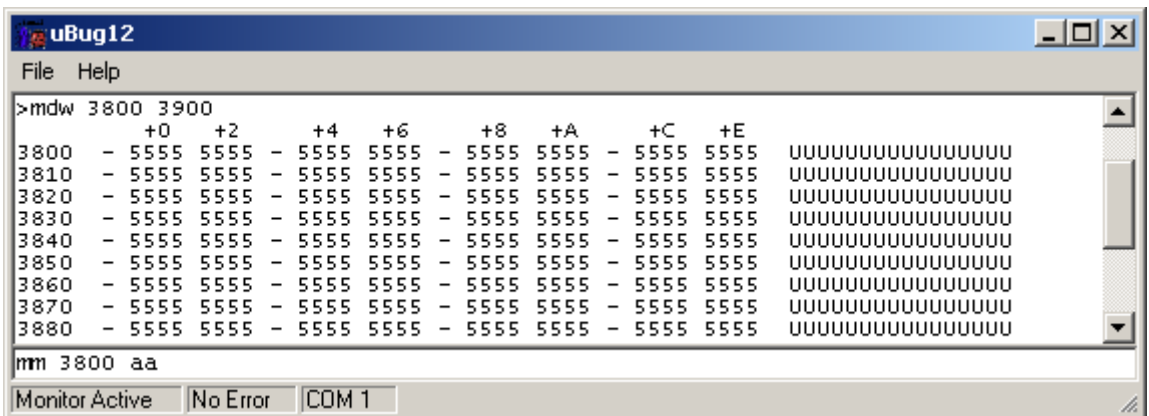


Memory dump of the address block \$3800 to \$3900



Memory Modify:

The commands are **MM** and **MMW**. Example of byte modify, type **mm 3800 AA** to modify the value at address memory \$3800 with \$AA.



```

uBug12
File Help
3870 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
3880 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
3890 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
38A0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
38B0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
38C0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
38D0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
38E0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
38F0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
>nm 3800 aa

```

Verifying the memory modify. Note the byte change at \$3800.

```

uBug12
File Help
38B0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
38C0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
38D0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
38E0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
38F0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
>nm 3800 aa
>md 3800
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
3800 - AA 55 55 55 - 55 55 55 55 - 55 55 55 55 - 55 55 55 55 .UUUUUUUUUUUUUUUUUU

```

Example of word modify, type **mmw 3800 FFFF** to modify the value at address memory \$3800 with \$FFFF.

```

uBug12
File Help
38C0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
38D0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
38E0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
38F0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555 UUUUUUUUUUUUUUUUUUU
>nm 3800 aa
>md 3800
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
3800 - AA 55 55 55 - 55 55 55 55 - 55 55 55 55 - 55 55 55 55 .UUUUUUUUUUUUUUUUUU
>mmw 3800 ffff

```

Verifying the memory modify. Note the word change at \$3800.

```

uBug12
File Help
38F0 - 5555 5555 - 5555 5555 - 5555 5555 - 5555 5555  UUUUUUUUUUUUUUUUUUU
>mm 3800 aa
>md 3800
      +0 +1 +2 +3   +4 +5 +6 +7   +8 +9 +A +B   +C +D +E +F
3800 - AA 55 55 55 - 55 55 55 55 - 55 55 55 55 - 55 55 55 55  .UUUUUUUUUUUUUUUUUU
>mmw 3800 ffff
>mdw 3800
      +0  +2   +4  +6   +8  +A   +C  +E
3800 - FFFF 5555 - 5555 5555 - 5555 5555 - 5555 5555  ..UUUUUUUUUUUUUUUU

```

**Registers:**

There are 9 commands to modify the different registers.

