

Learn Assembly Programming With ChibiAkumas!



Retro GameDev &
Want to support this content?
Back ChibiAkumas on Patreon!

[View Options](#)
[Default Dark](#)
[Simple \(Hide this menu\)](#)
[Print Mode \(white background\)](#)

65c02 Assembly programming for the Apple IIe

The Apple II saw many generations of it's hardware, from the early Apple II to the final Apple IIgs the hardware remained mostly compatible, however the hardware was heavily upgraded, moving from a 4k 6502, to a 8mb 16 bit 65816




The most curious thing from our point of view is that the Apple II used 3 generations of the 6502, and it's the Apple IIe with its enhanced 8 bit 65C02 we'll cover in these tutorials



[Top Menu](#)
[Main Menu](#)
[Youtube channel](#)
[Forum](#)
[AkuSprite Editor](#)
[Dec/Bin/Hex/Oct/Ascii Table](#)

There are many versions of the Apple 2, we're only going to cover the Apple IIe in these tutorials.

	Apple II	Apple IIe	Apple IIgs
Cpu	1mhz 6502	1mhz 65C02	2mhz 65C816 (16 bit)
Ram	4k	48k	up to 8mb
Resolution	Text Only	280x192	320x200
Sound	beeper	beeper	Ensonique 5503

[Z80 Content](#)
[Z80 Tutorial List](#)
[Learn Z80 Assembly](#) 
[Hello World](#)
[Advanced Series](#)
[Multiplatform Series](#)
[Platform Specific Series](#)
[ChibiAkumas Series](#) 
[Grime Z80](#) 
[Z80 Downloads](#)
[Z80 Cheatsheet](#)
[Sources.7z](#)
[DevTools kit](#)



Z80 Platforms

- [Amstrad CPC](#)
- [Elan Enterprise](#)
- [Gameboy & Gameboy Color](#)
- [Master System & GameGear](#)
- [MSX & MSX2](#)
- [Sam Coupe](#)
- [TI-83](#)
- [ZX Spectrum](#)
- [Spectrum NEXT](#)
- [Computers Lynx](#)

6502 Content

*** [6502 Tutorial List](#) ***

- [Learn 6502 Assembly](#)
- [Advanced Series](#)
- [Platform Specific Series](#)
- [Hello World Series](#)
- [Grime 6502](#)
- [6502 Downloads](#)
- [6502 Cheatsheet](#)
- [Sources.7z](#)
- [DevTools kit](#)
- [6502 Platforms](#)
- [Apple IIe](#)
- [Atari 800 and 5200](#)
- [Atari Lynx](#)
- [BBC Micro](#)
- [Commodore 64](#)
- [Commander x16](#)
- [Super Nintendo \(SNES\)](#)
- [Nintendo NES / Famicom](#)
- [PC Engine \(TurboGrafx-16\)](#)
- [Vic 20](#)

68000 Content

ChibiAkumas Tutorials

Lesson P3 - Bitmap Functions on the Apple II
Lesson P12 - Joystick Reading on the Apple II
Lesson P23 (Z80) - Sound with the 'Beeper' on the Apple II

Apple IIe Memory Map

The Apple IIc memory map is pretty typical.

We'll use Graphics Mode, and Page 2 - this means for our purposes the area \$0C00-\$3FFF can be used for our main program code.

Notice that the area \$C000-\$FFFF allows us to access the hardware...

Each "Port" has a different purpose, but rather strangely when we want to do something like set the graphics mode, we write ANY value to the graphics port... the value makes no difference!

