

M. David Johnson
<http://www.bds-soft.com>
info@bds-soft.com

Key Codes and VIDRAM

by M. David Johnson

2021/09/29

Abstract

A detailed examination of CoCo 2, CoCo 3, and VCC Keyboards, as well as the CoCo VIDRAM Text Screen, is presented. Differences between ASCII and the CoCo Character Set are discussed. The paper is configured to serve as a reference work.

—

This paper and its associated code are available online at:

<http://www.bds-soft.com/cocoPapers.php> .

====

Table of Contents

Abstract	1
Introduction	5
General Methodology	6

Appendix A: ASCII (0-127)	8
Appendix B: 64K CoCo 2 Key Codes All Upper Case Keyboard	35
Appendix C: 64K CoCo 2 Key Codes Upper/Lower Case Keyboard	38
Appendix D: 64K CoCo 2 Keyboard Key Code Coverage	41
Appendix E: CoCo 3 Key Codes All Upper Case Keyboard	44
Appendix F: CoCo 3 Key Codes Upper/Lower Case Keyboard	48
Appendix G: CoCo 3 Keyboard Key Code Coverage	52
Appendix H: VCC CoCo (DECB) All Upper Case Keyboard Key Codes	55
Appendix I: VCC CoCo (DECB) Upper/Lower Case Keyboard Key Codes	60
Appendix J: VCC CoCo (DECB) Keyboard Key Code Coverage	65
Appendix K: VCC Natural (OS-9) All Upper Case Keyboard Key Codes	68
Appendix L: VCC Natural (OS-9) Upper/Lower Case Keyboard Key Codes	74

Appendix M: VCC Natural (OS-9)	
Keyboard Key Code Coverage	80
Appendix N: VCC Compact (OS-9)	
All Upper Case Keyboard Key Codes	83
Appendix O: VCC Compact (OS-9)	
Upper/Lower Case Keyboard Key Codes	88
Appendix P: VCC Compact (OS-9)	
Keyboard Key Code Coverage	93
Appendix Q: Various PC Keyboards (For Reference Only) ...	96
Appendix R: Extended ASCII (128-255)	99
Appendix S: Decimal to Hexadecimal Conversions	122
Appendix T: Programs	124
Appendix U: My 64K CoCo 2 System	140
Appendix V: My 128K CoCo 3 System	141
Appendix W: My 512K CoCo 3 System	143
Appendix X: My CoCo Philosophy	145
Appendix Y: New BDS Software License	147

Works Cited	148

=====

Introduction

After what is now roughly a 30-year absence from all things CoCo, I needed to reacquaint myself with some stuff, learn some other stuff all over again from scratch, and also learn some new stuff that didn't even exist thirty years ago.

Many of you "Old Heads" (those who have been actively and continuously engaged with the CoCo over that 30-year period) probably already know all this stuff, or, if you don't, you'd approach it differently (and probably more efficiently) than I have here.

But, perhaps someone new to the CoCo world will find some of this to be somewhat helpful. And, since my CoCo Philosophy (cf. Appendix X) involves ~~dragging some newbies, kicking and screaming...~~ uh... um... that is... gently inviting some of those who have never even heard of the CoCo to enter into the wonders of exploring our little machine, I've put this information together as a possible help.

Since I'm more interested in text processing and number crunching than I am in graphics, one of the first things I wanted to do was to re-familiarize myself with the CoCo Keyboard(s) and the CoCo's VIDRAM Text Screen.

I have a CoCo 2 and two CoCo 3's, but I'm doing the majority of my investigating, programming, and developing in the VCC Emulator. So, essentially, one VIDRAM and five Keyboards are available to play with:

- Physical CoCo 2 Keyboard
- Physical CoCo 3 Keyboard
- VCC CoCo (DECB) Keyboard
- VCC Natural (OS-9) Keyboard
- VCC Compact (OS-9) Keyboard

In this paper, I'm primarily engaged in exploring the differences between the five Keyboards and how they each interact with the VIDRAM.

Much of what you'll find in this paper is highly redundant. That's intentional: I want this to serve as a convenient reference, where you won't have to be paging back-and-forth to find a particular piece of related information.

A Note on Numbers: To keep everything simple to understand, and also neatly lined-up, I generally refer to numbers as decimal bytes with three full digits, e.g. 004, 027, 229, etc. See Appendix S for conversions between the decimal and hexadecimal representations of bytes.

In works of this complexity (at least for me) typos and other errors are bound to sneak in. Please let me know about any you discover so I can note and correct them.

M.D.J. 2021/09/29
info@bds-soft.com

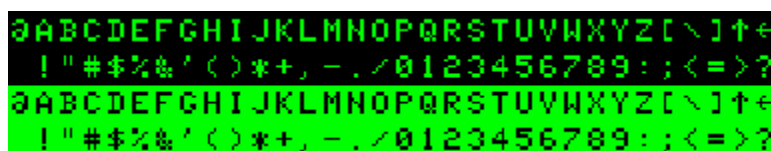
=====

General Methodology

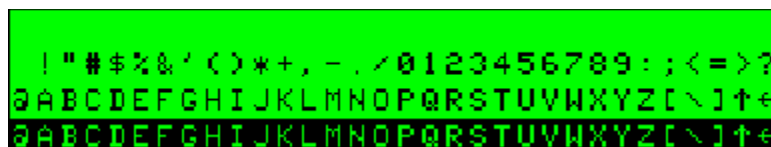
The CoCo Text Screen is an interesting place, even though (or perhaps because) it has some intriguing traps for the unwary.

VIDRAM (Video RAM) is (at least initially) located between hexadecimal addresses &H0400 and &H05FF inclusive. If we POKE decimal character codes 000-127 into VIDRAM, we get one result; but if we PRINT those same codes to VIDRAM, we get a different result.

Using the **POKECHK.BAS** program (See Appendix T), the POKE mechanism display for Code Points 000-127:



Using the **PRTCHRCK.BAS** program (See Appendix T), the PRINT mechanism display for Code Points 000-127:



Note that the top line of the PRINT mechanism display is all green, not because it's filled with printed spaces, but because the Carriage Return+Line Feed (CRLF) character drops down to the following (second) line. Characters 000-031 are non-printing control codes, both in ASCII, and also in the CoCo's character set.

Code Point 008 is the Backspace character. When it is encountered during this PRINT mechanism test, the cursor is still located in the upper-left corner of the screen since Code Points 000 through 007 were non-printing. The Backspace thus does nothing on the screen in this instance.

Similarly, when Code Point 013 (CRLF) is encountered, the cursor is still located in the upper-left corner of the screen since Code Points 009 through 012 were also non-printing. The CRLF thus returns and drops down to the first position on the second line.

When Code Point 032 (Space) is encountered, the cursor is still located at the first position of the second line since Code Points 014 through 031 were also non-printing. Code Point 032 is then displayed as the green space in the first position on the second line, just to the left of the Exclamation Point.

For perhaps a little more visual clarity, here are the same images in black-and-white:

POKE mechanism BW display for Code Points 000-127:

```
0ABCDEFGHIJKLMN0PQRSTUVWXYZ[\]↑+
!"#$%&'()*+,-./0123456789:;<=>?
0ABCDEFGHIJKLMN0PQRSTUVWXYZ[\]↑+
!"#$%&'()*+,-./0123456789:;<=>?
```

PRINT mechanism BW display for Code Points 000-127:

```
!"#$%&'()*+,-./0123456789:;<=>?
0ABCDEFGHIJKLMN0PQRSTUVWXYZ[\]↑+
0ABCDEFGHIJKLMN0PQRSTUVWXYZ[\]↑+
```

In this paper, I begin by examining the similarities and differences between ASCII and the CoCo's VIDRAM-oriented Alphanumeric character set (Code Points 000-127).

Then, I examine the similarities and differences between the five Keyboards in detail.

And, finally, I explore and discuss the Semigraphics character set (Code Points 128-255) which accompanies the Alphanumeric set pretty much everywhere it goes.

=====

Appendix A: ASCII (0-127)

ASCII is the American Standard Code for Information Interchange and, as such, it only applies to Code Points 0-127.

The original IBM PC used a set of 255 symbols which was then known as DOS Latin. This set is now known as Code Page 437.

For each of Code Points 0-127 (000-127), the following entries show:

1. The ASCII Symbol (and Name, if any).
2. For reference, how Code Page 437 handled that Code Point.
3. The Symbol displayed by the CoCo as a result of:
POKE (addr, cp) where:
addr = address in VIDRAM (&H0400-&H05FF)
cp = the Code Point (0-127).
4. The Symbol displayed by the CoCo as a result of:
PRINT (CHR\$ (cp)) where:
cp = the Code Point (0-127).

Note that although *Getting Started With Extended Color BASIC* (241-243) refers to the CoCo's character set as ASCII, it really isn't ASCII at all. It bears some useful resemblance to ASCII, but differs in many ways.

First, instead of true uppercase and lowercase characters, the CoCo 2 uses "Standard" uppercase characters (black on green) for uppercase and "Reversed" uppercase characters (green on black) for lowercase. A more conventional character set is available on the CoCo 3 on its 40- and 80- column screens, but such is not available on either the CoCo 2, or on the CoCo 3's out of the box 32-column screen.

Second, ASCII Code Points 0-31 and 127 are (generally non-printing) control codes. But the CoCo displays a character for Code Point 127.

Third, the CoCo's display characters (symbols) for Code Points 094, 095, 096, 123, 124, 125, and 126 do not match those shown in the ASCII Standard.

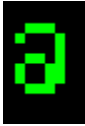
Finally, the characters (symbols) displayed by the CoCo vary depending upon whether the characters are displayed via the POKE mechanism, or via the PRINT mechanism. The characters are the same for Code Points 064-095, but are different for Code Points 000-063 and 096-127.

All Character Codes (000-127) presented below are in decimal format. See Appendix S for decimal to hexadecimal conversions.

The ASCII Codes and Symbols are taken from https://en.wikipedia.org/wiki/ASCII_1967_data.

The CP437 (Code Page 437) Codes and Symbols are taken from:
https://en.wikipedia.org/wiki/Code_page_437 ,
and are as produced by PC BASIC.

ASCII 000 = NUL = Null -> CP437 = Non-printing



Poke 000 (none) Print 000

ASCII 001 = SOH = Start of Heading -> CP437 displays _



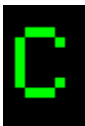
Poke 001 (none) Print 001

ASCII 002 = STX = Start of Text -> CP437 = Non-printing



Poke 002 (none) Print 002

ASCII 003 = ETX = End of Text -> CP437 displays _



Poke 003 (none) Print 003

ASCII 004 = EOT = End of Transmission -> CP437 displays _



Poke 004 (none) Print 004

ASCII 005 = ENQ = Enquiry -> CP437 displays _



Poke 005 (none) Print 005

—

ASCII 006 = ACK = Acknowledgment -> CP437 displays _



Poke 006 (none) Print 006

—

ASCII 007 = BEL = Bell -> CP437 displays _



Poke 007 (none) Print 007

—

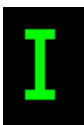
ASCII 008 = BS = Backspace -> CP437 displays _



Poke 008 (none) Print 008 (Does a Backspace)

—

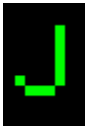
ASCII 009 = HT = Horizontal Tab -> CP437 = Non-printing



Poke 009 (none) Print 009

—

ASCII 010 = LF = Line Feed ->
CP437 displays a line feed with PRINT A\$
But is non-printing with PRINT A\$;



Poke 010 (none) Print 010

—

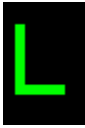
ASCII 011 = VT = Vertical Tab -> CP437 = Non-printing



Poke 011 (none) Print 011

—

ASCII 012 = FF = Form Feed -> CP437 = Non-printing



Poke 012 (none) Print 012

—

ASCII 013 = CR = Carriage Return ->
CP437 displays a carriage return with PRINT A\$
But is non-printing with PRINT A\$;



Poke 013 (none) Print 013 (Does a CRLF*)
*Carriage Return + Line Feed

—

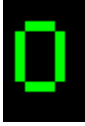
ASCII 014 = SO = Shift Out -> CP437 displays _



Poke 014 (none) Print 014

—

ASCII 015 = SI = Shift In -> CP437 displays _



Poke 015 (none) Print 015

—

ASCII 016 = DLE = Data Link Escape ->
CP437 displays _! with PRINT A\$
But displays _ with PRINT A\$;