- RD** - Register Display
- RM** <RegisterName> <Data8/16> - Register Modify
- CCR** <Data8> - Set CCR register
- D** <Data16> - Set D register
- PC** <Data16> - Set PC register
- PP** <Data8> - Set PP register
- SP** <Data16> - Set SP register
- X** <Data16> - Set X register
- Y** <Data16> - Set Y register

**Data8** – 8 bit value  
**Data16** – 16 bit value

The **RD** command shows the content of PPAGE (**PP**), Program Counter (**PC**), Stack Pointer (**SP**), Conditional Code (**CCR**) **X**, **Y**, and **D** registers.

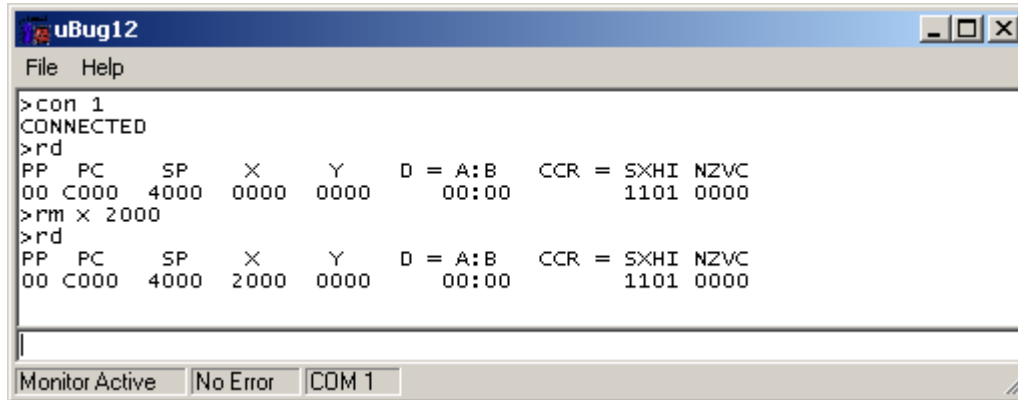
```

uBug12
File Help
>CON 1
CONNECTED
>rd
PP  PC   SP   X   Y   D = A:B   CCR = SXHI NZVC
00 C000 4000 0000 0000 00:00      1101 0000

```

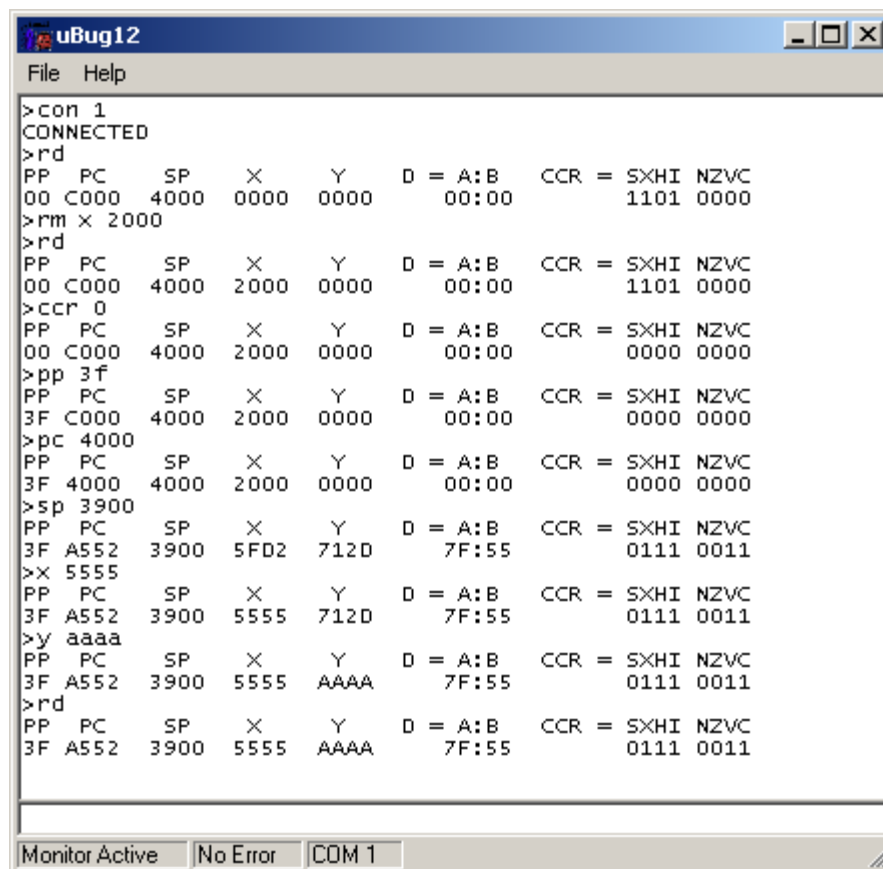
The **RM** command will allow the registers to be changed. For example, change Register X content with value 2000. Type the command **RM X 2000**.

