6502 Assembly programming for the BBC Micro B

The BBC Micro was made by Acorn for the UK's public broadcasting system, it was presented as part of the TV show "BBC Micro Live"

The BBC had highly configurable hardware, with support for many external devices, and even non 6502 coprocessors... it's Basic even supported in-line assembly language!

the Model A was the cheaper model, however it's 16k limit will be rather restrictive, so we will be covering the more usable 32k system the Micro B

	Model A	Model B	Master
Cpu	2mhz 6502	2mhz 6502	2mhz 6502
Ram	16k	32k	128k
Vram	Uses internal		

View Options
Default Dark
Simple (Hide this menu)
Print Mode (white background)

Top Menu
Main Menu
Youtube channel
Forum
AkuSprite Editor
Dec/Bin/Hex/Oct/Ascii Table

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

	memory		
Resolution	640x256	640x256	640x256
Colors	8	8	8
Sound chip	SN76489 (4 channel)	SN76489 (4 channel)	SN76489 (4 channel)

ChibiAkumas Tutorials

Lesson P1 - Bitmap Functions on the BBC
Lesson P10 - Joystick Reading on the BBC
Lesson P17 - Palette definitions on the BBC
Lesson P22 (z80) - Sound with the SN76489 on the BBC Micro

Video Registers

Write the regsiter you want to set to $FE00

then write the new value to $FE01

RegNum	register description	Mode 1 320x256 4 color
$FE20	Screen mode	$D8
0	Horizontal total	$7F
1	Horizontal displayed characters	$50
2	Horizontal sync position	$62
3	Horizontal sync width/Vertical sync time	$28
4	Vertical total	$26

Z80 Platforms
Amstrad CPC
Elan Enterprise
Gameboy & Gameboy Color
Master System & GameGear
MSX & MSX2
Sam Coupe
TI-83
ZX Spectrum
Spectrum NEXT
Camputers 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

5	Vertical total adjust	$00		
6	Vertical displayed characters	$19		
7	Vertical sync position	$22		
8	Interlace/Display delay/Cursor delay	$01		
9	Scan lines per character	$30		
10	Cursor start line and blink type	$00		
11	Cursor end line	$08		
12	Screen start address H (Address /8)	$30		
13	Screen start address L (Address /8)			
14	Cursor position H			
15	Cursor position L			
16	Light pen position			
17	Light pen position			
18	Cursor width (BBFW)			

Physical Colors

the way Screen Bytes map to visible colors is strangely configurable, they are defined by the "Video ULA palette"... which maps nibbles to colors... if we map colors wrong, the same byte may appear a different color depending if it's on odd or even vertical strips

Color Num	EOR	Color
0	7	**Black**
1	6	**Red**
2	5	**Green**
3	4	**Yellow**
4	3	**Blue**
5	2	**Magenta**
6	1	**Cyan**
7	0	**White**

Hardware Addresses

From	To	Purpose	Details

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

$FE00	**$FE07**	6845 CRTC Video controller 18	Set these to change screen mode
$FE08	**$FE0F**	6850 ACIA Serial controller 20.3	
$FE10	**$FE1F**	Serial ULA Serial system chip 20.9	
$FE20	**$FE2F**	Video ULA Video system chip 19	Set these to change screen mode
$FE30	**$FE3F**	74LS161 Paged ROM selector 21	
$FE40	**$FE5F**	6522 VIA SYSTEM VIA 23	Sound & Keyboard
$FE60	**$FE7F**	6522 VIA USER VIA 24	
$FE80	**$FE9F**	8271 FDC Floppy disc controller 25.1	
$FEA0	**$FEBF**	68B54 ADLC ECONET controller 25.2	
$FEC0	**$FEDF**	uPD7002 Analogue to digital converter 26	
$FEE0	**$FEFF**	Tube ULA Tube system interface 27	

Sound Controller - SN76489

The Sound Chip shares a port with the keybord... Before we can send any data to the sound chip, we have to set the port to WRITE... we do this by writing 255 to address $FE43 (we only do this once)