Poke 016 (none) Print 016

—

ASCII 017 = DC1 = Device Control 1 -> CP437 displays _



Poke 017 (none) Print 017

—

ASCII 018 = DC2 = Device Control 2 -> CP437 displays _



Poke 018 (none) Print 018

—

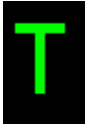
ASCII 019 = DC3 = Device Control 3 -> CP437 displays _



Poke 019 (none) Print 019

—

ASCII 020 = DC4 = Device Control 4 -> CP437 displays _



Poke 020 (none) Print 020

—

ASCII 021 = NAK = Negative Acknowledgement ->
CP437 displays _



Poke 021 (none) Print 021

—

ASCII 022 = SYN = Synchronous Idle -> CP437 displays _



Poke 022 (none) Print 022

—

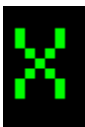
ASCII 023 = ETB = End of Transmission Block ->
CP437 displays _



Poke 023 (none) Print 023

—

ASCII 024 = CAN = Cancel -> CP437 displays _



Poke 024 (none) Print 024

—

ASCII 025 = EM = End of Medium -> CP437 displays _



Poke 025 (none) Print 025

—

ASCII 026 = SUB = Substitute -> CP437 displays _



Poke 026 (none) Print 026

—

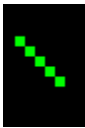
ASCII 027 = ESC = Escape -> CP437 displays _



Poke 027 (none) Print 027

—

ASCII 028 = FS = File Separator -> CP437 = Non-printing



Poke 028 (none) Print 028

—

ASCII 029 = GS = Group Separator -> CP437 = Non-printing



Poke 029 (none) Print 029

—

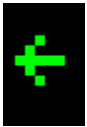
ASCII 030 = RS = Record Separator -> CP437 = Non-printing



Poke 030 (none) Print 030

—

ASCII 031 = US = Unit Separator -> CP437 = Non-printing



Poke 031 (none) Print 031

—

ASCII 032 = CP437 = space



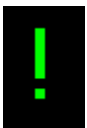
Poke 032



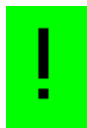
Print 032

—

ASCII 033 = CP437 = !



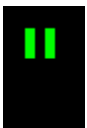
Poke 033



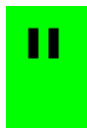
Print 033

—

ASCII 034 = CP437 = "



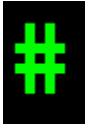
Poke 034



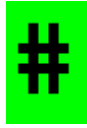
Print 034

—

ASCII 035 = CP437 = #



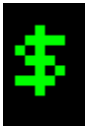
Poke 035



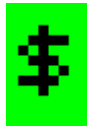
Print 035

—

ASCII 036 = CP437 = \$



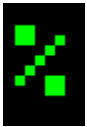
Poke 036



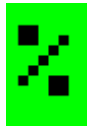
Print 036

—

ASCII 037 = CP437 = %



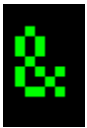
Poke 037



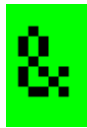
Print 037

—

ASCII 038 = CP437 = & (ampersand)



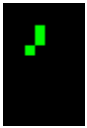
Poke 038



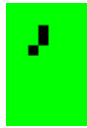
Print 038

—

ASCII 039 = CP437 = ' (single quote)



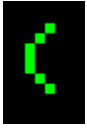
Poke 039



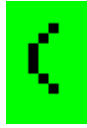
Print 039

—

ASCII 040 = CP437 = ((opening parenthesis)



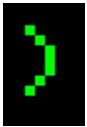
Poke 040



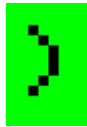
Print 040

—

ASCII 041 = CP437 =) (closing parenthesis)



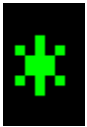
Poke 041



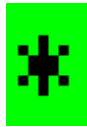
Print 041

—

ASCII 042 = CP437 = *



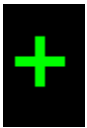
Poke 042



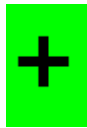
Print 042

—

ASCII 043 = CP437 = +



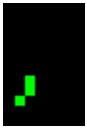
Poke 043



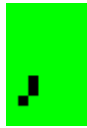
Print 043

—

ASCII 044 = CP437 = , (comma)



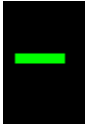
Poke 044



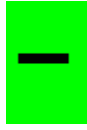
Print 044

—

ASCII 045 = CP437 = -



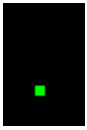
Poke 045



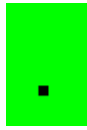
Print 045

—

ASCII 046 = CP437 = .



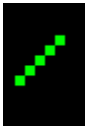
Poke 046



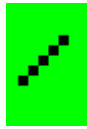
Print 046

—

ASCII 047 = CP437 = /



Poke 047



Print 047

—

ASCII 048 = CP437 = 0 (numeral zero)



Poke 048



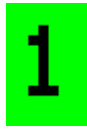
Print 048

—

ASCII 049 = CP437 = 1 (numeral one)



Poke 049



Print 049

—

ASCII 050 = CP437 = 2 (numeral two)



Poke 050



Print 050

—

ASCII 051 = CP437 = 3



Poke 051



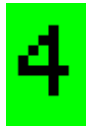
Print 051

—

ASCII 052 = CP437 = 4



Poke 052



Print 052

—

ASCII 053 = CP437 = 5 (numeral five)



Poke 053



Print 053

—

ASCII 054 = CP437 = 6 (numeral six)



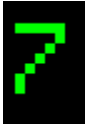
Poke 054



Print 054

—

ASCII 055 = CP437 = 7



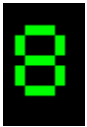
Poke 055



Print 055

—

ASCII 056 = CP437 = 8 (numeral eight)



Poke 056



Print 056

—

ASCII 057 = CP437 = 9 (numeral nine)



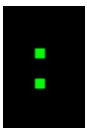
Poke 057



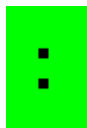
Print 057

—

ASCII 058 = CP437 = :



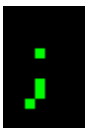
Poke 058



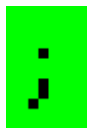
Print 058

—

ASCII 059 = CP437 = ;



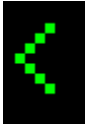
Poke 059



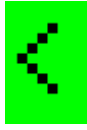
Print 059

—

ASCII 060 = CP437 = <



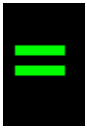
Poke 060



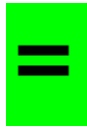
Print 060

—

ASCII 061 = CP437 = =



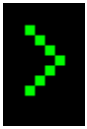
Poke 061



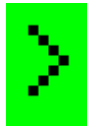
Print 061

—

ASCII 062 = CP437 = >



Poke 062



Print 062

—

ASCII 063 = CP437 = ?



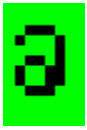
Poke 063



Print 063

—

ASCII 064 = CP437 = @ ("AT" symbol)



Poke 064



Print 064

—

ASCII 065 = CP437 = A



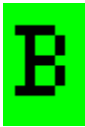
Poke 065



Print 065

—

ASCII 066 = CP437 = B (uppercase "B")



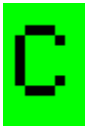
Poke 066



Print 066

—

ASCII 067 = CP437 = C



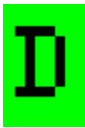
Poke 067



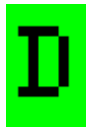
Print 67

—

ASCII 068 = CP437 = D (uppercase "D")



Poke 068



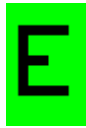
Print 068

—

ASCII 069 = CP437 = E



Poke 069



Print 069

—

ASCII 070 = CP437 = F



Poke 070



Print 070

—

ASCII 071 = CP437 = G



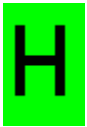
Poke 071



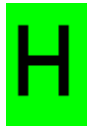
Print 071

—

ASCII 072 = CP437 = H



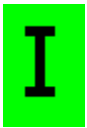
Poke 072



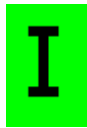
Print 073

—

ASCII 073 = CP437 = I



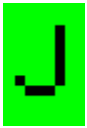
Poke 073



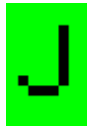
Print 073

—

ASCII 074 = CP437 = J



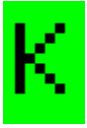
Poke 074



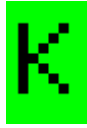
Print 074

—

ASCII 075 = CP437 = K



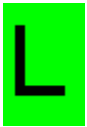
Poke 075



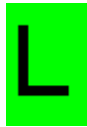
Print 075

—

ASCII 076 = CP437 = L



Poke 076



Print 076

—

ASCII 077 = CP437 = M



Poke 077



Print 077

—

ASCII 078 = CP437 = N



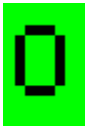
Poke 078



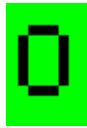
Print 078

—

ASCII 079 = CP437 = O (uppercase "O")



Poke 079



Print 079

—

ASCII 080 = CP437 = P



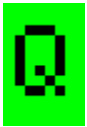
Poke 080



Print 080

—

ASCII 081 = CP437 = Q



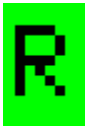
Poke 081



Print 081

—

ASCII 082 = CP437 = R



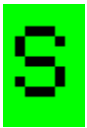
Poke 082



Print 082

—

ASCII 083 = CP437 = S (uppercase "S")



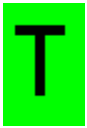
Poke 083



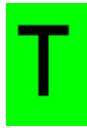
Print 083

—

ASCII 084 = CP437 = T



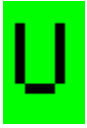
Poke 084



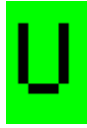
Print 084

—

ASCII 085 = CP437 = U



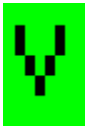
Poke 085



Print 085

—

ASCII 086 = CP437 = V



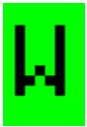
Poke 086



Print 086

—

ASCII 087 = CP437 = W



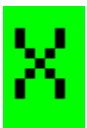
Poke 087



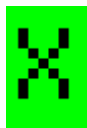
Print 087

—

ASCII 088 = CP437 = X



Poke 088



Print 088

—

ASCII 089 = CP437 = Y



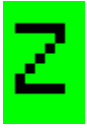
Poke 089



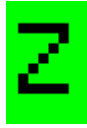
Print 089

—

ASCII 090 = CP437 = Z (uppercase "Z")



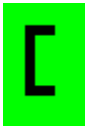
Poke 090



Print 090

—

ASCII 091 = CP437 = [(opening bracket)



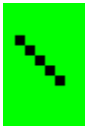
Poke 091



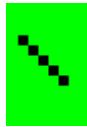
Print 091

—

ASCII 092 = CP437 = \ (backslash)



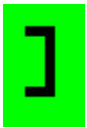
Poke 092



Print 092

—

ASCII 093 = CP437 =] (closing bracket)



Poke 093



Print 093

—

ASCII 094 = CP437 = ^ (exponentiation symbol)



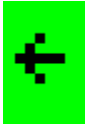
Poke 094



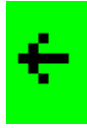
Print 094

—

ASCII 095 = CP437 = _ (underscore)



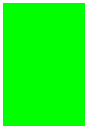
Poke 095



Print 095

—

ASCII 096 = CP437 = ` (backtick)



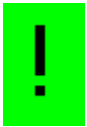
Poke 096



Print 096

—

ASCII 097 = CP437 = a (lowercase "A")



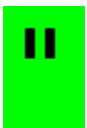
Poke 097



Print 097

—

ASCII 098 = CP437 = b (lowercase "B")



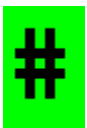
Poke 098



Print 098

—

ASCII 099 = CP437 = c



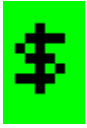
Poke 099



Print 099

—

ASCII 100 = CP437 = d



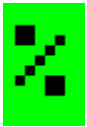
Poke 100



Print 100

—

ASCII 101 = CP437 = e



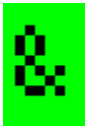
Poke 101



Print 101

—

ASCII 102 = CP437 = f



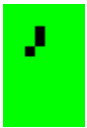
Poke 102



Print 102

—

ASCII 103 = CP437 = g



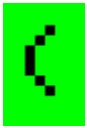
Poke 103



Print 103

—

ASCII 104 = CP437 = h



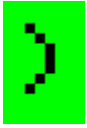
Poke 104



Print 104

—

ASCII 105 = CP437 = I



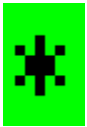
Poke 105



Print 105

—

ASCII 106 = CP437 = j



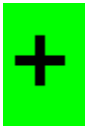
Poke 106



Print 106

—

ASCII 107 = CP437 = k



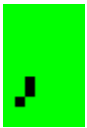
Poke 107



Print 107

—

ASCII 108 = CP437 = l (lowercase "L")