Note that Register X has changed from \$0000 to \$2000



```
uBug12
File Help
>con 1
CONNECTED
>rd
PP PC SP X Y D = A:B CCR = SXHI NZVC
00 C000 4000 0000 0000 00:00 1101 0000
>rm X 2000
>rd
PP PC SP X Y D = A:B CCR = SXHI NZVC
00 C000 4000 2000 0000 00:00 1101 0000
Monitor Active No Error COM 1
```

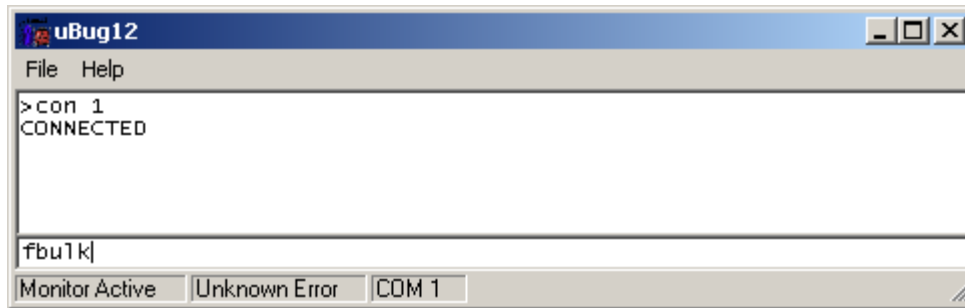
Notice the registers as each one are changed.



```
uBug12
File Help
>con 1
CONNECTED
>rd
PP PC SP X Y D = A:B CCR = SXHI NZVC
00 C000 4000 0000 0000 00:00 1101 0000
>rm X 2000
>rd
PP PC SP X Y D = A:B CCR = SXHI NZVC
00 C000 4000 2000 0000 00:00 1101 0000
>ccr 0
PP PC SP X Y D = A:B CCR = SXHI NZVC
00 C000 4000 2000 0000 00:00 0000 0000
>pp 3f
PP PC SP X Y D = A:B CCR = SXHI NZVC
3F C000 4000 2000 0000 00:00 0000 0000
>pc 4000
PP PC SP X Y D = A:B CCR = SXHI NZVC
3F 4000 4000 2000 0000 00:00 0000 0000
>sp 3900
PP PC SP X Y D = A:B CCR = SXHI NZVC
3F A552 3900 5FD2 712D 7F:55 0111 0011
>x 5555
PP PC SP X Y D = A:B CCR = SXHI NZVC
3F A552 3900 5555 712D 7F:55 0111 0011
>y aaaa
PP PC SP X Y D = A:B CCR = SXHI NZVC
3F A552 3900 5555 AAAA 7F:55 0111 0011
>rd
PP PC SP X Y D = A:B CCR = SXHI NZVC
3F A552 3900 5555 AAAA 7F:55 0111 0011
Monitor Active No Error COM 1
```

