Word Juggler IIe v2.9 © 1982-1984, Quark, Inc.

This is a step-by-step guide to crack Word Juggler v2.9. The crack is for the US version of the program. As you own the German one, some addresses may vary and we'll tell where to pay attention and put the right values.

A message from an old man: "Protect your original disk against stupidity, put a sticker on the write notch."

Let's play...

1. Get my world famous copy disk @ http://www.brutaldeluxe.fr/products/apple2/diskcopiers.html and transfer it onto a real floppy thanks to the wonderful ADTPro software by David Schmidt. Then, boot the disk.

LUGU	05/06/90
	presente son
	DISQUE DE COPIEURS
	comprenant
- e	<pre>1- Bit Copy JE+ 4.4C 2- Disk Fixer V4.0 Locksmith 6.3 F-disk Backup 4- Advanced Demuffin 1.4 5- Mobby Disk II 6- Copy JE plus V5.x 7- Disk Muncher V8.0 8- Converteur Rdos V1.01 9- Saut au Dos 3.3 et quelques touches secretes -</pre>
BooT 94	L-BooT 6 by LoGo

THE FIRST MILESTONE IS TO COPY THE ORIGINAL DISK

2. Let's make a copy of the original with Locksmith 6.3 F-disk Backup. Press 3, please.



3. If you have one drive only, press 11 then return. Otherwise, go to 4.



4. Your original disk is in drive 1, a blank 5.25" floppy is in drive 2. Perform the copy by pressing return. You'll see that tracks 1 and 2 show only inverse A characters. The meaning is that Locksmith could not decode the address field of the sectors of these tracks.

THE SECOND MILESTONE IS TO COPY THE TWO PROTECTED TRACKS

- THE A.C.S PRESENTS MOBBY-DISK II 1972788 PISTE \$00/SECTEUR \$00/OCTET \$00/VOL. \$FE >CD
 CD
 C MODIFIED BY LOGO FOR THE L-BOOT \$ 000110203334455566777 MOBBY DISK I BY THE A.C.S ΙI 20/04/90 Ad Dn -> 85 96 AD BUFFER \$4000 DRIVE 1 AA AA $\langle ?$ for Help> 1 # ł
- 5. Now, press ctrl-openapple-reset to reboot and select Mobby Disk II.

6. Press ctrl-E then space then space to enter the monitor



7. Once in the monitor, at \$B971 replace BD 8C C0 with 4C 00 61 then press return



8. At \$6100, enter the following code and press return

002461000 0024680 0026680 0026680 0026680 0026800 00268000 002680000000000000000000000000000000000		00000000000000000000000000000000000000	
#6100÷BC	8C C0 10 FB	89 00 BA 85 28	BC 8
C C0 10 F	FB B9 00 BA	4A 29 0F 85 2D	18 60

9. The disassembled version must look like the following picture:

1=m 1=× 0=d		0=d	1=LCbank (0/1)	
00/610 00/610 000/610 000/610 000/610 000/611 000/611 000/611 000/611 000/611 000/611 000/611 000/611 000/611 000/611 000/611 000/611 000/611		CO CO BA CB BA CO SA CO SA CO SA CO SA CO SO CO SO CO SO SO SO SO SO SO SO SO SO SO SO SO SO	LDY C08C,X BPL 6100 (-05) LDA BA00,Y STA 2E LDY C08C,X BPL 610A (-05) LDA BA00,Y LSR AND #0F STA 2D CLC RTS BRK 00 BRK 00	
*				

10. Let's read the two protected tracks:

- a. Press ctrl-y to return to Mobby Disk II
- b. Put the original disk in drive 1
- c. Press R to read track 1, sector 0
- d. If OK, press down-arrow then press the letter O to confirm that you want to read the entire track 1
- e. Press down-arrow then press the letter O to confirm that you want to read the entire track 2

11. Let's write the tracks on our copy:

- a. Press ctrl-e, then space, then space to enter the monitor
- b. At \$B971, put BD 8C C0 then return
- c. Press ctrl-y to return to Mobby Disk II
- d. Put the copy disk in drive 1
- e. Press W and write track 1, sector 0
- f. Press ctrl-W then press the letter O to confirm that you want to write the entire buffer \$4000..\$4FFF to track 1
- g. Press ctrl-W then press the letter O to confirm that you want to write the entire buffer \$5000..\$5FFF to track 2