From	To	Purpose
\$0000	\$00FF	Zero page
\$0100	\$01FF	Stack
\$0200	\$02FF	GETLN buffer
\$0300	\$03CF	Free Space
\$03D0	\$03FF	DOS & Interrupt vectors
\$0400	\$07FF	Text

		Screen Page 1
\$0800	\$0BFF	Text Screen Page 2
\$0C00	\$1FFF	Free space
\$2000	\$3FFF	Graphics Screen Page 1
\$4000	\$5FFF	Graphics Screen Page 2
\$6000	\$95FF	Applesoft String Data
\$9600	\$BFFF	Operating System Memory
\$C000	\$FFFF	System Hardware ports

[68000 Tutorial List](#)
[Learn 68000 Assembly](#) ▶
[Hello World Series](#)
[Platform Specific Series](#)
[Grime 68000](#) ▶
[68000 Downloads](#)
[68000 Cheatsheet](#)
[Sources.7z](#)
[DevTools kit](#)
[68000 Platforms](#)
[Amiga 500](#) ▶
[Atari ST](#) ▶
[Neo Geo](#) ▶
[Sega Genesis / Mega Drive](#) ▶
[Sinclair QL](#) ▶
[X68000 \(Sharp x68k\)](#) ▶

[8086 Content](#)
[Learn 8086 Assembly](#) ▶
[Platform Specific Series](#)
[Hello World Series](#)
[8086 Downloads](#)
[8086 Cheatsheet](#)
[Sources.7z](#)
[DevTools kit](#)
[8086 Platforms](#)
[Wonderswan](#)
[MsDos](#)

[ARM Content](#)
[Learn ARM Assembly](#) ▶
[Platform Specific Series](#)
[ARM Downloads](#)
[ARM Cheatsheet](#)
[Sources.7z](#)
[DevTools kit](#)

Hardware Ports Memory Map

Writing any value to these memory addresses causes the hardware change.

For example, to change the system to graphics mode:

```
Ida #0
sta $C050 ; Text off
sta $C052 ; Mixed Mode off
sta $c057 ; Display hires
sta $C055 ; Hires screen 2
```

Reading from the ports will also have the same effect!

Address	Code	Details
C050	TXTCLR	Display Graphics
C051	TXTSET	Display Text
C052	MIXCLR	Display Full Screen
C053	MIXSET	Display Split Screen
C054	TXTPAGE1	Display Page 1
C055	TXTPAGE2	Display Page 2
C056	LORES	Display LoRes Graphics
C057	HIRES	Display HiRes Graphics

Highres Screen - Screen Colors

Colors on the Apple II are effectively an 'Artifact' of the screen...

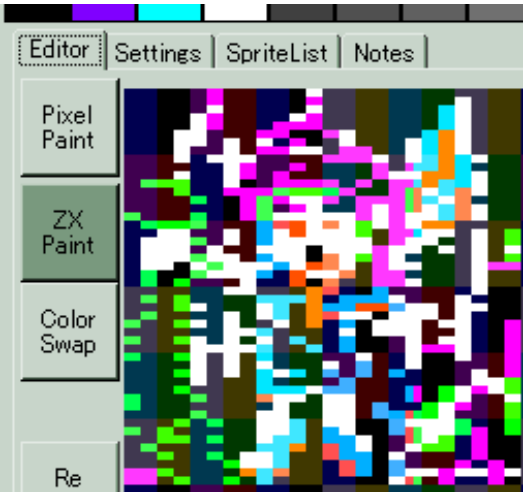
certain combinations of Off (0) and On (1) pixels will appear colored... this is known as Composite Artifact colors...

Unlike pretty much every system in existence, 8 bits of a byte draw 7 pixels!.... the top bit is a 'Color bit'... selecting 'Palette 0 or 1

The remaining 7 bits are the 7 pixels of bitmap data... because each line is 40 bytes wide, the Apple II screen is a rather odd resolution of 280x192