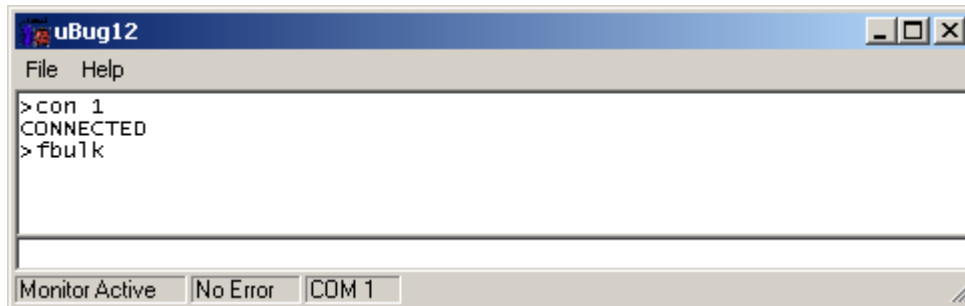
## Flash erase and programming:

To erase the Flash memory the command is **FBULK**.



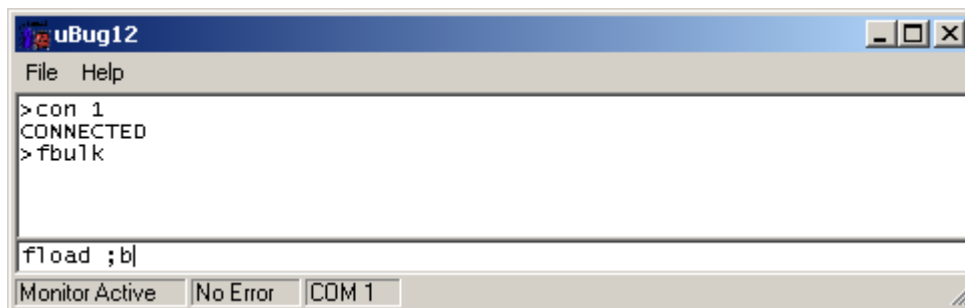
```
uBug12
File Help
> con 1
CONNECTED
fbulk
Monitor Active Unknown Error COM 1
```

Successful erase



```
uBug12
File Help
> con 1
CONNECTED
> fbulk
Monitor Active No Error COM 1
```

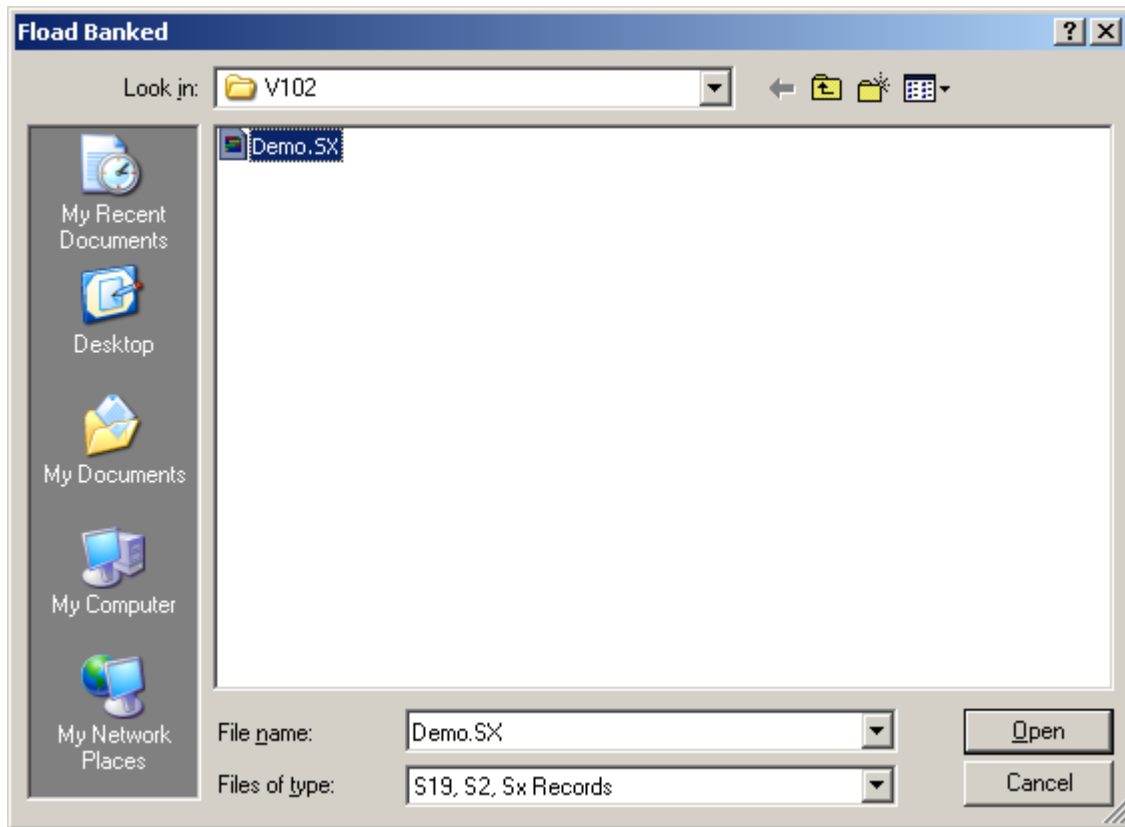
To program FLASH the command is **Fload ;b** for banked S19, SX, S2 records. For non banked S2 or formatted S19 (went thru SrecCVT) record the command is **Fload**.