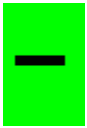
Poke 108



Print 108

—

ASCII 109 = CP437 = m



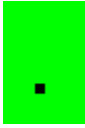
Poke 109



Print 109

—

ASCII 110 = CP437 = n



Poke 110



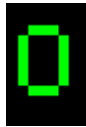
Print 110

—

ASCII 111 = CP437 = o (lowercase "O")



Poke 111



Print 111

—

ASCII 112 = CP437 = p



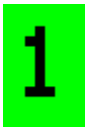
Poke 112



Print 112

—

ASCII 113 = CP437 = q (lowercase "Q")



Poke 113



Print 113

—

ASCII 114 = CP437 = r



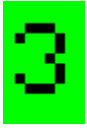
Poke 114



Print 114

—

ASCII 115 = CP437 = s (lowercase "S")



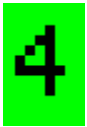
Poke 115



Print 115

—

ASCII 116 = CP437 = t



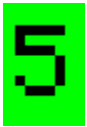
Poke 116



Print 116

—

ASCII 117 = CP437 = u



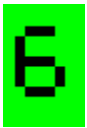
Poke 117



Print 117

—

ASCII 118 = CP437 = v



Poke 118



Print 118

—

ASCII 119 = CP437 = w



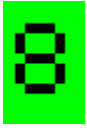
Poke 119



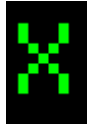
Print 119

—

ASCII 120 = CP437 = x



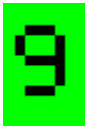
Poke 120



Print 120

—

ASCII 121 = CP437 = y



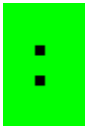
Poke 121



Print 121

—

ASCII 122 = CP437 = z (lowercase "z")



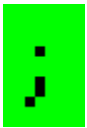
Poke 122



Print 122

—

ASCII 123 = CP437 = { (opening brace)



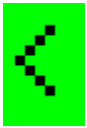
Poke 123



Print 123

—

ASCII 124 = CP437 = | (vertical bar/pipe)



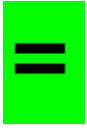
Poke 124



Print 124

—

ASCII 125 = CP437 = } (closing brace)



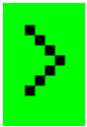
Poke 125



Print 125

—

ASCII 126 = CP437 = ~ (tilde)



Poke 126



Print 126

—

ASCII 127 = DEL = Delete -> CP437 displays _



Poke 127



Print 127

=====

Appendix B: 64K CoCo 2 Key Codes

“All Upper Case” Keyboard

Physical 64K CoCo 2 Keyboard:

```
[1!][2"][3#][4$][5%][6&][7'][8()][9)][ 0][:][*][-=][BR]
[UA][ Q][ W][ E][ R][ T][ Y][ U][ I][ O][ P][ @][LA][RA]
[DA][ A][ S][ D][ F][ G][ H][ J][ K][ L][;+][ENTR][CLR]
[SHFT][ Z][ X][ C][ V][ B][ N][ M][,<][.>][/?][SHFT]
[                               Space                               ]
```

BR = BREAK
 CLR = CLEAR
 ENTR = ENTER
 SHFT = SHIFT

UA = Up Arrow
 RA = Right Arrow
 DA = Down Arrow
 LA = Left Arrow

These are the characters and Key Codes returned by key presses on a physical 64K CoCo 2 keyboard. (See Appendix U for my physical 64K CoCo 2 setup).

They are from the keyboard as it is immediately after startup.

They also reflect line “245 POKE 282,1” in the **KEYCHECK.BAS** program (See Appendix T).

All characters in this list are CoCo Standard Characters, i.e. black-on-green, except as noted.

KEY PRESS	UNSHIFTED		SHIFTED		NOTE
	CHAR	CODE	CHAR	CODE	
1!	1	049	!	033	
2"	2	050	"	034	
3#	3	051	#	035	
4\$	4	052	\$	036	
5%	5	053	%	037	
6&	6	054	&	038	
7'	7	055	'	039	
8(8	056	(040	

9)	9	057)	041	
0	0	048		nothing	1
:*	:	058	*	042	
--	-	045	=	061	
BREAK		exits		exits	2
UA	UA	094	LA	095	3
Q	Q	081	Q	081	
W	W	087	W	087	
E	E	069	E	069	
R	R	082	R	082	
T	T	084	T	084	
Y	Y	089	Y	089	
U	U	085	U	085	
I	I	073	I	073	
O	O	079	O	079	
P	P	080	P	080	
@	@	064		nothing	4
LA	BkSp	008	NoCh	021	5
RA	NoCh	009]	093	6
DA	NoCh	010	[091	7
A	A	065	A	065	
S	S	083	S	083	
D	D	068	D	068	
F	F	070	F	070	
G	G	071	G	071	
H	H	072	H	072	
J	J	074	J	074	
K	K	075	K	075	
L	L	076	L	076	
;	;	059	+	043	
ENTER	CRLF	013	CRLF	013	8
CLEAR	NoCh	012	\	092	9
LSHFT		nothing		nothing	10
Z	Z	090	Z	090	
X	X	088	X	088	
C	C	067	C	067	

V	V	086	V	086	
B	B	066	B	066	
N	N	078	N	078	
M	M	077	M	077	
,<	,	044	<	060	
.>	.	046	>	062	
/?	/	047	?	063	
RSHFT	nothing		nothing		11
SPACEBAR	sp	032	sp	032	12

Notes:

1. **SHIFT-0**, which would normally toggle the upper/lower case keyboards, does not do so while **KEYCHECK.BAS** is running under “**245 POKE 282,1**”.
2. The **BREAK** key, shifted or unshifted, exits whatever program is running and returns to Command Level BASIC. If no program is running, it simply does a carriage return.
3. **UA** = Up Arrow, **LA** = Left Arrow.
4. **nothing** ==> no character is displayed, and no keycode is returned, I.e. no action occurs on the screen at all. **SHFT-@** pauses a running program until the spacebar is pressed.
5. **LA** = Left Arrow, **NoCh** ==> no character is displayed. **SHFT-LA** erases the current line.
6. **RA** = Right Arrow.
7. **DA** = Down Arrow.
8. **CRLF** = Carriage Return + Line Feed.
9. **CLEAR** clears the screen.
10. **LSHFT** = Left Shift key.
11. **RSHFT** = Right Shift key.
12. **sp** = Space.

=====

Appendix C: 64K CoCo 2 Key Codes

“Upper/Lower Case” Keyboard

Physical 64K CoCo 2 Keyboard:

```
[1!][2"][3#][4$][5%][6&][7'][8()][9)][ 0][:][*][-=][BR]
[UA][ Q][ W][ E][ R][ T][ Y][ U][ I][ O][ P][ @][LA][RA]
[DA][ A][ S][ D][ F][ G][ H][ J][ K][ L][;+][ENTR][CLR]
[SHFT][ Z][ X][ C][ V][ B][ N][ M][,<][.>][/?][SHFT]
[                               Space                               ]
```

BR = BREAK
 CLR = CLEAR
 ENTR = ENTER
 SHFT = SHIFT

UA = Up Arrow
 RA = Right Arrow
 DA = Down Arrow
 LA = Left Arrow

These are the characters and Key Codes returned by key presses on a physical 64K CoCo 2 keyboard. (See Appendix U for my physical 64K CoCo 2 setup).

They are from the keyboard after initial startup, followed by the **SHIFT-0** case toggle.

They also reflect line “245 POKE 282,0” in the **KEYCHECK.BAS** program (See Appendix T).

Alphabetic characters in this list which are enclosed in brackets are CoCo Reversed Characters, i.e. green-on-black. All other characters in this list are CoCo Standard Characters, i.e. black-on-green, except as noted.

KEY PRESS	UNSHIFTED		SHIFTED		NOTE
	CHAR	CODE	CHAR	CODE	
1!	1	049	!	033	
2"	2	050	"	034	
3#	3	051	#	035	
4\$	4	052	\$	036	
5%	5	053	%	037	
6&	6	054	&	038	
7'	7	055	'	039	
8(8	056	(040	

9)	9	057)	041	
0	0	048		nothing	1
:*	:	058	*	042	
--	-	045	=	061	
BREAK		exits	NoCh	003	2
UA	UA	094	LA	095	3
Q	[Q]	113	Q	081	
W	[W]	119	W	087	
E	[E]	101	E	069	
R	[R]	114	R	082	
T	[T]	116	T	084	
Y	[Y]	121	Y	089	
U	[U]	117	U	085	
I	[I]	105	I	073	
O	[O]	111	O	079	
P	[P]	112	P	080	
@	@	064		nothing	4
LA	BkSp	008	NoCh	021	5
RA	NoCh	009]	093	6
DA	NoCh	010	[091	7
A	[A]	097	A	065	
S	[S]	115	S	083	
D	[D]	100	D	068	
F	[F]	102	F	070	
G	[G]	103	G	071	
H	[H]	104	H	072	
J	[J]	106	J	074	
K	[K]	107	K	075	
L	[L]	108	L	076	
;	;	059	+	043	
ENTER	CRLF	013	CRLF	013	8
CLEAR	NoCh	012	\	092	9
LSHFT		nothing		nothing	10
Z	[Z]	122	Z	090	
X	[X]	120	X	088	
C	[C]	099	C	067	

V	[V]	118	V	086	
B	[B]	098	B	066	
N	[N]	110	N	078	
M	[M]	109	M	077	
,<	,	044	<	060	
.>	.	046	>	062	
/?	/	047	?	063	
RSHFT	nothing		nothing		11
SPACEBAR	sp	032	sp	032	12

Notes:

1. **SHIFT-0**, toggles the upper/lower case keyboards, even while **KEYCHECK.BAS** is running under "245 POKE 282,0".
2. The unshifted **BREAK** key exits whatever program is running and returns to Command Level BASIC. NoCh ==> no character is displayed. If no program is running, the **BREAK** key, shifted or unshifted, simply does a carriage return.
3. UA = Up Arrow, LA = Left Arrow.
4. **nothing** ==> no character is displayed, and no keycode is returned, I.e. no action occurs on the screen at all. **SHIFT-@** pauses a running program until the spacebar is pressed.
5. LA = Left Arrow. **SHIFT-LA** erases the current line.
6. RA = Right Arrow.
7. DA = Down Arrow.
8. CRLF = Carriage Return + Line Feed.
9. **CLEAR** clears the screen.
10. LSHIFT = Left Shift key.
11. RSHIFT = Right Shift key.
12. sp = Space.

=====

Appendix D: 64K CoCo 2 Keyboard Key Code Coverage

This chart shows, for each Key Code (000-127), which key press on a physical 64K CoCo 2 keyboard will return that code.

The key presses are as from the keyboard after initial startup, followed by the **SHIFT-0** case toggle.

They also reflect line “245 POKE 282,0” in the **KEYCHECK.BAS** program (See Appendix T).

Keys in this list which are enclosed in brackets are unshifted keys. Those without brackets are shifted keys.

As is clear from the following chart, the Key Code Coverage is not complete, i.e. there are several Key Codes which cannot be generated by a simple key press from the keyboard.