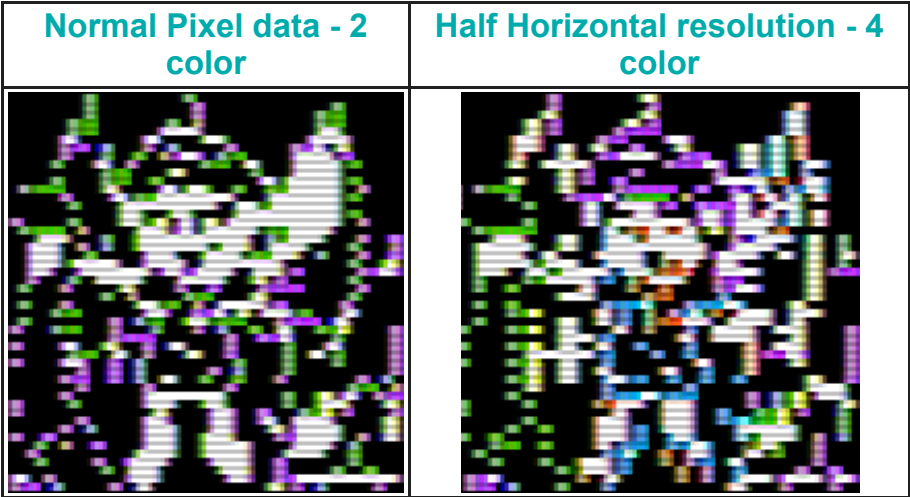
Because of these artifacts, a '2 color' bitmap will show colors depending on the combination of the pixels...

My Akusprite editor offers a half horizontal resolution mode, where the 4 colors will be converted to the correct bit combinations



Bitnum	7	6	5	4	3	2	1	0
Function	Color	Pixel 1	Pixel 2	Pixel 3	Pixel 4	Pixel 5	Pixel 6	Pixel 7

	Pixel Pair			
Color Bit	00	01	10	11
0	00	01	10	11
1	00	01	10	11



ARM Platforms
[Gameboy Advance](#)
[Nintendo DS](#)
[Risc Os](#)

Risc-V Content
[Learn Risc-V Assembly](#)
[Risc-V Downloads](#)
[Risc-V Cheatsheet](#)
[Sources.7z](#)
[DevTools kit](#)

PDP-11 Content
[Learn PDP-11 Assembly](#)
[PDP-11 Downloads](#)
[PDP-11 Cheatsheet](#)
[Sources.7z](#)
[DevTools kit](#)

TMS9900 Content
[Learn TMS9900 Assembly](#)
[TMS9900 Downloads](#)
[TMS9900 Cheatsheet](#)
[Sources.7z](#)
[DevTools kit](#)
[TMS9900 Platforms](#)
[Ti 99](#)

6809 Content
[Learn 6809 Assembly](#)
[6809 Downloads](#)
[6809/6309 Cheatsheet](#)
[Sources.7z](#)
[DevTools kit](#)
[6809 Platforms](#)

[Dragon 32/Tandy Coco](#)
[Fujitsu FM7](#)
[TRS-80 Coco 3](#)
[Vectrex](#)

My Game projects
[Chibi Aliens](#)
[Chibi Akumas](#)

Work in Progress
[Learn 65816 Assembly](#)
[Learn eZ80 Assembly](#)

Misc bits
[Ruby programming](#)

[Buy my Assembly programming book on Amazon in Print or Kindle!](#)



Highres Screen Mode 2 - Memory map

Memory addresses for Screen Mode 2 is split into 3 chunks,also, every 8 lines we effectively 'reset' our high memory address and add \$80

1st Third

Lines 0-63

2nd Third

Lines 64-127

3rd Third

Lines 128-191

Pixels in Each line are in normal Left->Right format, however remeber 7 pixels are defined by each byte, with 1 bit defining the color palette.

We can calculate the address of the start of a line by splitting the bits of the Y line number...

YPOS:

7	6	5	4	3	2	1	0
A	A	B	B	B	C	C	C

Address= Base+(AA*\$0028) + (BBB*\$0080) + (CC*\$0400) + XPOS

	1st Third Lines 0-63	2nd Third Lines 64-127	3rd Third Lines 128-191	Unused
0	\$4000-\$4027	\$4028-\$404F	\$4050-\$4077	\$4078-\$407F
1	\$4400-\$4427	\$4428-\$444F	\$4450-\$4477	\$4478-\$447F
2	\$4800-\$4827	\$4828-\$484F	\$4850-\$4877	\$4878-\$487F
3	\$4C00-\$4C27	\$4C28-\$4C4F	\$4C50-\$4C77	\$4C78-\$4C7F
4	\$5000-\$5027	\$5028-\$504F	\$5050-\$5077	\$5078-\$507F
5	\$5400-\$5427	\$5428-\$544F	\$5450-\$5477	\$5478-\$547F
6	\$5800-\$5827	\$5828-\$584F	\$5850-\$5877	\$5878-\$587F
7	\$5C00-\$5C27	\$5C28-\$5C4F	\$5C50-\$5C77	\$5C78-\$5C7F
8	\$4080-\$40A7	\$40A8-\$40CF	\$40D0-\$40F7	\$40F8-\$40FF
9	\$4480-\$44A7	\$44A8-\$44CF	\$44D0-\$44F7	\$44F8-\$44FF
10	\$4880-\$48A7	\$48A8-\$48CF	\$48D0-\$48F7	\$48F8-\$48FF
11	\$4C80-\$4CA7	\$4CA8-\$4CCF	\$4CD0-\$4CF7	\$4CF8-\$4CFF
12	\$5080-\$50A7	\$50A8-\$50CF	\$50D0-\$50F7	\$50F8-\$50FF
13	\$5480-\$54A7	\$54A8-\$54CF	\$54D0-\$54F7	\$54F8-\$54FF
14	\$5880-\$58A7	\$58A8-\$58CF	\$58D0-\$58F7	\$58F8-\$58FF
15	\$5C80-\$5CA7	\$5CA8-\$5CCF	\$5CD0-\$5CF7	\$5CF8-\$5CFF
16	\$4100-\$4127	\$4128-\$414F	\$4150-\$4177	\$4178-\$417F
17	\$4500-\$4527	\$4528-\$454F	\$4550-\$4577	\$4578-\$457F
18	\$4900-\$4927	\$4928-\$494F	\$4950-\$4977	\$4978-\$497F
19	\$4D00-\$4D27	\$4D28-\$4D4F	\$4D50-\$4D77	\$4D78-\$4D7F
20	\$5100-\$5127	\$5128-\$514F	\$5150-\$5177	\$5178-\$517F
21	\$5500-\$5527	\$5528-\$554F	\$5550-\$5577	\$5578-\$557F
22	\$5900-\$5927	\$5928-\$594F	\$5950-\$5977	\$5978-\$597F
23	\$5D00-\$5D27	\$5D28-\$5D4F	\$5D50-\$5D77	\$5D78-\$5D7F
24	\$4180-\$41A7	\$41A8-\$41CF	\$41D0-\$41F7	\$41F8-\$41FF
25	\$4580-\$45A7	\$45A8-\$45CF	\$45D0-\$45F7	\$45F8-\$45FF
26	\$4980-\$49A7	\$49A8-\$49CF	\$49D0-\$49F7	\$49F8-\$49FF
27	\$4D80-\$4DA7	\$4DA8-\$4DCF	\$4DD0-\$4DF7	\$4DF8-\$4DFF
28	\$5180-\$51A7	\$51A8-\$51CF	\$51D0-\$51F7	\$51F8-\$51FF
29	\$5580-\$55A7	\$55A8-\$55CF	\$55D0-\$55F7	\$55F8-\$55FF