THE THIRD MILESTONE IS TO NORMALIZE THE DISK

12. Now, press ctrl-openapple-reset to reboot and select Disk Fixer V4.0 (the world's most powerful 5.25" sector editor on Earth)

TRACK \$00/SEC	- DISK EDIT TOR \$00/VOLUME :	\$00∕BYTE \$00
\$00: \$00: \$10: \$10: \$10: \$10: \$10: \$00: \$10: \$00: \$10: \$00: \$0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 </td <td></td>	
BUFFER 0/SLOT	6/DRIVE 1/MASK	OFF/NORMAL
COMMAND :		

13. As the Word Juggler disk is ProDOS based, we'll tell Disk Fixer to use the ProDOS interleaving. Press the letter O to enter the Options menu:

INPUT/OUTPUT CONTROL
ADRESS FIELD MARK : US-AA-96 (D5-AA-96) ADRESS FIELD EPILOG: DE-AA-EB (DE-AA-EB) DATA FIELD MARK : D5-AA-AD (D5-AA-AD) DATA FIELD EPILOG : DE-AA-EB (DE-AA-EB) CHECKSUM ENABLED : YES NUMBER OF TRACKS : \$24 (\$23)
DOS TYPE (DOS3.3 (DOS3.3)
\$\$\frac{1}{\$\$\\$00\$}\$

14. Now, press return then right arrow to move the cursor to the DOS TYPE entry. Once there, press P (for ProDOS ;-)) then press escape two times (note that I've also changed the number of tracks from \$24 to \$23 but you do not care, it is useful if you want to use the Find feature of the program)



15. We'll make changes to the WJ2E.SYSTEM file. Press D to enter the Directory feature of Disk Fixer. Select WJ2E.SYSTEM and press return.

The WJ2E.SYSTEM file contains strings in English. It is important to recall (I wrote that earlier) that some addresses may change due to your German version.

The WJ2E.SYSTEM file checks the computer the program is launched on, clears the screen, displays some messages and loads the WJ2E.2.9.0 file that is located on the first blocks of the ProDOS disk. The two formerly protected tracks are part of the file (and vice versa).

We must change some bytes in the WJ2E.SYSTEM file because it still contains the code to read protected tracks and our tracks are now readable. We must tell the program!

A ProDOS-based title has the advantage of (normally) being hard-disk-drive installable. That is not the case here, where tracks and sectors are read as if we were DOS 3.3-based. *Boo Quark*

	DIRECTORY MODE
NAME IS	ZWORD.JUGGLER
\$00:\$0E	WJ2E.2.9.0
\$07:\$02	PRODOS
\$0E:\$08	WJ2ELSYSTEM .
\$12:\$02	WJ.CONSOLE
\$12:\$0A	WJ.PARAMS
<u>\$1</u> <u>A</u> :≸08	TS.PARAMS
- 훈련용 : 훈련을	- 생님 - 특징수명
- 훈련한 : 훈련된 -	- 찜험· 특징부경
- 楽14:楽図し	- 씸닉·튼승부왇
- 単丁子:単均し - 本丁戸:本内人	· 쌈님·듣승부용
- 今1日:今回日 - 本回日:本回オ	83.6012
- 우인드 : 우인부 - 수이드 : 수이가	
	DE EDENNMY
- ≠0F÷≠0C - \$10:\$00	E E E SONEO
↓ 10: ↓00 ↓ 10: ↓06	PF ANAMEY
- <u>\$11:</u> \$йй	PF NP
- śiń:śйč	PESCRIBE
	PEIOKIDATA
\$18:\$08	PF.NEC8023
\$18:\$0E	PF.DTC380
	PRODOS

16. This is the first block of the WJ2E.SYSTEM file (note that I've pressed Y to remove the inverse characters on the right side of the window)