KEY CODE	KEY PRESS	KEY CODE	KEY PRESS
000		064	[@]
001		065	A
002		066	B
003	BRK	067	C
004		068	D
005		069	E
006		070	F
007		071	G
008	[LA]	072	H
009	[RA]	073	I
010	[DA]	074	J
011		075	K
012	[CLR]	076	L
013	[ENT] , ENT	077	M
014		078	N
015		079	O
016		080	P
017		081	Q
018		082	R
019		083	S
020		084	T
021	RA	085	U

022		086	V
023		087	W
024		088	X
025		089	Y
026		090	Z
027		091	DA
028		092	CLR
029		093	RA
030		094	[UA]
031		095	UA
032	[SP],SP	096	
033	1!	097	[A]
034	2"	098	[B]
035	3#	099	[C]
036	4\$	100	[D]
037	5%	101	[E]
038	6&	102	[F]
039	7'	103	[G]
040	8(104	[H]
041	9)	105	[I]
042	:*	106	[J]
043	;+	107	[K]
044	[,<]	108	[L]
045	[-=]	109	[M]
046	[.>]	110	[N]
047	[/?]	111	[O]
048	[0]	112	[P]
049	[1!]	113	[Q]
050	[2"]	114	[R]
051	[3#]	115	[S]
052	[4\$]	116	[T]
053	[5%]	117	[U]
054	[6&]	118	[V]
055	[7']	119	[W]
056	[8(]	120	[X]
057	[9)]	121	[Y]
058	[:*]	122	[Z]
059	[;+]	123	
060	,<	124	
061	-=	125	
062	.>	126	
063	/?	127	

In the above chart:

BRK = BREAK

CLR = CLEAR

ENT = ENTER

SP = SPACE

UA = Up Arrow

RA = Right Arrow

DA = Down Arrow

LA = Left Arrow

=====

Appendix E: CoCo 3 Key Codes

“All Upper Case” Keyboard

Physical CoCo 3 Keyboard:

```
[1!][2"][3#][4$][5%][6&][7'][8()][9)][ 0][:][*][ -=]      [BR]
[ALT][ Q][ W][ E][ R][ T][ Y][ U][ I][ O][ P][ @][CLR] [UA]
[CTRL][ A][ S][ D][ F][ G][ H][ J][ K][ L][;+][ENTR][LA][RA]
[SHFT][ Z][ X][ C][ V][ B][ N][ M][ ,<][.>][/?][SHFT] [DA]
[          Space          ]      [F1][F2]
```

BR = ESC BREAK
 ALT = ALT
 CLR = CLEAR
 ENTR = ENTER
 SHFT = SHIFT

UA = Up Arrow
 RA = Right Arrow
 DA = Down Arrow
 LA = Left Arrow
 F1 = Function 1
 F2 = Function 2

These are the characters and Key Codes returned by key presses on a physical 128K CoCo 3 keyboard (See Appendix V) and on a physical 512K CoCo 3 keyboard (See Appendix W).

They are from the keyboard as it is immediately after startup.

They also reflect line “245 POKE 282,1” in the **KEYCHECK.BAS** program (See Appendix T).

All characters in this list are CoCo Standard Characters, i.e. black-on-green, except as noted.

KEY PRESS	UNSHIFTED		SHIFTED		NOTE
	CHAR	CODE	CHAR	CODE	
1!	1	049	!	033	
2"	2	050	"	034	
3#	3	051	#	035	
4\$	4	052	\$	036	
5%	5	053	%	037	
6&	6	054	&	038	
7'	7	055	'	039	
8(8	056	(040	

9)	9	057)	041	
0	0	048		nothing	1
:*	:	058	*	042	
--	-	045	=	061	
BREAK		exits		exits	2
ALT	@	064		nothing	3
Q	Q	081	Q	081	
W	W	087	W	087	
E	E	069	E	069	
R	R	082	R	082	
T	T	084	T	084	
Y	Y	089	Y	089	
U	U	085	U	085	
I	I	073	I	073	
O	O	079	O	079	
P	P	080	P	080	
@	@	064		nothing	4
CLEAR	NoCh	012	\	092	5
UA	UA	094	LA	095	6
CTRL	Graphic	189	NoCh	001	
A	A	065	A	065	
S	S	083	S	083	
D	D	068	D	068	
F	F	070	F	070	
G	G	071	G	071	
H	H	072	H	072	
J	J	074	J	074	
K	K	075	K	075	
L	L	076	L	076	
;+	;	059	+	043	
ENTER	CRLF	013	CRLF	013	7
LA	BkSp	008	NoCh	021	8
RA	NoCh	009]	093	9
LSHFT	nothing		nothing		10
Z	Z	090	Z	090	
X	X	088	X	088	

C	C	067	C	067	
V	V	086	V	086	
B	B	066	B	066	
N	N	078	N	078	
M	M	077	M	077	
,<	,	044	<	060	
.>	.	046	>	062	
/?	/	047	?	063	
RSHFT	nothing		nothing		11
DA	NoCh	010	[091	12
SPACEBAR	sp	032	sp	032	13
F1	[G]	103	4	052	14
F2	NoCh	004	Graphic	214	

Notes:

1. **SHIFT-0**, which would normally toggle the upper/lower case keyboards, does not do so while **KEYCHECK.BAS** is running under “**245 POKE 282,1**”.
2. The **BREAK** key, shifted or unshifted, exits whatever program is running and returns to Command Level BASIC. If no program is running, it simply does a carriage return.
3. The shifted **ALT** key can sporadically produce a Code Point 19 or, more rarely, a Code Point 12.
4. **nothing** ==> no character is displayed, and no keycode is returned, I.e. no action occurs on the screen at all. **SHIFT-@** pauses a running program until the spacebar is pressed.
5. **CLEAR** clears the screen.
6. UA = Up Arrow, LA = Left Arrow.
7. CRLF = Carriage Return + Line Feed.
8. LA = Left Arrow, NoCh ==> no character is displayed. **SHIFT-LA** erases the current line.
9. RA = Right Arrow.
10. LSHFT = Left Shift key.
11. RSHFT = Right Shift key.
12. DA = Down Arrow.

13. sp = Space.

14. [**G**] = Reversed G, i.e. green-on-black. Hold **F1** down during power up to select alternate color set.

=====

Appendix F: CoCo 3 Key Codes “Upper/Lower Case” Keyboard

Physical CoCo 3 Keyboard:

```
[1!][2"][3#][4$][5%][6&][7'][8()][9)][ 0][:][*][ -=]      [BR]
[ALT][ Q][ W][ E][ R][ T][ Y][ U][ I][ O][ P][ @][CLR] [UA]
[CTRL][ A][ S][ D][ F][ G][ H][ J][ K][ L][;+][ENTR][LA][RA]
[SHFT][ Z][ X][ C][ V][ B][ N][ M][ ,<][.>][/?][SHFT] [DA]
[           Space           ]      [F1][F2]
```

```
BR = ESC BREAK
ALT = ALT
CLR = CLEAR
ENTR = ENTER
SHFT = SHIFT
```

```
UA = Up Arrow
RA = Right Arrow
DA = Down Arrow
LA = Left Arrow
F1 = Function 1
F2 = Function 2
```

These are the characters and Key Codes returned by key presses on a physical 128K CoCo 3 keyboard (See Appendix V) and on a physical 512K CoCo 3 keyboard (See Appendix W).

They are from the keyboard after initial startup, followed by the **SHIFT-0** case toggle.

They also reflect line “**245 POKE 282,0**” in the **KEYCHECK.BAS** program (See Appendix T).

Alphabetic characters in this list which are enclosed in brackets are CoCo Reversed Characters, i.e. green-on-black. All other characters in this list are CoCo Standard Characters, i.e. black-on-green, except as noted.

KEY PRESS	UNSHIFTED		SHIFTED		NOTE
	CHAR	CODE	CHAR	CODE	
1!	1	049	!	033	
2"	2	050	"	034	
3#	3	051	#	035	
4\$	4	052	\$	036	

5%	5	053	%	037	
6&	6	054	&	038	
7'	7	055	'	039	
8(8	056	(040	
9)	9	057)	041	
0	0	048	nothing		1
:*	:	058	*	042	
--	-	045	=	061	
BREAK		exits		exits	2
ALT	@	064	nothing		3
Q	[Q]	113	Q	081	
W	[W]	119	W	087	
E	[E]	101	E	069	
R	[R]	114	R	082	
T	[T]	116	T	084	
Y	[Y]	121	Y	089	
U	[U]	117	U	085	
I	[I]	105	I	073	
O	[O]	111	O	079	
P	[P]	112	P	080	
@	@	064	nothing		4
CLEAR	NoCh	012	\	092	5
UA	UA	094	LA	095	6
CTRL	Graphic	189	NoCh	001	
A	[A]	097	A	065	
S	[S]	115	S	083	
D	[D]	100	D	068	
F	[F]	102	F	070	
G	[G]	103	G	071	
H	[H]	104	H	072	
J	[J]	106	J	074	
K	[K]	107	K	075	
L	[L]	108	L	076	
;+	;	059	+	043	
ENTER	CRLF	013	CRLF	013	7
LA	BkSp	008	NoCh	021	8

RA	NoCh	009]	093	9
LSHFT	nothing		nothing		10
Z	[Z]	122	Z	090	
X	[X]	120	X	088	
C	[C]	099	C	067	
V	[V]	118	V	086	
B	[B]	098	B	066	
N	[N]	110	N	078	
M	[M]	109	M	077	
,<	,	044	<	060	
.>	.	046	>	062	
/?	/	047	?	063	
RSHFT	nothing		nothing		11
DA	NoCh	010	[091	12
SPACEBAR	sp	032	sp	032	13
F1	[G]	103	4	052	14
F2	NoCh	004	Graphic	214	

Notes:

1. **SHIFT-0**, which would normally toggle the upper/lower case keyboards, does not do so while **KEYCHECK.BAS** is running under “**245 POKE 282,1**”.
2. The unshifted **BREAK** key exits whatever program is running and returns to Command Level BASIC. NoCh ==> no character is displayed. If no program is running, the **BREAK** key, shifted or unshifted, simply does a carriage return.
3. The shifted **ALT** key can sporadically produce a Code Point 19 or, more rarely, a Code Point 12.
4. nothing ==> no character is displayed, and no keycode is returned, I.e. no action occurs on the screen at all. **SHFT-@** pauses a running program until the spacebar is pressed.
5. **CLEAR** clears the screen.
6. UA = Up Arrow, LA = Left Arrow.
7. CRLF = Carriage Return + Line Feed.
8. LA = Left Arrow, NoCh ==> no character is displayed. **SHFT-LA** erases the current line.
9. RA = Right Arrow.

10. LSHFT = Left Shift key.

11. RSHFT = Right Shift key.

12. DA = Down Arrow.

13. sp = Space.

14. [**G**] = Reversed G, i.e. green-on-black. Hold **F1** down during power up to select alternate color set.

=====

Appendix G: CoCo 3 Keyboard Key Code Coverage

This chart shows, for each Key Code (000-127), which key press on a physical CoCo 3 keyboard will return that code.

The key presses are as from the keyboard after initial startup, followed by the **SHIFT-0** case toggle.

They also reflect line “**245 POKE 282,0**” in the **KEYCHECK.BAS** program (See Appendix T).

Keys in this list which are enclosed in brackets are unshifted keys. Those without brackets are shifted keys.

As is clear from the following chart, the Key Code Coverage is not complete, i.e. there are several Key Codes which cannot be generated by a simple key press from the keyboard.