We've covered the sound chip in the **Z80 tutorials here**

Once we've done that, we can write our data to $FE41 in the format below

		Bits							
Command	**Bit Details**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Format Template	L=Latch C=Channel T=Type XXXX=Data	L	C	C	T	D	D	D	D
Tone - Command 1/2	C=Channel L=tone Low data	1	C	C	0	L	L	L	L
Tone - Command 2/2	H= High tone data (Higher numbers = lower tone)	0	-	H	H	H	H	H	H
Volume	C=Channel (0-2) V=Volume (15=silent 0=max)	1	C	C	1	V	V	V	V
Noise Channel	(Channel 3) M=Noise mode (1=white) R=Rate (3=use tone 2)	1	1	1	0	-	M	R	R

Sheila ADC - Analog to Digital Converter (Joystick)

The Joystick is analog on the BBC... we need to read UD and LR, which will return a value from 0-255....

When it comes to reading the Fire, we use $FE40 - part of the sound/keyboard controller!

Port	R/W	Purpose	Bits	Details

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

$FEC0	W	Data Latch / Conversation Start	----MFCC	M=Mode (0=8 bit 1=10 bit)... F=Flag (usually 0)... CC=Channel (0/1 = joy1 2/3=joy2)
$FEC0	R	Status	CBMMm-CC	C=Conversation complete (1=no)...B=busy... M=top two bits of conversiation... m=mode (8/10 bit)... CC=Channel
$FEC1	R	High Data byte	DDDDDDDD	8 Bit Data
$FEC2	R	Low Data byte	DDDD----	extra 4 low bits of 10/12 bit data

Key Reading

Key reading on the BBC is a little weird and rather poorly documented

Essentially you have to select a row (0-8) and Column (0-9), then read in the state of each key one bit at a time!.. these are both read and written to port $**FE4F**

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
K	R	R	R	C	C	C	C

When Written Bits 0-3 (Marked C) select the Column (0-9) and bits 4-6 (Marked R) select the Row (0-8)... Bit 7 has no purpose

When Read Bit 7 returns the Keystate and bits 6-0 have no purpose

An example working piece of code is shown to the right, it's partially based on the disassemblies of the firmware.

You will need a PrintHex command to show the read byte to screen.

```
    LDA #$7F    ;set port A for input on bit 7 others outputs
    STA $FE43
    LDA #$03    ;stop auto scan
    STA $FE40
    ;    This section may not be needed
    ;LDA #$0F    ;select non-existent keyboard column F (0-9 only!)
    ;STA $FE4F   ;
    ;LDA #$01    ;cancel keyboard interrupt
    ;STA $FE4D   ;

    ldy #0
KeyNextLine:
    ldx #8
    tya
KeyNextBit:
    pha
        sta $FE4F
        lda $FE4F
        rol
    pla
```


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!

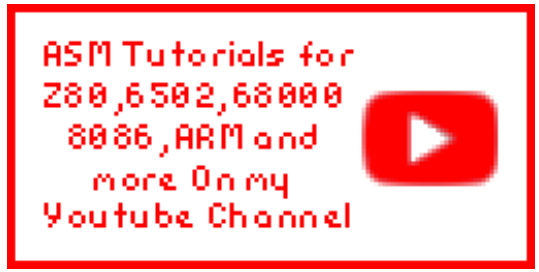
```
    rol z_as
    clc
    adc #%00010000
    dex
    bne KeyNextBit

    lda z_as
    iny
    sta (z_hl),y
    jsr PrintHex
    tya
    cmp #8
    bne KeyNextLine

    LDA #$0B    ;select auto scan
of keyboard
    STA $FE40   ;tell VIA
```

Available worldwide!
Search 'ChibiAkumas' on
your local Amazon website!
Click here for more info!

Useful Links

Advanced User Guide PDF


Buy ChibiAkumas
merchandise from
Teespring &
Support my content


ASM Tutorials for
Z80,6502,68000
8086,ARM and
more On my
Youtube Channel


Questions,
Suggestions
Advice?
Discuss on the
Forums!
phpbb

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 Z80,6502,68000 8086,ARM and more On my Youtube Channel

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

ASM Tutorials for
Z80,6502,68000
8086,ARM and
more On my
Youtube Channel

Questions,
Suggestions
Advice?
Discuss on the
Forums!

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)