TRACK \$0A/SEC	- DISK EDIT ToR \$0E∕VOLUME	\$FE∕BYTE \$00
\$000 \$0000 \$000 \$000 \$000 \$000 \$000 \$000 \$000 \$000 \$000 \$000	0125E97D8E400090 076664624637772000090 017779020120000090 01777902012000090 0177790202012000090 01777909995E4000090 01777309995E4500090 0117773737666000090 011777909995E4500090 011777909995E4500090 011777909995E4500090 0117779099955 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 0117779099995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 011777909995 0110777909995 0110777909995 0110777909995 0110777909995 0110777909995 0110777909995 01107790995 01107790995 01107790995 01107790995 01107790995 01107790995 01107790995 01107790995 01107790995 01107790995 01107790995 0100000000000000000000000000000000	LU **0ua rk**.Wor d Juggle rversion 2.9 (C) Copyrig ht 1981 85 Quark Incorpo rated.x: JJ) *.i*h((8 \$.i
BUFFER 0/SLOT PRODOS:WJ2E.S	6/DRIVE 1/MASK YSTEM	OFF∕NORMAL ∕\$00
COMMAND : _		

17. The first change: remove the call to the read sector routine.

This is on the second half of the first block. So press right-arrow once to reach that part. Then, press M to move to offset \$DF:

TRACK ≸0A/SECTOR ≸0F/VOLUME ≸FE/BYTE ≸DF
\$80: B0 07 88 10 F7 4C C8 B2 0wLH2 \$80: A5 24 85 36 A5 25 85 37 %\$.6%%.7 \$90: A9 00 85 28 A0 0F 84 29 0* \$98: A5 24 85 36 A5 25 85 37 %\$.6%%.7 \$90: A9 00 85 28 A0 0F 84 29 0* \$98: A5 24 85 26 40 0F 84 29 0* \$98: A9 00 85 28 A0 0F 82 E6 29 AD .* \$98: B5 26 20 0F 82 E6 29 AD .* \$40: E5 25 28 F0 03 18 69 01 0E\$ \$40: E5 25 28 F0 03 18 69 01 0E\$ \$40: E5 28 C5 20 90 1D 20 0F . \$80: B5 28 C5 20 98 F0 80 27 C5 2%'170.% \$80: B5 28 65 38 85 38 E6 27 26 \$80: B2 A5 38 85 38 E6 38 E6 27 8e; \$65 38 85 38 85 38 E6 38 85 38 E6 27 8e;
\$C8: 20 90 08 E6 29 E6 37 C6 (†)†/†/ \$D0: 28 D0 E3 A5 28 0A 0A 0A (Pc%+) \$D8: 26 2A 0A 26 2A 85 29№20 \$E0: 0F B2 6C 24 00 41 50 5021\$.APP \$E8: 4C 45 20 2F 2F 45 20 52 LE //E R \$F0: 45 51 55 49 52 45 44 38 EQUIRED8 \$F8: 30 20 43 4F 4C 55 4D 4E 0 COLUMN
BUFFER 0/SLOT 6/DRIVE 1/MASK OFF/NORMAL PRODOS:WJ2E.SYSTEM /\$00 COMMAND :

The same sector as viewed from the Disassembly mode (press L). You can see that at offset \$DF we have the 20 0F B2 (or JSR \$B20F or call subroutine located at \$B20F)



Warning: the offset may vary on your German version. If so, search for the code shown at \$D3 above (A5 2B 0A 0A 0A 26 2A 0A 26 2A 85 29) and put a 2C instead of a 20 one byte after.

18. We must change the 20 (JSR) with a 2C (BIT) to bypass that call. Exit the disassembly mode by pressing the escape key. Move with I J K M to offset \$DF, press return and type 2C. Then press the escape key to exit the edit mode. Press W and press return two times to write the sector back onto disk.

TRACK \$0A/SECTOR \$0F/VOLUME \$FE/BYTE \$D3
\$80: B0 07 88 10 F7 4C C8 B2 0wLH2 \$88: A5 24 85 36 A5 25 85 37 %\$.6%.7
\$98: 85 2A 20 0F 82 E6 29 AD (* 12f)- \$A0: 13 80 C5 24 08 AD 14 80 (0E\$0
- \$H8: E5 25 28 F0 03 18 69 01 - €4(P1. \$B0: 85 28 C5 2C 90 1D 20 0F - (E, \$B8: B2 A5 27 C2 B7 B0 0C A5 - 2%'170,½ -
\$C0: 38 65 38 85 38 E6 27 C5 8e;.;f`E \$C8: 2C 90 08 E6 29 E6 37 C6 ,f)f7F \$D0: 28 D0 E3⊠A5≪28 0A 0A 0A (Pc%+
\$D8: 26 2A 0A 26 2A 85 29 20 &*.&*.), \$E0: 0F B2 6C 24 00 41 50 50 .21\$.APP \$E8: 4C 45 20 2F 2F 45 20 52 .LE //E R
\$F0: 45 51 55 49 52 45 44 38 EQUIRED8 \$F8: 30 20 43 4F 4C 55 4D 4E 0 COLUMN
BUFFER 0/SLOT 6/DRIVE 1/MASK OFF/NORMAL PRODOS:WJ2E.SYSTEM /\$00
COMMAND : _