KEY CODE	KEY PRESS	KEY CODE	KEY PRESS
000		064	[@]
001	CTRL	065	A
002		066	B
003	BRK	067	C
004	[F2]	068	D
005		069	E
006		070	F
007		071	G
008	[LA]	072	H
009	[RA]	073	I
010	[DA]	074	J
011		075	K
012	[CLR]	076	L
013	[ENT] , ENT	077	M
014		078	N
015		079	O
016		080	P
017		081	Q
018		082	R
019		083	S
020		084	T
021	RA	085	U

022		086	V
023		087	W
024		088	X
025		089	Y
026		090	Z
027		091	DA
028		092	CLR
029		093	RA
030		094	[UA]
031		095	UA
032	[SP],SP	096	
033	1!	097	[A]
034	2"	098	[B]
035	3#	099	[C]
036	4\$	100	[D]
037	5%	101	[E]
038	6&	102	[F]
039	7'	103	F1,[G]
040	8(104	[H]
041	9)	105	[I]
042	:*	106	[J]
043	;+	107	[K]
044	[,<]	108	[L]
045	[-=]	109	[M]
046	[.>]	110	[N]
047	[/?]	111	[O]
048	[0]	112	[P]
049	[1!]	113	[Q]
050	[2"]	114	[R]
051	[3#]	115	[S]
052	[4\$]	116	[T]
053	[5%]	117	[U]
054	[6&]	118	[V]
055	[7']	119	[W]
056	[8(]	120	[X]
057	[9)]	121	[Y]
058	[:*]	122	[Z]
059	[;+]	123	
060	,<	124	
061	-=	125	
062	.>	126	
063	/?	127	

In the above chart:

BRK = BREAK

CLR = CLEAR

ENT = ENTER

SP = SPACE

UA = Up Arrow

RA = Right Arrow

DA = Down Arrow

LA = Left Arrow

=====

Appendix H: VCC CoCo (DECB) “All Upper Case” Keyboard Key Codes

VCC CoCo (DECB) Keyboard:

```
[ ] [F1][F2][ ] [ ] [RS][RGB][ ] [TH][PW][StB][FSc][ ] [ ] [ ] [ ] [ ]
[ ] [1!][2"][3#][4$][5%][6&][7'] [8()][9)][0 ] [[:*][-=] [BkSpc] [ ] [ ] [ ] [ ]
[ ] [ ] [ Q][ W][ E][ R][ T][ Y][ U][ I][ O][ P][ ] [ ] [ ] [ ] [ ] [ ]
[ Caps ] [ A][ S][ D][ F][ G][ H][ J][ K][ L][;:] [ ] [Enter]
[ Shift ] [ Z][ X][ C][ V][ B][ N][ M][,<][.>][/?] [Shift] [ ] [UpA]
[Cntl][ ] [ @ ] [ ] [Space] [ ] [ @ ] [ ] [ ] [Cntl] [LftA][DnA][RgtA]
```

VCC CoCo (DECB) Keyboard Number Pad Extension:

```
[ ] [ ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
[ ] [ ] [ ] [ ] [Clr][ ] [ ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
[UpA] [Brk][ ] [ ] [ ] [ ]
[LftA][DnA][RgtA] [ ] [ ] [ ] [ ]
```

BkSpc = Back Space
 Brk = BREAK (1End)
 Caps = CAPS LOCK
 Clr = CLEAR (7Home)
 Cntl = CNTL
 Enter = ENTER
 Shift = Shift

UpA = Up Arrow
 RgtA = Right Arrow
 DnA = Down Arrow
 LftA = Left Arrow

F1 = Function 1
 F2 = Function 2
 RS = Soft Reset (F5)
 RGB = RGB/Composite Toggle (F6)
 TH = Throttle Toggle (F8)
 PW = Hard Reset (F9)
 StB = Status Bar Toggle (F10)
 FSc = Full Screen/Windowed Toggle (F11)

These are the characters and Key Codes returned by key presses on a VCC Emulator PC Keyboard using the VCC’s CoCo (DECB) Keyboard selection.

The key presses were all made using a Victsing PC176B USB Wireless PC Keyboard. Where the key assignments shown above do not match those shown in (Pierce 37), the key presses were also tested and double-checked on the Unidentified (Made in China) Wired PC Keyboard. (See Appendix Q for PC Keyboard layouts). It should be noted

that the PC keyboards depicted in (Pierce 37) do not include number pads. This may be the cause of any differences noted herein.

These key presses are from the keyboard as it is immediately after startup.

They also reflect line "245 POKE 282,1" in the KEYCHECK.BAS program (See Appendix T).

All characters in this list are CoCo Standard Characters, i.e. black-on-green, except as noted. (NP) ==> a key on the PC keyboard's Number Pad.

KEY PRESS	UNSHIFTED		SHIFTED		NOTE
	CHAR	CODE	CHAR	CODE	
F1	[G]	103	4	052	1
F2	NoCh	004	Graphic	214	
RS	nothing		nothing		2
RGB	nothing		nothing		
TH	nothing		nothing		
PW	nothing		nothing		
StB	nothing		nothing		
FSc	nothing		nothing		
1!	1	049	!	033	
2"	2	050	"	034	
3#	3	051	#	035	
4\$	4	052	\$	036	
5%	5	053	%	037	
6&	6	054	&	038	
7'	7	055	'	039	
8(8	056	(040	
9)	9	057)	041	
0	0	048	nothing		3
:*	:	058	*	042	
--	-	045	=	061	
BkSpC	BkSp	008	NoCh	021	4
Q	Q	081	Q	081	
W	W	087	W	087	
E	E	069	E	069	
R	R	082	R	082	
T	T	084	T	084	

Y	Y	089	Y	089	
U	U	085	U	085	
I	I	073	I	073	
O	O	079	O	079	
P	P	080	P	080	
CLEAR (NP)	NoCh	012	\	092	5
CAPS	nothing		nothing		6
A	A	065	A	065	
S	S	083	S	083	
D	D	068	D	068	
F	F	070	F	070	
G	G	071	G	071	
H	H	072	H	072	
J	J	074	J	074	
K	K	075	K	075	
L	L	076	L	076	
;+	;	059	+	043	
ENTER	CRLF	013	CRLF	013	7
LSHFT	nothing		nothing		8
Z	Z	090	Z	090	
X	X	088	X	088	
C	C	067	C	067	
V	V	086	V	086	
B	B	066	B	066	
N	N	078	N	078	
M	M	077	M	077	
,<	,	044	<	060	
.>	.	046	>	062	
/?	/	047	?	063	
RSHFT	nothing		nothing		9
UpA	UpA	094	LftA	095	10
BREAK (NP)	exits		exits		11
LCTRL	Graphic	189	NoCh	001	
L@	@	064	nothing		12
SPACEBAR	sp	032	sp	032	13
R@	@	064	nothing		12,14
RCTRL	nothing		nothing		

LftA	BkSp	008	NoCh	021	4
DnA	NoCh	010	[091	15
RgtA	NoCh	009]	093	16

Notes:

1. **F1** key: [G] = Reversed G, i.e. green-on-black. Hold **F1** down during power up to select alternate color set.
2. **nothing** ==> no character is displayed, and no keycode is returned, I.e. no action occurs on the screen at all.
3. **SHFT-0**, which would normally toggle the upper/lower case keyboards, does not do so while **KEYCHECK.BAS** is running under “**245 POKE 282,1**”. Also see Note 6 below.
4. **BkSp** = **LftA** = Left Arrow, **NoCh** ==> no character is displayed. Both **SHFT-BkSp** and **SHFT-LftA** erases the current line. The **BkSp** and **LftA** keys do exactly the same thing.
5. **CLEAR** clears the screen.
6. The **CAPS** key, similar to many other keyboards, toggles between all uppercase mode and “lowercase”/uppercase mode. This is true whether the key press is shifted or unshifted. **SHFT-0** also accomplishes the same function - See Note 3 above.
7. **CRLF** = Carriage Return + Line Feed.
8. **LSHFT** = Left Shift key. Also see Note 2 above.
9. **RSHFT** = Right Shift key. Also see Note 2 above.
10. **UA** = Up Arrow, **LA** = Left Arrow.
11. The **BREAK** key, shifted or unshifted, exits whatever program is running and returns to Command Level BASIC. If no program is running, it simply does a carriage return.
12. **SHFT-@** pauses a running program until the spacebar is pressed. This works with either **SHFT** key and with either @ key in any combination.
13. **sp** = Space.
14. When held down, the **R@** (e.g. PC keyboard Right ALT key) produces the “@”, but can then be used in combination with the “F”, “E”, “C”, “H”, etc. Keys to access the VCC top menu items (File, Edit, Configuration, Help, etc.)

15. **DA** = Down Arrow.

16. **RA** = Right Arrow.

=====

Appendix I: VCC CoCo (DECB) “Upper/Lower Case” Keyboard Key Codes

VCC CoCo (DECB) Keyboard:

```
[ ] [F1][F2][ ] [ ] [RS][RGB][ ] [TH][PW][StB][FSc][ ] [ ] [ ] [ ] [ ]
[ ] [1!][2"][3#][4$][5%][6&][7'] [8()][9)][0 ] [[:*][-=] [BkSpc] [ ] [ ] [ ]
[ ] [ ] [ Q][ W][ E][ R][ T][ Y][ U][ I][ O][ P][ ] [ ] [ ] [ ] [ ]
[ Caps ] [ A][ S][ D][ F][ G][ H][ J][ K][ L][;:] [ ] [Enter]
[ Shift ] [ Z][ X][ C][ V][ B][ N][ M][,<][.>][/?] [Shift] [ ] [UpA]
[Cntl][ ] [ @ ] [ ] [Space] [ ] [ @ ] [ ] [ ] [Cntl] [LftA][DnA][RgtA]
```

VCC CoCo (DECB) Keyboard Number Pad Extension:

```
[ ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
[ ] [ ] [ ] [ ] [Clr][ ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
[UpA] [Brk][ ] [ ] [ ] [ ]
[LftA][DnA][RgtA] [ ] [ ] [ ] [ ]
```

BkSpc = Back Space
 Brk = BREAK (1End)
 Caps = CAPS LOCK
 Clr = CLEAR (7Home)
 Cntl = CNTL
 Enter = ENTER
 Shift = Shift

UpA = Up Arrow
 RgtA = Right Arrow
 DnA = Down Arrow
 LftA = Left Arrow

F1 = Function 1
 F2 = Function 2
 RS = Soft Reset (F5)
 RGB = RGB/Composite Toggle (F6)
 TH = Throttle Toggle (F8)
 PW = Hard Reset (F9)
 StB = Status Bar Toggle (F10)
 FSc = Full Screen/Windowed Toggle (F11)

These are the characters and Key Codes returned by key presses on a VCC Emulator PC keyboard using the VCC’s CoCo (DECB) Keyboard selection.

The key presses were all made using a Victsing PC176B USB Wireless PC Keyboard. Where the key assignments shown above do not match those shown in (Pierce 37), the key presses were also tested and double-checked on the Unidentified (Made in China) Wired PC Keyboard. (See Appendix Q for PC Keyboard layouts). It should be noted

that the PC keyboards depicted in (Pierce 37) do not include number pads. This may be the cause of any differences noted herein.

These key presses are from the keyboard after initial startup, followed by the **SHIFT-0** case toggle.

They also reflect line "245 POKE 282,0" in the **KEYCHECK.BAS** program (See Appendix T).

Alphabetic characters in this list which are enclosed in brackets are CoCo Reversed Characters, i.e. green-on-black. All other characters in this list are CoCo Standard Characters, i.e. black-on-green, except as noted. (NP) ==> a key on the PC keyboard's Number Pad.

KEY PRESS	UNSHIFTED		SHIFTED		NOTE
	CHAR	CODE	CHAR	CODE	
F1	[G]	103	4	052	1
F2	NoCh	004	Graphic	214	
RS	nothing		nothing		2
RGB	nothing		nothing		
TH	nothing		nothing		
PW	nothing		nothing		
StB	nothing		nothing		
FSc	nothing		nothing		
1!	1	049	!	033	
2"	2	050	"	034	
3#	3	051	#	035	
4\$	4	052	\$	036	
5%	5	053	%	037	
6&	6	054	&	038	
7'	7	055	'	039	
8(8	056	(040	
9)	9	057)	041	
0	0	048	nothing		3
:*	:	058	*	042	
--	-	045	=	061	
BkSpC	BkSp	008	NoCh	021	4
Q	[Q]	113	Q	081	
W	[W]	119	W	087	
E	[E]	101	E	069	

R	[R]	114	R	082	
T	[T]	116	T	084	
Y	[Y]	121	Y	089	
U	[U]	117	U	085	
I	[I]	105	I	073	
O	[O]	111	O	079	
P	[P]	112	P	080	
CLEAR (NP)	NoCh	012	\	092	5
CAPS	nothing		nothing		6
A	[A]	097	A	065	
S	[S]	115	S	083	
D	[D]	100	D	068	
F	[F]	102	F	070	
G	[G]	103	G	071	
H	[H]	104	H	072	
J	[J]	106	J	074	
K	[K]	107	K	075	
L	[L]	108	L	076	
;+	;	059	+	043	
ENTER	CRLF	013	CRLF	013	7
LSHFT	nothing		nothing		8
Z	[Z]	122	Z	090	
X	[X]	120	X	088	
C	[C]	099	C	067	
V	[V]	118	V	086	
B	[B]	098	B	066	
N	[N]	110	N	078	
M	[M]	109	M	077	
,<	,	044	<	060	
.>	.	046	>	062	
/?	/	047	?	063	
RSHFT	nothing		nothing		9
UpA	UpA	094	LftA	095	10
BREAK	exits		exits		11
LCTRL	Graphic	189	NoCh	001	
L@	@	064	nothing		12
SPACEBAR	sp	032	sp	032	13
R@	@	064	nothing		12,14

RCTRL	nothing	nothing	nothing	nothing	nothing
LftA	BkSp	008	NoCh	021	4
DnA	NoCh	010	[091	15
RgtA	NoCh	009]	093	16

Notes:

1. **F1** key: [G] = Reversed G, i.e. green-on-black. Hold **F1** down during power up to select alternate color set.
2. **nothing** ==> no character is displayed, and no keycode is returned, I.e. no action occurs on the screen at all.
3. **SHFT-0**, toggles the upper/lower case keyboards, even while **KEYCHECK.BAS** is running under “**245 POKE 282,0**”. Also see Note 6 below.
4. **BkSp** = **LftA** = Left Arrow, **NoCh** ==> no character is displayed. Both **SHFT-BkSp** and **SHFT-LftA** erases the current line. The **BkSp** and **LftA** keys do exactly the same thing.
5. **CLEAR** clears the screen.
6. The **CAPS** key, similar to many other keyboards, toggles between all uppercase mode and “lowercase”/uppercase mode. This is true whether the key press is shifted or unshifted. **SHFT-0** also accomplishes the same function - See Note 3 above.
7. **CRLF** = Carriage Return + Line Feed.
8. **LSHFT** = Left Shift key. Also see Note 2 above.
9. **RSHFT** = Right Shift key. Also see Note 2 above.
10. **UA** = Up Arrow, **LA** = Left Arrow.
11. The unshifted **BREAK** key exits whatever program is running and returns to Command Level BASIC. **NoCh** ==> no character is displayed. If no program is running, the **BREAK** key, shifted or unshifted, simply does a carriage return.
12. **SHFT-@** pauses a running program until the spacebar is pressed. This works with either **SHFT** key and with either @ key in any combination.
13. **sp** = Space.