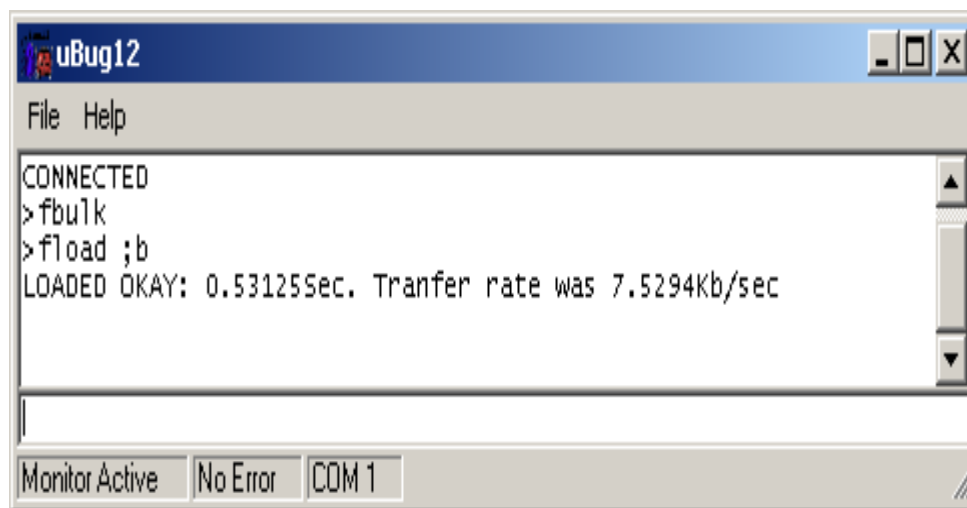
```
uBug12
File Help
> con 1
CONNECTED
> fbulk
fload ;b
Monitor Active No Error COM 1
```

Once the Fload ;b command is invoked, uBUG12 will open an explorer window to help and locate the S-record. In this example, demo.sx will be the target S-record file.

Double click on the file to initiate upload.