This is the disassembled view of the modified sector. At offset \$DF, the JSR was changed to a BIT:

00D3:A5 00D5:0A 00D6:0A	28	- DISASSEÞ	IBLY LDA ASL ASL	MODE \$2B
00D7:0H 00D8:26	2A		ROL	\$2A
000DB - 265 000DF - 265 000DF - 260 000EF - 260 000E7 - 540 000E7 - 545 000E9 - 25 000E9 - 25 000E9 - 25	49E4000 22022542	82 00	ILATARCRE IRSBJEBEE	\$2A \$29 \$B20F (\$0024) (\$50,X) \$0135 \$20
00000000000000000000000000000000000000	000524 24554 2		BUCCOCOUNT BUCCOCOUNT BUCCCCCHO	\$20 (\$45),Y (\$55),Y #\$52 \$44 \$011A

19. The second change: force the address field read routine to always read a standard address field. Ahem! See the last part of the guide to understand!

Press right-arrow to move to the next block (see in the lower right corner of the window, "/\$01" means that we are on block 1 of the WJ2E.SYSTEM file). Move to offset \$23 and press L to view the disassembled listing.

TRACK \$0E/SECTOR \$0B/VOLUME \$FE/BYTE \$23
\$00: E0 B8 90 F4 B0 FE AE 02 ^8.t0~. \$08: 02 A0 F8 84 39 C8 D0 04 . ×.9HP. \$10: E6 39 F0 68 BD 8C C0 10 f9ph=.@.
\$18: FB C9 D5 D0 F0 EA BD 8C €IUPpj=. \$20: C0 10 FB⊠C9€AA D0 F2 EA €.€I*Prj \$28: BD 8C C0 10 FB C9 96 D0 =.€.€I.P \$70: DC 0D 00 00 F9 01 40 D0
\$38: 4E 20 7E B3 8D 07 02 20 N *3 \$40: 7E B3 4A 29 0F 8D 06 02 *3J) \$48: 09 AA 85 39 A5 00 A0 17
\$50: 88 D0 FD BD 8C C0 10 FB
\$70: 18 BD 8C C0 10 FB 49 AA .=.0.€I* \$78: 85 2C F0 01 38 60 BC 8C
BUFFER Ø∕SLOT 6/DRIVE 1/MASK OFF/NORMAL PRODOS÷WJ2E.SYSTEM /≉01
COMMAND : _

At offset \$31, the code reads the address \$200, subtracts 1 to it, divide it by 2 and if the value is non-zero, it branches to offset \$87, otherwise it continues at offset \$39.

0023:C9 0025:D0	AA F2	DISASSEN	IBLY N CMP BNE	10DE #\$AA \$0019
0027:EA 0028:BD 0028:10 0028:10	80 FB 96	C0	NUP LDA BPL CMP	\$C08C,X \$0028 #≴96
002F:00 0031:AD 0034:E9	DC 00 01	02	BNE LDA SBC	\$000D \$0200 #\$01
0036:40 0037:20 0039:20 0030:80 0031:20	4E 707 7E	83 92 83	LSE JSE JSE JSE JSE JSE	\$0087 \$837E \$0207 \$837E
00443:29 0045:80 0045:80 0048:85 0044:45	0F 06490	02	LONDA STA STA STA LDS	#\$0F \$0206 #\$AA \$39 \$00_
004E:40 0050:88 0051:D0	FD		DEY BNE	#\$17 \$0050

Warning: the offset may vary on your version. If so, search for the code shown at \$31 (AD 00 02 E9 01 4A D0 4E) and put a 4C xx one byte after. Xx is the value shown to the right of offset \$37. On the US version, it shows "BNE \$0087" so xx is 87. If your version shows "BNE \$0089" then xx is 89.