14. When held down, the **R@** (e.g. PC keyboard Right ALT key) produces the “@”, but can then be used in combination with the “F”, “E”, “C”, “H”, etc. Keys to access the VCC top menu items (File, Edit, Configuration, Help, etc.)

15. **DA** = Down Arrow.

16. **RA** = Right Arrow.

=====

Appendix J: VCC CoCo (DECB) Keyboard Key Code Coverage

This chart shows, for each Key Code (000-127), which key press on a VCC Emulator PC Keyboard using the VCC's CoCo (DECB) Keyboard will return that code.

The key presses are as from the keyboard after initial startup, followed by the **SHIFT-0** case toggle.

They also reflect line "245 POKE 282,0" in the **KEYCHECK.BAS** program (See Appendix T).

Keys in this list which are enclosed in brackets are unshifted keys. Those without brackets are shifted keys.

As is clear from the following chart, the Key Code Coverage is not complete, i.e. there are several Key Codes which cannot be generated by a simple key press from the keyboard.

<u>KEY CODE</u>	<u>KEY PRESS</u>	<u>KEY CODE</u>	<u>KEY PRESS</u>
000		064	[@]
001	CTRL	065	A
002		066	B
003	BRK	067	C
004	[F2]	068	D
005		069	E
006		070	F
007		071	G
008	[LA]	072	H
009	[RA]	073	I
010	[DA]	074	J
011		075	K
012	[CLR]	076	L
013	[ENT], ENT	077	M
014		078	N
015		079	O
016		080	P
017		081	Q
018		082	R
019		083	S
020		084	T
021	RA	085	U

022		086	V
023		087	W
024		088	X
025		089	Y
026		090	Z
027		091	DA
028		092	CLR
029		093	RA
030		094	[UA]
031		095	UA
032	[SP],SP	096	
033	1!	097	[A]
034	2"	098	[B]
035	3#	099	[C]
036	4\$	100	[D]
037	5%	101	[E]
038	6&	102	[F]
039	7'	103	[G]
040	8(104	[H]
041	9)	105	[I]
042	:*	106	[J]
043	;+	107	[K]
044	[,<]	108	[L]
045	[-=]	109	[M]
046	[.>]	110	[N]
047	[/?]	111	[O]
048	[0]	112	[P]
049	[1!]	113	[Q]
050	[2"]	114	[R]
051	[3#]	115	[S]
052	[4\$]	116	[T]
053	[5%]	117	[U]
054	[6&]	118	[V]
055	[7']	119	[W]
056	[8(]	120	[X]
057	[9)]	121	[Y]
058	[:*]	122	[Z]
059	[;+]	123	
060	,<	124	
061	-=	125	
062	.>	126	
063	/?	127	

In the above chart:

BRK = BREAK

CLR = CLEAR

ENT = ENTER

SP = SPACE

UA = Up Arrow

RA = Right Arrow

DA = Down Arrow

LA = Left Arrow

=====

that the PC keyboards depicted in (Pierce 37) do not include number pads. This may be the cause of any differences noted herein.

These key presses are from the keyboard as it is immediately after startup.

They also reflect line "245 POKE 282,1" in the KEYCHECK.BAS program (See Appendix T).

All characters in this list are CoCo Standard Characters, i.e. black-on-green, except as noted. (NP) ==> a key on the PC keyboard's Number Pad.

KEY PRESS	UNSHIFTED		SHIFTED		NOTE
	CHAR	CODE	CHAR	CODE	
BREAK	exits		exits		1
F1	[G]	103	4	052	2
F2	NoCh	004	Graphic	214	
RS	nothing		nothing		3
RGB	nothing		nothing		
TH	nothing		nothing		
PW	nothing		nothing		
StB	nothing		nothing		
FSc	nothing		nothing		
F12	nothing		nothing		1
1!	1	049	!	033	
2@	2	050	@	064	
3#	3	051	#	035	
4\$	4	052	\$	036	
5%	5	053	%	037	
6	6	054	7	055	
7&	7	055	&	038	
8*	8	056	*	042	
9(9	057	(040	
0)	0	048)	041	4
-	-	045	Corrupt	189+045	5
=+	=	061	+	043	
BkSpC	BkSp	008	NoCh	021	6
Q	Q	081	Q	081	

W	W	087	W	087	
E	E	069	E	069	
R	R	082	R	082	
T	T	084	T	084	
Y	Y	089	Y	089	
U	U	085	U	085	
I	I	073	I	073	
O	O	079	O	079	
P	P	080	P	080	
,	Graphic	189	,	044	
.	Graphic	189	.	046	
\	Corrupt	189+047	Corrupt	049+189	7
CLEAR (NP)	NoCh	012	\	092	8
UpA (NP)	UpA	094	UpA	094	9
LftA (NP)	LftA	095	LftA	095	10
CAPS	nothing		nothing		4
A	A	065	A	065	
S	S	083	S	083	
D	D	068	D	068	
F	F	070	F	070	
G	G	071	G	071	
H	H	072	H	072	
J	J	074	J	074	
K	K	075	K	075	
L	L	076	L	076	
;+	;	059	+	043	
'"	'	039	"	034	
ENTER	CRLF	013	CRLF	013	11
BkSp (NP)	BkSp	008	BkSp	008	6
LSHFT	nothing		nothing		12
Z	Z	090	Z	090	
X	X	088	X	088	
C	C	067	C	067	
V	V	086	V	086	
B	B	066	B	066	
N	N	078	N	078	

M		M	077	M	077	
,<		,	044	<	060	
.>		.	046	>	062	
/?		/	047	?	063	
RSHFT		nothing		nothing		13
UpA		UpA	094	LftA	095	14
]	(NP)]	093]	093	15
[(NP)	[091	[091	16
LCTRL		Graphic	189	NoCh	001	
L@		@	064	nothing		17
SPACEBAR		sp	032	sp	032	18
RCTRL		nothing		nothing		
LftA		BkSp	008	NoCh	021	6
DnA		NoCh	010	[091	19
RgtA		NoCh	009]	093	20
=	(NP)	Corrupt	189+009	Corrupt	189+009	21
(.del)	(NP)	Corrupt	189+008	Corrupt	001+021	22

Notes:

1. The **BREAK** key, shifted or unshifted, exits whatever program is running and returns to Command Level BASIC. If no program is running, it simply does a carriage return. The **F12** key does the same thing.
2. **F1** key: [G] = Reversed G, i.e. green-on-black. Hold **F1** down during power up to select alternate color set.
3. **nothing** ==> no character is displayed, and no keycode is returned, i.e. no action occurs on the screen at all.
4. **SHFT-0**, which toggles the upper/lower case keyboards on a physical CoCo, does not do so on the Natural (OS-9) or Compact (OS-9) keyboards: use **SHFT-CAPS** instead. Note that unshifted **CAPS** does not accomplish this.
5. In **KEYCHECK.BAS**, the shifted - (minus sign) key often returns a graphic (Code Point 189) before returning the minus sign (Code Point 045). Outside of that program, it displays the two characters “=-”.
6. **BkSp** = **LftA** = Arrow Pad Left Arrow, **NoCh** ==> no character is displayed. Both **SHFT-BkSp** and **SHFT-LftA** on the Arrow Pad erase the current line. The **BkSp** and Arrow Pad **LftA**

keys do exactly the same thing. The Number Pad BkSpC key, on the other hand, does a plain backspace whether shifted or unshifted.

7. The “\ |” key is not a valid CoCo key. But it nevertheless is a key on the PC keyboard which has an effect on the VCC screen. Unshifted, it returns the Code Points 189 + 047 in **KEYCHECK.BAS** and displays the two characters “= /” on the screen when pressed outside of that program. Shifted, it returns the Code Points 049 + 189 in **KEYCHECK.BAS** and displays the two characters “1=” on the screen when pressed outside of that program: Frequently, it fails to return the 049 and just returns the 189 alone.
8. **CLEAR** clears the screen.
9. The Number Pad Up Arrow Key (8Up), unshifted or shifted, displays an Up Arrow (Code Point 094) on the screen.
10. The Number Pad Left Arrow Key (9PgUp), unshifted or shifted, displays an Left Arrow (Code Point 095) on the screen.
11. **CRLF** = Carriage Return + Line Feed.
12. **LSHFT** = Left Shift key. Also see Note 3 above.
13. **RSHFT** = Right Shift key. Also see Note 3 above.
14. **UpA** = Up Arrow, **LftA** = Left Arrow.
15. The “]” key on the Number Pad (1end) returns Code Point 093 in **KEYCHECK.BAS** and displays “]” on the screen whether shifted or unshifted.
16. The “[” key on the Number Pad (3PgDn) returns Code Point 091 in **KEYCHECK.BAS** and displays “[” on the screen whether shifted or unshifted.
17. **SHFT-@** pauses a running program until the spacebar is pressed. This works with either **SHFT** key and with either @ key in any combination.
18. **sp** = Space.
19. **DnA** = Down Arrow. The PC keyboard’s Number Pad (2Dn) key also returns Code Point 10; whether the key is shifted or unshifted..
20. **RgtA** = Right Arrow. The PC keyboard’s Number Pad (6Rt) key also returns Code Point 10; whether the key is shifted or unshifted
21. The “=” key on the Number Pad (0ins) returns Code Point 189+009 in **KEYCHECK.BAS** and displays “=” on the screen whether shifted or unshifted.

22. The PC keyboard's Number Pad (.del) key produces no action on the screen when unshifted, but clears the screen when shifted.

=====

Appendix L: VCC Natural (OS-9) “Upper/Lower Case” Keyboard Key Codes

VCC Natural (OS-9) Keyboard:

```
[Brk][F1][F2][  ][  ][RS][RGB][ ][TH][PW][StB][FSc][F12][  ][  ][  ][  ]
[  ][ 1!][2@][3#][4$][5%][6 ][7&][8*][9()][0][- ][=+] [BkSpc][  ][  ][  ]
[  ][  ][ Q][ W][ E][ R][ T][ Y][ U][ I][ O][ P][ ,][ .][ \][  ][  ][  ][  ]
[ Caps ][ A][ S][ D][ F][ G][ H][ J][ K][ L][;:]['"] [Enter]
[ Shift ][ Z][ X][ C][ V][ B][ N][ M][,<][.>][/?][ Shift ][  ][UpA]
[Cntl][  ][ @ ][  ][  ][  ][  ][  ][  ][  ][  ][Cntl][LftA][DnA][RgtA]
```

VCC Natural (OS-9) Keyboard Number Pad Extension:

```
[  ][  ][  ][  ]
[  ][  ][  ][  ][  ][  ][  ][  ]
[  ][  ][  ][  ][Clr][UpA][LftA][  ]
[  ][  ][  ][  ][BkSpc][  ][  ][  ]
[  ][  ][  ][  ][UpA][  ][  ][  ][  ][  ]
[LftA][DnA][RgtA][=  ][  ][  ][  ]
```

BkSpc = Back Space
 Brk = BREAK (1End)
 Caps = CAPS LOCK
 Clr = CLEAR (7Home)
 Cntl = CNTL
 Enter = ENTER
 Shift = Shift

UpA = Up Arrow
 RgtA = Right Arrow
 DnA = Down Arrow
 LftA = Left Arrow

F1 = Function 1
 F2 = Function 2
 RS = Soft Reset (F5)
 RGB = RGB/Composite Toggle (F6)
 TH = Throttle Toggle (F8)
 PW = Hard Reset (F9)
 StB = Status Bar Toggle (F10)
 FSc = Full Screen/Windowed Toggle (F11)

These are the characters and Key Codes returned by key presses on a VCC Emulator PC Keyboard using the VCC’s Natural (OS-9) Keyboard selection.