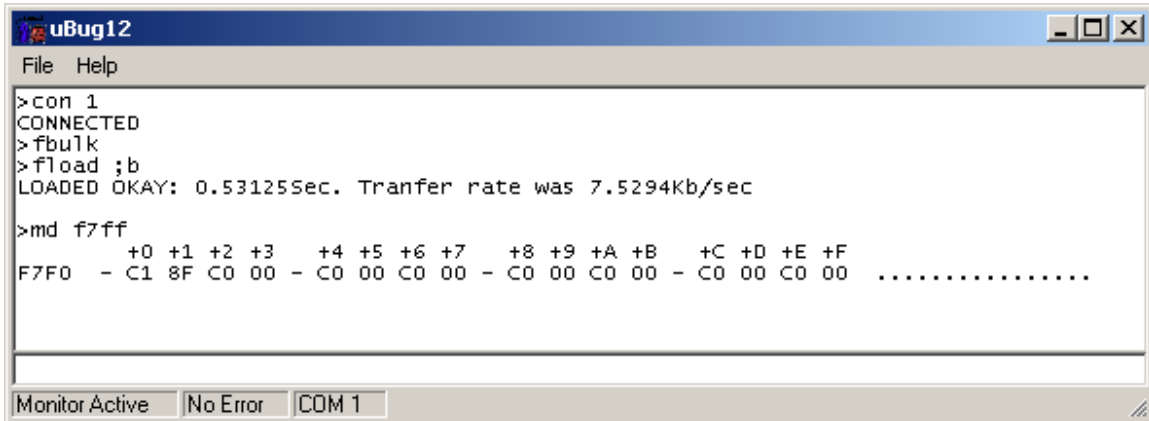
Successful programming of Demo.sx



Note that the Serial Monitor resides at \$F800 - \$FFFF. Therefore uBUG12 will automatically re-locate the vector addresses at below \$F800.

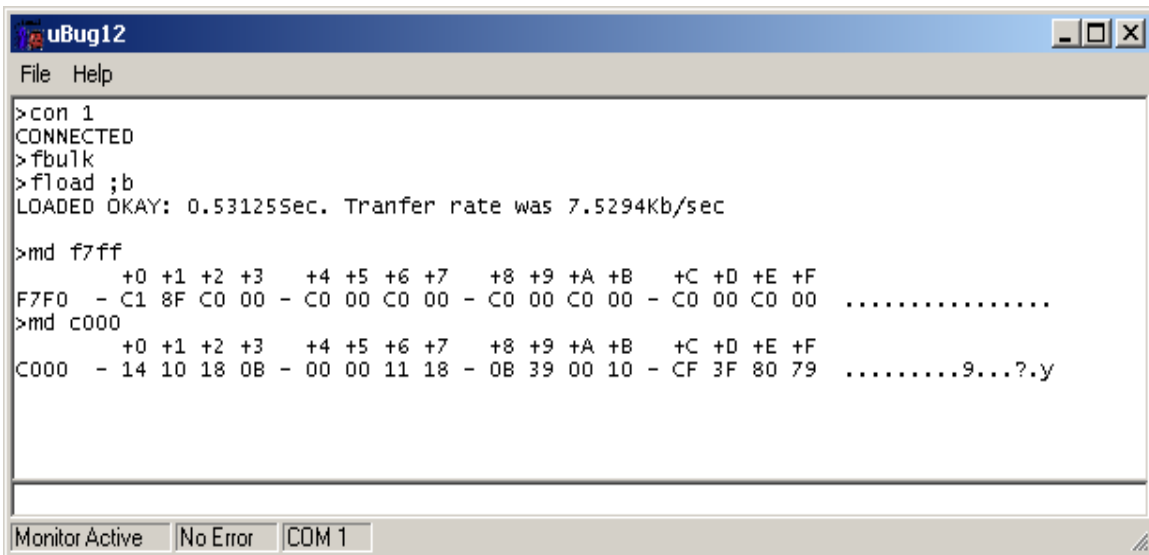
Briefly look at the Pseudo Vector address to check where the start of the program. The command is **md f7ff** to show a memory dump of the Pseudo Vector address at power up or reset.