20. We must make the change after the code shown above to always call the standard address field read routine instead of the modified one for tracks 1 and 2.

Move to offset \$39 and replace the 20 7E values with 4C 87 and write the sector back onto disk. To change the values, press return once at offset \$39 to enter the edit mode, type 4C 87 then press escape to exit the edit mode. Press W then return two times to write the sector.

TRACK \$0E/SECTOR \$0B/VOLUME \$FE/BYTE \$39	
\$00: E0 B8 90 F4 B0 FE AE 02 `8.t0" \$08: 02 A0 F8 84 39 C8 D0 04 x.9HP. \$10: E6 39 F0 68 BD 8C C0 10 f9ph=.0. \$18: FB C9 D5 D0 F0 EA BD 8C €IUPpj=. \$20: C0 10 F8 C9 AA D0 F2 EA 0.€.€I*Prj	
\$28: BU 80 00 10 FB 09 96 D0 = .@.(1.F \$30: DC AD 00 02 E9 01 4A D0 \i.JF \$38: 4E <mark>>4C<87</mark> B3 8D 07 02 20 NL.3 \$40: 7E B3 4A 29 0F 8D 06 02 "3J) \$48: 09 AA 85 39 A5 00 A0 17	
\$58: C5 39 DØ 20 20 7E B3 4D E9P ~3M \$60: Ø6 Ø2 4D Ø7 Ø2 DØ 15 BD MP.= \$68: 8C CØ 10 FB C9 DE DØ ØC MP.= \$70: 18 BD 8C CØ 10 FB 49 AA P.8\<	
BUFFER Ø/SLOT 6/DRIVE 1/MASK OFF/NORMAL PRODOS:WJ2E.SYSTEM /#01	

The same sector once modified in the disassembled mode (press L):

0023:C9 AF	- DISASSE	MBLY N CMP BNF	10DE #\$АА ≴ЙЙ19
0027:EA 0028:BD 80 0028:10 FE	cø	NÖP LDA BPL	\$C08C,X \$0028
002D:C9 98 002F:D0 D0 0031:AD 00	02	CMP BNE LDA	#\$96° \$000D \$0200
0034:E9 01 0036:44 0037:D0 4E		SBC LSR BNE	#≄01 \$0087
0039:4C 87 003C:8D 07 003F:20 7E	2 83 92 83	JMP STA JSR	\$B387 \$0207 \$B37E
0042:4H 0043:29 0F 0045:8D 06	02		#\$0F ≸0206
0046:05 HB 0046:85 39 004C:65 00		STA LBA	##HH \$39 \$00 ##17
0050:88 0051:D0 FD)	DEY BNE	\$0050

You can see at offset \$39 that the JSR was replaced with the JMP to the same address as offset \$39. It is the same address as that code runs at address \$B300. So "BNE \$0087" and "JMP \$B3B7" go to the same address.

CONGRATULATIONS! YOU'VE CRACKED YOUR COPY

Let's share some information about the protection type. A standard 5.25" 16-sector floppy disk is divided into 35 tracks of 16 sectors and the data stored on disk is stored in a different way than the bytes in memory. We call them nibbles.

Each track/sector information is stored in an address field and a sector data is stored in a data field. Both in a format that is well designed and known.

A standard address field looks like this:

- header markers: D5 AA 96
- volume information: AA AA (coded in 4*4 format, only 4 bits of each nibble contain valid data)
- track information: AA AA (ditto)
- sector information: AA AA (ditto)
- checksum: AA AA (ditto)
- epilog markers: DE AA (with a final EB but due to a bug, it is not written)

What Quark did for tracks 1 and 2 is a change of the address field:

- header marker: D5 AA 96
- track information on one nibble
- sector information on one nibble
- some bits to desync and lose time
- sector information in AA format on one nibble
- checksum on one nibble
- epilog markers: DE AA

The second change we did was to force the usage of the standard address field read routine whatever the track we are in.