The key presses were all made using a Victsing PC176B USB Wireless PC Keyboard. Where the key assignments shown above do not match those shown in (Pierce 37), the key presses were also tested and double-checked on the Unidentified (Made in China) Wired PC Keyboard. (See Appendix Q for PC Keyboard layouts). It should be noted

that the PC keyboards depicted in (Pierce 37) do not include number pads. This may be the cause of any differences noted herein.

These key presses are from the keyboard after initial startup, followed by the **SHIFT-0** case toggle.

They also reflect line “245 POKE 282,0” in the **KEYCHECK.BAS** program (See Appendix T).

Alphabetic characters in this list which are enclosed in brackets are CoCo Reversed Characters, i.e. green-on-black. All other characters in this list are CoCo Standard Characters, i.e. black-on-green, except as noted. (NP) ==> a key on the PC keyboard’s Number Pad.

KEY PRESS	UNSHIFTED		SHIFTED		NOTE
	CHAR	CODE	CHAR	CODE	
BREAK		exits		exits	1
F1	[G]	103	4	052	2
F2	NoCh	004	Graphic	214	
RS		nothing		nothing	3
RGB		nothing		nothing	
TH		nothing		nothing	
PW		nothing		nothing	
StB		nothing		nothing	
FSc		nothing		nothing	
F12		nothing		nothing	1
1!	1	049	!	033	
2@	2	050	@	064	
3#	3	051	#	035	
4\$	4	052	\$	036	
5%	5	053	%	037	
6	6	054	7	055	
7&	7	055	&	038	
8*	8	056	*	042	
9(9	057	(040	
0)	0	048)	041	4
-	-	045	Corrupt	189+045	5
=+	=	061	+	043	
BkSpC	BkSp	008	NoCh	021	6
Q	[Q]	113	Q	081	

W		[W]	119	W	087	
E		[E]	101	E	069	
R		[R]	114	R	082	
T		[T]	116	T	084	
Y		[Y]	121	Y	089	
U		[U]	117	U	085	
I		[I]	105	I	073	
O		[O]	111	O	079	
P		[P]	112	P	080	
,		Graphic	189	,	044	
.		Graphic	189	.	046	
\		Corrupt	189+047	Corrupt	049+189	7
CLEAR	(NP)	NoCh	012	\	092	8
UpA	(NP)	UpA	094	UpA	094	9
LftA	(NP)	LftA	095	LftA	095	10
CAPS		nothing		nothing		4
A		[A]	097	A	065	
S		[S]	115	S	083	
D		[D]	100	D	068	
F		[F]	102	F	070	
G		[G]	103	G	071	
H		[H]	104	H	072	
J		[J]	106	J	074	
K		[K]	107	K	075	
L		[L]	108	L	076	
;+		;	059	+	043	
'"		'	039	"	034	
ENTER		CRLF	013	CRLF	013	11
BkSp	(NP)	BkSp	008	BkSp	008	6
LSHFT		nothing		nothing		12
Z		[Z]	122	Z	090	
X		[X]	120	X	088	
C		[C]	099	C	067	
V		[V]	118	V	086	
B		[B]	098	B	066	
N		[N]	110	N	078	

M	[M]	109	M	077	
,<	,	044	<	060	
.>	.	046	>	062	
/?	/	047	?	063	
RSHFT	nothing		nothing		13
UpA	UpA	094	LftA	095	14
] (NP)]	093]	093	15
[(NP)	[091	[091	16
LCTRL	Graphic	189	NoCh	001	
L@	@	064	nothing		17
SPACEBAR	sp	032	sp	032	18
RCTRL	nothing		nothing		
LftA	BkSp	008	NoCh	021	6
DnA	NoCh	010	[091	19
RgtA	NoCh	009]	093	20
= (NP)	Corrupt	189+009	Corrupt	189+009	21
(.del) (NP)	Corrupt	189+008	Corrupt	001+021	22

Notes:

1. The **BREAK** key, shifted or unshifted, exits whatever program is running and returns to Command Level BASIC. If no program is running, it simply does a carriage return. The **F12** key does the same thing.
2. **F1** key: [G] = Reversed G, i.e. green-on-black. Hold **F1** down during power up to select alternate color set.
3. **nothing** ==> no character is displayed, and no keycode is returned, i.e. no action occurs on the screen at all.
4. **SHFT-0**, which toggles the upper/lower case keyboards on a physical CoCo, does not do so on the Natural (OS-9) or Compact (OS-9) keyboards: use **SHFT-CAPS** instead. Note that unshifted **CAPS** does not accomplish this.
5. In **KEYCHECK.BAS**, the shifted - (minus sign) key often returns a graphic (Code Point 189) before returning the minus sign (Code Point 045). Outside of that program, it displays the two characters “=-”.
6. **BkSp** = **LftA** = Arrow Pad Left Arrow, **NoCh** ==> no character is displayed. Both **SHFT-BkSp** and **SHFT-LftA** on the Arrow Pad erase the current line. The **BkSp** and Arrow Pad **LftA**

keys do exactly the same thing. The Number Pad BkSpC key, on the other hand, does a plain backspace whether shifted or unshifted.

7. The “\ |” key is not a valid CoCo key. But it nevertheless is a key on the PC keyboard which has an effect on the VCC screen. Unshifted, it returns the Code Points 189 + 047 in **KEYCHECK.BAS** and displays the two characters “= /” on the screen when pressed outside of that program. Shifted, it returns the Code Points 049 + 189 in **KEYCHECK.BAS** and displays the two characters “1=” on the screen when pressed outside of that program: Frequently, it fails to return the 049 and just returns the 189 alone.
8. **CLEAR** clears the screen.
9. The Number Pad Up Arrow Key (8Up), unshifted or shifted, displays an Up Arrow (Code Point 094) on the screen.
10. The Number Pad Left Arrow Key (9PgUp), unshifted or shifted, displays an Left Arrow (Code Point 095) on the screen.
11. **CRLF** = Carriage Return + Line Feed.
12. **LSHFT** = Left Shift key. Also see Note 3 above.
13. **RSHFT** = Right Shift key. Also see Note 3 above.
14. **UpA** = Up Arrow, **LftA** = Left Arrow.
15. The “]” key on the Number Pad (1end) returns Code Point 093 in **KEYCHECK.BAS** and displays “]” on the screen whether shifted or unshifted.
16. The “[” key on the Number Pad (3PgDn) returns Code Point 091 in **KEYCHECK.BAS** and displays “[” on the screen whether shifted or unshifted.
17. **SHFT-@** pauses a running program until the spacebar is pressed. This works with either **SHFT** key and with either @ key in any combination.
18. **sp** = Space.
19. **DnA** = Down Arrow. The PC keyboard’s Number Pad (2Dn) key also returns Code Point 10; whether the key is shifted or unshifted..
20. **RgtA** = Right Arrow. The PC keyboard’s Number Pad (6Rt) key also returns Code Point 10; whether the key is shifted or unshifted
21. The “=” key on the Number Pad (0ins) returns Code Point 189+009 in **KEYCHECK.BAS** and displays “=” on the screen whether shifted or unshifted.

22. The PC keyboard's Number Pad (.del) key produces no action on the screen when unshifted, but clears the screen when shifted.

=====

Appendix M: VCC Natural (OS-9) Keyboard Key Code Coverage

This chart shows, for each Key Code (000-127), which key press on a VCC Emulator PC Keyboard using the VCC's Natural (OS-9) Keyboard will return that code.

The key presses are as from the keyboard after initial startup, followed by the **SHIFT-0** case toggle.

They also reflect line "245 POKE 282,0" in the **KEYCHECK.BAS** program (See Appendix T).

Keys in this list which are enclosed in brackets are unshifted keys. Those without brackets are shifted keys.

As is clear from the following chart, the Key Code Coverage is not complete, i.e. there are several Key Codes which cannot be generated by a simple key press from the keyboard.

KEY CODE	KEY PRESS	KEY CODE	KEY PRESS
000		064	[@]
001	CTRL	065	A
002		066	B
003	BRK	067	C
004	[F2]	068	D
005		069	E
006		070	F
007		071	G
008	[LA]	072	H
009	[RA]	073	I
010	[DA]	074	J
011		075	K
012	[CLR]	076	L
013	[ENT] , ENT	077	M
014		078	N
015		079	O
016		080	P
017		081	Q
018		082	R
019		083	S
020		084	T
021	RA	085	U

022		086	V
023		087	W
024		088	X
025		089	Y
026		090	Z
027		091	DA
028		092	CLR
029		093	RA
030		094	[UA]
031		095	UA
032	[SP],SP	096	
033	1!	097	[A]
034	2"	098	[B]
035	3#	099	[C]
036	4\$	100	[D]
037	5%	101	[E]
038	6&	102	[F]
039	7'	103	[G]
040	8(104	[H]
041	9)	105	[I]
042	:*	106	[J]
043	;+	107	[K]
044	[,<]	108	[L]
045	[-=]	109	[M]
046	[.>]	110	[N]
047	[/?]	111	[O]
048	[0]	112	[P]
049	[1!]	113	[Q]
050	[2"]	114	[R]
051	[3#]	115	[S]
052	[4\$]	116	[T]
053	[5%]	117	[U]
054	[6&]	118	[V]
055	[7']	119	[W]
056	[8(]	120	[X]
057	[9)]	121	[Y]
058	[:*]	122	[Z]
059	[;+]	123	
060	,<	124	
061	-=	125	
062	.>	126	
063	/?	127	

In the above chart:

BRK = BREAK

CLR = CLEAR

ENT = ENTER

SP = SPACE

UA = Up Arrow

RA = Right Arrow

DA = Down Arrow

LA = Left Arrow

=====

Appendix N: VCC Compact (OS-9) “All Upper Case” Keyboard Key Codes

VCC Compact (OS-9) Keyboard:

```
[ ] [F1][F2][ ] [ ] [RS][RGB][ ] [TH][PW][StB][FSc][ ] [ ] [ ] [ ] [ ]
[Brk][1!][2@][3#][4$][5%][6 ] [7&][8*][9()][0][ - ][=+] [BkSpc] [ ] [ ] [ ] [ ]
[Clr ] [ Q][ W][ E][ R][ T][ Y][ U][ I][ O][ P][ ,][ .][ \][ ] [ ] [ ] [ ] [ ]
[ Caps ] [ A][ S][ D][ F][ G][ H][ J][ K][ L][;:]['"] [Enter]
[ Shift ] [ Z][ X][ C][ V][ B][ N][ M][,<][.>][/?][ Shift ] [UpA]
[ = ] [ ] [ @ ] [ Space ] [ ] [ ] [ ] [ Cntl ] [LftA][DnA][RgtA]
```

VCC Compact (OS-9) Keyboard Number Pad Extension:

```
[ ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
[ ] [ ] [ ] [ ] [UpA][ ] [ ] [ ]
[ ] [ ] [ ] [ ] [BkSpc][ ] [ ] [ ]
[UpA] [ ] [ ] [ ] [ ] [ ]
[LftA][DnA][RgtA] [ ] [ ] [ ] [ ]
```

BkSpc = Back Space
Brk = BREAK (1End)
Caps = CAPS LOCK
Clr = CLEAR (7Home)
Cntl = CNTL
Enter = ENTER
Shift = Shift

UpA = Up Arrow
RgtA = Right Arrow
DnA = Down Arrow
LftA = Left Arrow

F1 = Function 1
F2 = Function 2
RS = Soft Reset (F5)
RGB = RGB/Composite Toggle (F6)
TH = Throttle Toggle (F8)
PW = Hard Reset (F9)
StB = Status Bar Toggle (F10)
FSc = Full Screen/Windowed Toggle (F11)

These are the characters and Key Codes returned by key presses on a VCC Emulator PC Keyboard using the VCC’s Compact (OS-9) Keyboard selection.