The power up reset value at \$F7FE is \$C000. Therefore the program will start at \$C000



```
uBug12
File Help
>con 1
CONNECTED
>fbulk
>fload ;b
LOADED OKAY: 0.53125Sec. Transfer rate was 7.5294Kb/sec
>md f7ff
  +0 +1 +2 +3   +4 +5 +6 +7   +8 +9 +A +B   +C +D +E +F
F7F0 - C1 8F C0 00 - C0 00 C0 00 - C0 00 C0 00 - C0 00 C0 00 .....
Monitor Active | No Error | COM 1
```

Memory dump at \$C000



```
uBug12
File Help
>con 1
CONNECTED
>fbulk
>fload ;b
LOADED OKAY: 0.53125Sec. Transfer rate was 7.5294Kb/sec
>md f7ff
  +0 +1 +2 +3   +4 +5 +6 +7   +8 +9 +A +B   +C +D +E +F
F7F0 - C1 8F C0 00 - C0 00 C0 00 - C0 00 C0 00 - C0 00 C0 00 .....
>md c000
  +0 +1 +2 +3   +4 +5 +6 +7   +8 +9 +A +B   +C +D +E +F
C000 - 14 10 18 0B - 00 00 11 18 - 0B 39 00 10 - CF 3F 80 79 .....9...?.y
Monitor Active | No Error | COM 1
```

One can see that there are Data at \$C000. To execute the program using uBUG12, several registers needs to be initialized. Firstly, look at the registers by the **RD** command then invoke the **RESET** command to initialize the registers.

Memory dumps of the registers before **RESET** command is invoked.

```

uBug12
File Help
>con 1
CONNECTED
>fbulk
>fload ;b
LOADED OKAY: 0.53125Sec. Transfer rate was 7.5294Kb/sec

>md f7ff
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
F7F0 - C1 8F C0 00 - C0 00 C0 00 - C0 00 C0 00 - C0 00 C0 00 .....
>md c000
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
C000 - 14 10 18 0B - 00 00 11 18 - 0B 39 00 10 - CF 3F 80 79 .....9...?.y
>r
PP PC SP X Y D = A:B CCR = SXHI NZVC
3F A552 3900 5555 AAAA 7F:55 0111 0011

```

After **RESET** command please note the changes with various registers.

```

uBug12
File Help
LOADED OKAY: 0.53125Sec. Transfer rate was 7.5294Kb/sec

>md f7ff
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
F7F0 - C1 8F C0 00 - C0 00 C0 00 - C0 00 C0 00 - C0 00 C0 00 .....
>md c000
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
C000 - 14 10 18 0B - 00 00 11 18 - 0B 39 00 10 - CF 3F 80 79 .....9...?.y
>r
PP PC SP X Y D = A:B CCR = SXHI NZVC
3F A552 3900 5555 AAAA 7F:55 0111 0011
>reset
>r
PP PC SP X Y D = A:B CCR = SXHI NZVC
00 C000 4000 0000 0000 00:00 1101 0000

```

To execute program the command is simply type **go** (after RESET is invoked) or **go C000**.

```

uBug12
File Help
>md f7ff
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
F7F0 - C1 8F C0 00 - C0 00 C0 00 - C0 00 C0 00 - C0 00 C0 00 .....
>md c000
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
C000 - 14 10 18 0B - 00 00 11 18 - 0B 39 00 10 - CF 3F 80 79 .....9...?.y
>r
PP PC SP X Y D = A:B CCR = SXHI NZVC
3F A552 3900 5555 AAAA 7F:55 0111 0011
>reset
>r
PP PC SP X Y D = A:B CCR = SXHI NZVC
00 C000 4000 0000 0000 00:00 1101 0000
>go

```



The BR, TRACE commands are not to be use and will at some point be fully implemented.

This concludes the uBUG12. Stay tune for further development.

Lastly, the **op** command will change the look of the GUI. One should try to see what it does as it cannot be shown in this document.