Reboot and ... enjoy,

LoGo 5/2017

AS USUAL, SOME CODE

0031:AD 00 02 0034:E9 01	LDA SBC	\$0200 #\$01	; ;	get track -1
0036:4A	LSR		;	/2
0037:D0 4E	BNE	\$0087	;	branch if >0
* Read a modified a	address	field		
0039:20 7E B3	JSR	\$B37E	;	read nibble
003C:8D 07 02	STA	\$0207	;	save as track
003F:20 7E B3	JSR	\$B37E	;	read nibble
0042:4A	LSR		;	/2
0043:29 OF	AND	#\$0F	;	keep lower 4 bits
0045:8D 06 02	STA	\$0206	;	save as sector
0048:09 AA	ORA	#\$AA	;	make it 4*4
004A:85 39	STA	\$39	;	save it
004C:A5 00	LDA	\$00	;	waste time
004E:A0 17	LDY	#\$17	;	
0050:88	DEY		;	••
0051:D0 FD	BNE	\$0050	;	
0053:BD 8C C0	LDA	\$C08C,X	;	read nibble
0056:10 FB	BPL	\$0053		
0058:C5 39	CMP	\$39	;	same as requested?
005A:D0 20	BNE	\$007C	;	nope, error
005C:20 7E B3	JSR	\$B37E	;	read nibble
005F:4D 06 02	EOR	\$0206	;	checksum
0062:40 07 02	EOR	\$0207	;	checksum
0065:D0 15	BNE	\$007C	;	error, branch
0067:BD 8C C0	T.DA	\$C08C.X	;	get marker
006A:10 FB	BPI.	\$0067	,	900
006C·C9 DE	CMP	#\$DE		
$0.6E \cdot D0 0C$	BNE	\$007C		
0070.18	CLC	40070		
$0.071 \cdot BD \ 8C \ C0$	T.DA	ŚCO8C-X		and 2nd marker
0074·10 FB	RPI.	\$0071	'	
0076:49 33	EOR	#\$AA		
0078.85 20	STA	\$20		save it
0070.0320	BEU	\$20 \$007D	,	Save IC
007C.38	CEC CEC	ÇUU7D		orror
0070.60	DIC			or po error if c=0
0070.00	RIS		,	OI NO EIIOI II C-U
007E:BC 8C C0	LDY	\$C08C,X	;	read a nibble
0081:10 FB	BPL	\$007E	;	thanks to the denibblize
0083:B9 00 B5	LDA	\$B500,Y	;	table, transform it
0086:60	RTS			
* Read a standard a	address	field		
0087:A0 03	LDY	#\$03	;	the standard address
0089:A9 00	LDA	#\$00	;	field read routine
008B:38	SEC		,	
008C:85 39	STA	\$39		
008E:BD 8C C0	LDA	\$C08C,X		
0091:10 FB	BPL	\$008E		

0093:2A			ROL				
0094:85	ЗA		STA	\$3A			
0096:BD	8C	C0	LDA	\$C08C,X			
0099:10	FΒ		BPL	\$0096	;	\$205:	checksum
009B:25	ЗA		AND	\$3A	;	\$206:	sector
009D:99	05	02	STA	\$0205 , Y	;	\$207:	track
00A0:45	39		EOR	\$39	;	\$208:	volume
00A2:88			DEY				
00A3:10	Ε7		BPL	\$008C			
00A5:A8			TAY				
00A6:F0	ΒF		BEQ	\$0067			
00A8:D0	D2		BNE	\$007C	;	until	here

*--- The denibblize table

00/B590:00 00 00 00 00 00 00 01-.... 00/B598:98 99 02 03 9C 04 05 06-... 00/B5A0:A0 A1 A2 A3 A4 A5 07 08- !"#\$%.. 00/B5A8:A8 A9 AA 09 0A 0B 0C 0D-()*... 00/B5B0:B0 B1 0E 0F 10 11 12 13-01... 00/B5B8:B8 14 15 16 17 18 19 1A-8... 00/B5C0:C0 C1 C2 C3 C4 C5 C6 C7-@ABCDEFG 00/B5C8:C8 C9 CA 1B CC 1C 1D 1E-HIJ.L.. 00/B5D0:D0 D1 D2 1F D4 D5 20 21-PQR.TU ! 00/B5D8:D8 22 23 24 25 26 27 28-X"#\$%&'(00/B5E0:E0 E1 E2 E3 E4 29 2A 2B-`!"#\$)*+ 00/B5E8:E8 2C 2D 2E 2F 30 31 32-(,-./012 00/B5F0:F0 F1 33 34 35 36 37 38-01345678 00/B5F8:F8 39 3A 3B 3C 3D 3E 3F-89:;<=>?