30	\$5980-\$59A7	\$59A8-\$59CF	\$59D0-\$59F7	\$59F8-\$59FF
31	\$5D80-\$5DA7	\$5DA8-\$5DCF	\$5DD0-\$5DF7	\$5DF8-\$5DFF
32	\$4200-\$4227	\$4228-\$424F	\$4250-\$4277	\$4278-\$427F
33	\$4600-\$4627	\$4628-\$464F	\$4650-\$4677	\$4678-\$467F
34	\$4A00-\$4A27	\$4A28-\$4A4F	\$4A50-\$4A77	\$4A78-\$4A7F
35	\$4E00-\$4E27	\$4E28-\$4E4F	\$4E50-\$4E77	\$4E78-\$4E7F
36	\$5200-\$5227	\$5228-\$524F	\$5250-\$5277	\$5278-\$527F
37	\$5600-\$5627	\$5628-\$564F	\$5650-\$5677	\$5678-\$567F
38	\$5A00-\$5A27	\$5A28-\$5A4F	\$5A50-\$5A77	\$5A78-\$5A7F
39	\$5E00-\$5E27	\$5E28-\$5E4F	\$5E50-\$5E77	\$5E78-\$5E7F
40	\$4280-\$42A7	\$42A8-\$42CF	\$42D0-\$42F7	\$42F8-\$42FF
41	\$4680-\$46A7	\$46A8-\$46CF	\$46D0-\$46F7	\$46F8-\$46FF
42	\$4A80-\$4AA7	\$4AA8-\$4ACF	\$4AD0-\$4AF7	\$4AF8-\$4AFF
43	\$4E80-\$4EA7	\$4EA8-\$4ECF	\$4ED0-\$4EF7	\$4EF8-\$4EFF
44	\$5280-\$52A7	\$52A8-\$52CF	\$52D0-\$52F7	\$52F8-\$52FF
45	\$5680-\$56A7	\$56A8-\$56CF	\$56D0-\$56F7	\$56F8-\$56FF
46	\$5A80-\$5AA7	\$5AA8-\$5ACF	\$5AD0-\$5AF7	\$5AF8-\$5AFF
47	\$5E80-\$5EA7	\$5EA8-\$5ECF	\$5ED0-\$5EF7	\$5EF8-\$5EFF
48	\$4300-\$4327	\$4328-\$434F	\$4350-\$4377	\$4378-\$437F
49	\$4700-\$4727	\$4728-\$474F	\$4750-\$4777	\$4778-\$477F
50	\$4B00-\$4B27	\$4B28-\$4B4F	\$4B50-\$4B77	\$4B78-\$4B7F
51	\$4F00-\$4F27	\$4F28-\$4F4F	\$4F50-\$4F77	\$4F78-\$4F7F
52	\$5300-\$5327	\$5328-\$534F	\$5350-\$5377	\$5378-\$537F
53	\$5700-\$5727	\$5728-\$574F	\$5750-\$5777	\$5778-\$577F
54	\$5B00-\$5B27	\$5B28-\$5B4F	\$5B50-\$5B77	\$5B78-\$5B7F
55	\$5F00-\$5F27	\$5F28-\$5F4F	\$5F50-\$5F77	\$5F78-\$5F7F
56	\$4380-\$43A7	\$43A8-\$43CF	\$43D0-\$43F7	\$43F8-\$43FF
57	\$4780-\$47A7	\$47A8-\$47CF	\$47D0-\$47F7	\$47F8-\$47FF
58	\$4B80-\$4BA7	\$4BA8-\$4BCF	\$4BD0-\$4BF7	\$4BF8-\$4BFF
59	\$4F80-\$4FA7	\$4FA8-\$4FCF	\$4FD0-\$4FF7	\$4FF8-\$4FFF
60	\$5380-\$53A7	\$53A8-\$53CF	\$53D0-\$53F7	\$53F8-\$53FF
61	\$5780-\$57A7	\$57A8-\$57CF	\$57D0-\$57F7	\$57F8-\$57FF
62	\$5B80-\$5BA7	\$5BA8-\$5BCF	\$5BD0-\$5BF7	\$5BF8-\$5BFF
63	\$5F80-\$5FA7	\$5FA8-\$5FCF	\$5FD0-\$5FF7	\$5FF8-\$5FFF

Available worldwide!
 Search 'ChibiAkumas' on
 your local Amazon website!
[Click here for more info!](#)

Want to help support
 my content creation?

 BECOME A PATRON

Want to help support
 my content creation?

 SUBSCRIBESTAR



ASM Tutorials for
280,6502,68000
8086,ARM and
more On my
Youtube Channel



Questions,
Suggestions
Advice?
Discuss on the
Forums!



Want to help support
my content creation?



SUBSCRIBESTAR

Recent New Content

[Amiga - ASM PSET and POINT
for Pixel Plotting](#)

[Learn 65816 Assembly: 8 and 16
bit modes on the 65816](#)

[SNES - ASM PSET and POINT
for Pixel Plotting](#)

[ARM Assembly Lesson H3](#)

[Lesson P65 - Mouse reading on
the Sam Coupe](#)

[Mouse Reading in MS-DOS](#)

[Risc-V Assembly Lesson 3 - Bit](#)

[ops and more maths!](#)

[Mouse reading on the MSX](#)

[Hello World on RISC-OS](#)

[Atari 800 / 5200 - ASM PSET and
POINT for Pixel Plotting](#)

[Apple 2 - ASM PSET and POINT
for Pixel Plotting](#)

[Making a 6502 ASM Tron game...](#)
[Photon1 - Introduction and Data
Structures](#)

Gaming + more:

[Emily The Strange \(DS\) - Live
full playthrough](#)

[\\$150 calculator: Unboxing the
Ti-84 Plus CE \(eZ80 cpu\)](#)



Available worldwide!
Search 'ChibiAkumas' on
your local Amazon website!

[Click here for more info!](#)



Want to help support
my content creation?



SUBSCRIBESTAR



Buy ChibiAkuma
merchandise from
Teespring &
Support my content

ASM Tutorials for
280,6502,68000
8086,ARM and
more On my
Youtube Channel



Questions,
Suggestions
Advice?
Discuss on the
Forums!



Want to help support
my content creation?



SUBSCRIBESTAR

Recent New Content

[Amiga - ASM PSET and POINT
for Pixel Plotting](#)

[Learn 65816 Assembly: 8 and 16](#)

[bit modes on the 65816](#)

[SNES - ASM PSET and POINT
for Pixel Plotting](#)

[ARM Assembly Lesson H3](#)

[Lesson P65 - Mouse reading on
the Sam Coupe](#)

[Mouse Reading in MS-DOS](#)

[Risc-V Assembly Lesson 3 - Bit
ops and more maths!](#)

[Mouse reading on the MSX](#)

[Hello World on RISC-OS](#)

[Atari 800 / 5200 - ASM PSET and
POINT for Pixel Plotting](#)

[Apple 2 - ASM PSET and POINT
for Pixel Plotting](#)

[Making a 6502 ASM Tron game...](#)
[Photon1 - Introduction and Data
Structures](#)

Gaming + more:

[Emily The Strange \(DS\) - Live
full playthrough](#)

[\\$150 calculator: Unboxing the
Ti-84 Plus CE \(eZ80 cpu\)](#)

[Buy my Assembly programming book
on Amazon in Print or Kindle!](#)



[Available worldwide!
Search 'ChibiAkumas' on
your local Amazon website!](#)

[Click here for more info!](#)

Want to help support
my content creation?

 **BECOME A PATRON**

Want to help support
my content creation?

 **SUBSCRIBESTAR**

Buy ChibiAkumas
merchandise from
Teespring &
Support my content



ASM Tutorials for
280,6502,68000
8086,ARM and
more On my
Youtube Channel



Questions,
Suggestions
Advice?
Discuss on the
Forums!



Want to help support
my content creation?



SUBSCRIBESTAR

Recent New Content

[Amiga - ASM PSET and POINT
for Pixel Plotting](#)

[Learn 65816 Assembly: 8 and 16
bit modes on the 65816](#)

[SNES - ASM PSET and POINT
for Pixel Plotting](#)

[ARM Assembly Lesson H3](#)

[Lesson P65 - Mouse reading on
the Sam Coupe](#)

[Mouse Reading in MS-DOS](#)

[Risc-V Assembly Lesson 3 - Bit
ops and more maths!](#)

[Mouse reading on the MSX](#)

[Hello World on RISC-OS](#)

[Atari 800 / 5200 - ASM PSET and](#)

[POINT for Pixel Plotting](#)

[Apple 2 - ASM PSET and POINT
for Pixel Plotting](#)

[Making a 6502 ASM Tron game...](#)
[Photon1 - Introduction and Data
Structures](#)

Gaming + more:

[Emily The Strange \(DS\) - Live
full playthrough](#)

[\\$150 calculator: Unboxing the
Ti-84 Plus CE \(eZ80 cpu\)](#)