The key presses were all made using a Victsing PC176B USB Wireless PC Keyboard. Where the key assignments shown above do not match those shown in (Pierce 37), the key presses were also tested and double-checked on the Unidentified (Made in China) Wired PC Keyboard. (See Appendix Q for PC Keyboard layouts). It should be noted

that the PC keyboards depicted in (Pierce 37) do not include number pads. This may be the cause of any differences noted herein.

These key presses are from the keyboard as it is immediately after startup.

They also reflect line “245 POKE 282,1” in the KEYCHECK.BAS program (See Appendix T).

All characters in this list are CoCo Standard Characters, i.e. black-on-green, except as noted. (NP) ==> a key on the PC keyboard’s Number Pad.

KEY PRESS	UNSHIFTED		SHIFTED		NOTE
	CHAR	CODE	CHAR	CODE	
F1	[G]	103	4	052	1
F2	NoCh	004	Graphic	214	
RS	nothing		nothing		2
RGB	nothing		nothing		
TH	nothing		nothing		
PW	nothing		nothing		
StB	nothing		nothing		
FSc	nothing		nothing		
BREAK	exits		exits		3
1!	1	049	!	033	
2@	2	050	@	064	
3#	3	051	#	035	
4\$	4	052	\$	036	
5%	5	053	%	037	
6	6	054	7	055	
7&	7	055	&	038	
8*	8	056	*	042	
9(9	057	(040	
0)	0	048)	041	4
-	-	045	Corrupt	189+045	5
=+	=	061	+	043	
BkSp	BkSp	008	NoCh	021	6
CLEAR	NoCh	012	\	092	7
Q	Q	081	Q	081	

W		W	087	W	087	
E		E	069	E	069	
R		R	082	R	082	
T		T	084	T	084	
Y		Y	089	Y	089	
U		U	085	U	085	
I		I	073	I	073	
O		O	079	O	079	
P		P	080	P	080	
,		Graphic	189	,	044	
.		Graphic	189	.	046	
\		Corrupt	189+047	Corrupt	049+189	8
UpA	(NP)	UpA	094	4 UpA	094	9
CAPS		nothing		nothing		4
A		A	065	A	065	
S		S	083	S	083	
D		D	068	D	068	
F		F	070	F	070	
G		G	071	G	071	
H		H	072	H	072	
J		J	074	J	074	
K		K	075	K	075	
L		L	076	L	076	
;+		;	059	+	043	
"/		'	039	"	034	
ENTER		CRLF	013	CRLF	013	10
BkSpc	(NP)	BkSp	008	BkSp	008	6
LSHFT		nothing		nothing		11
Z		Z	090	Z	090	
X		X	088	X	088	
C		C	067	C	067	
V		V	086	V	086	
B		B	066	B	066	
N		N	078	N	078	
M		M	077	M	077	
,<		,	044	<	060	

.	>	.	046	>	062	
/	?	/	047	?	063	
RSHFT		nothing		nothing		12
UpA		UpA	094	LftA	095	13
=		Graphic	189	NoCh	001	14
L@		@	064	nothing		15
SPACEBAR		sp	032	sp	032	16
RCTRL		nothing		nothing		
LftA		BkSp	008	NoCh	021	6
DnA		NoCh	010	[091	17
RgtA		NoCh	009]	093	18

Notes:

1. **F1** key: [G] = Reversed G, i.e. green-on-black. Hold **F1** down during power up to select alternate color set.
2. **nothing** ==> no character is displayed, and no keycode is returned, I.e. no action occurs on the screen at all.
3. The **BREAK** key, shifted or unshifted, exits whatever program is running and returns to Command Level BASIC. If no program is running, it simply does a carriage return.
4. **SHFT-0**, which toggles the upper/lower case keyboards on a physical CoCo, does not do so on the Natural (OS-9) or Compact (OS-9) keyboards: use **SHFT-CAPS** instead. Note that unshifted **CAPS** does not accomplish this.
5. In **KEYCHECK.BAS**, the shifted - (minus sign) key often returns a graphic (Code Point 189) before returning the minus sign (Code Point 045). Outside of that program, it displays the two characters “=-”.
6. **BkSp** = **LftA** = Arrow Pad Left Arrow, **NoCh** ==> no character is displayed. Both **SHFT-BkSp** and **SHFT-LftA** on the Arrow Pad erase the current line. The **BkSp** and Arrow Pad **LftA** keys do exactly the same thing. The Number Pad **BkSp** key, on the other hand, does a plain backspace whether shifted or unshifted.
7. **CLEAR** clears the screen.
8. The “\|” key is not a valid CoCo key. But it nevertheless is a key on the PC keyboard which has an effect on the VCC screen. Unshifted, it returns the Code Points 189 + 047 in **KEYCHECK.BAS** and displays the two characters “= /” on the screen when pressed outside of that program. Shifted, it returns the Code Points 049 + 189 in **KEYCHECK.BAS** and displays the two

characters “**1=**” on the screen when pressed outside of that program: Frequently, it fails to return the 049 and just returns the 189 alone.

9. The Number Pad Up Arrow Key (8Up), unshifted or shifted, displays an Up Arrow (Code Point 094) on the screen.
10. **CRLF** = Carriage Return + Line Feed.
11. **LSHFT** = Left Shift key. Also see Note 2 above.
12. **RSHFT** = Right Shift key. Also see Note 2 above.
13. **UpA** = Up Arrow, **LftA** = Left Arrow.
14. The “=” key (The Left Control Key on the VCC CoCo and VCC Natural keyboards) displays an equals sign on the screen if unshifted. If shifted, it displays nothing.
15. **SHFT-@** pauses a running program until the spacebar is pressed. This works with either **SHFT** key and with either @ key in any combination.
16. **sp** = Space.
17. **DnA** = Down Arrow. The PC keyboard’s Number Pad (2Dn) key also returns Code Point 10; whether the key is shifted or unshifted..
18. **RgtA** = Right Arrow. The PC keyboard’s Number Pad (6Rt) key also returns Code Point 10; whether the key is shifted or unshifted

=====

Appendix O: VCC Compact (OS-9) “Upper/Lower Case” Keyboard Key Codes

VCC Compact (OS-9) Keyboard:

```
[ ] [F1][F2][ ] [ ] [RS][RGB][ ] [TH][PW][StB][FSc][ ] [ ] [ ] [ ] [ ]
[Brk][1!][2@][3#][4$][5%][6 ] [7&][8*][9()][0][ - ][=+] [BkSpc] [ ] [ ] [ ] [ ]
[Clr ] [ Q][ W][ E][ R][ T][ Y][ U][ I][ O][ P][ ,][ .][ \] [ ] [ ] [ ] [ ]
[ Caps ] [ A][ S][ D][ F][ G][ H][ J][ K][ L][;:]['"] [Enter]
[ Shift ] [ Z][ X][ C][ V][ B][ N][ M][,<][.>][/?][ Shift ] [UpA]
[ = ] [ ] [ @ ] [ Space ] [ ] [ ] [ ] [ Cntl ] [LftA][DnA][RgtA]
```

VCC Compact (OS-9) Keyboard Number Pad Extension:

```
[ ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
[ ] [ ] [ ] [ ] [UpA][ ] [ ] [ ]
[ ] [ ] [ ] [ ] [BkSpc][ ] [ ] [ ]
[UpA] [ ] [ ] [ ] [ ] [ ]
[LftA][DnA][RgtA] [ ] [ ] [ ] [ ]
```

BkSpc = Back Space
Brk = BREAK (1End)
Caps = CAPS LOCK
Clr = CLEAR (7Home)
Cntl = CNTL
Enter = ENTER
Shift = Shift

UpA = Up Arrow
RgtA = Right Arrow
DnA = Down Arrow
LftA = Left Arrow

F1 = Function 1
F2 = Function 2
RS = Soft Reset (F5)
RGB = RGB/Composite Toggle (F6)
TH = Throttle Toggle (F8)
PW = Hard Reset (F9)
StB = Status Bar Toggle (F10)
FSc = Full Screen/Windowed Toggle (F11)

These are the characters and Key Codes returned by key presses on a VCC Emulator PC Keyboard using the VCC’s Compact (OS-9) Keyboard selection.

The key presses were all made using a Victsing PC176B USB Wireless PC Keyboard. Where the key assignments shown above do not match those shown in (Pierce 37), the key presses were also tested and double-checked on the Unidentified (Made in China) Wired PC Keyboard. (See Appendix Q for PC Keyboard layouts). It should be noted

that the PC keyboards depicted in (Pierce 37) do not include number pads. This may be the cause of any differences noted herein.

These key presses are from the keyboard after initial startup, followed by the **SHIFT-0** case toggle.

They also reflect line “245 POKE 282,0” in the **KEYCHECK.BAS** program (See Appendix T).

Alphabetic characters in this list which are enclosed in brackets are CoCo Reversed Characters, i.e. green-on-black. All other characters in this list are CoCo Standard Characters, i.e. black-on-green, except as noted. (NP) ==> a key on the PC keyboard’s Number Pad.

KEY PRESS	UNSHIFTED		SHIFTED		NOTE
	CHAR	CODE	CHAR	CODE	
F1	[G]	103	4	052	1
F2	NoCh	004	Graphic	214	
RS	nothing		nothing		2
RGB	nothing		nothing		
TH	nothing		nothing		
PW	nothing		nothing		
StB	nothing		nothing		
FSc	nothing		nothing		
BREAK		exits		exits	3
1!	1	049	!	033	
2@	2	050	@	064	
3#	3	051	#	035	
4\$	4	052	\$	036	
5%	5	053	%	037	
6	6	054	7	055	
7&	7	055	&	038	
8*	8	056	*	042	
9(9	057	(040	
0)	0	048)	041	4
-	-	045	Corrupt	189+045	5
=+	=	061	+	043	
BkSpC	BkSp	008	NoCh	021	6
CLEAR	NoCh	012	\	092	7
Q	[Q]	113	Q	081	

W		[W]	119	W	087	
E		[E]	101	E	069	
R		[R]	114	R	082	
T		[T]	116	T	084	
Y		[Y]	121	Y	089	
U		[U]	117	U	085	
I		[I]	105	I	073	
O		[O]	111	O	079	
P		[P]	112	P	080	
,		Graphic	189	,	044	
.		Graphic	189	.	046	
\		Corrupt	189+047	Corrupt	049+189	8
UpA	(NP)	UpA	094	4 UpA	094	9
CAPS		nothing		nothing		4
A		[A]	097	A	065	
S		[S]	115	S	083	
D		[D]	100	D	068	
F		[F]	102	F	070	
G		[G]	103	G	071	
H		[H]	104	H	072	
J		[J]	106	J	074	
K		[K]	107	K	075	
L		[L]	108	L	076	
;+		;	059	+	043	
		'	039	"	034	
ENTER		CRLF	013	CRLF	013	10
BkSpc	(NP)	BkSp	008	BkSp	008	6
LSHFT		nothing		nothing		11
Z		[Z]	122	Z	090	
X		[X]	120	X	088	
C		[C]	099	C	067	
V		[V]	118	V	086	
B		[B]	098	B	066	
N		[N]	110	N	078	
M		[M]	109	M	077	
,<		,	044	<	060	

.>	.	046	>	062	
/?	/	047	?	063	
RSHFT	nothing		nothing		12
UpA	UpA	094	LftA	095	13
=	Graphic	189	NoCh	001	14
L@	@	064	nothing		15
SPACEBAR	sp	032	sp	032	16
RCTRL	nothing		nothing		
LftA	BkSp	008	NoCh	021	6
DnA	NoCh	010	[091	17
RgtA	NoCh	009]	093	18

Notes:

1. **F1** key: [G] = Reversed G, i.e. green-on-black. Hold **F1** down during power up to select alternate color set.
2. `nothing` ==> no character is displayed, and no keycode is returned, I.e. no action occurs on the screen at all.
3. The **BREAK** key, shifted or unshifted, exits whatever program is running and returns to Command Level BASIC. If no program is running, it simply does a carriage return.
4. **SHFT-0**, which toggles the upper/lower case keyboards on a physical CoCo, does not do so on the Natural (OS-9) or Compact (OS-9) keyboards: use **SHFT-CAPS** instead. Note that unshifted **CAPS** does not accomplish this.
5. In **KEYCHECK.BAS**, the shifted - (minus sign) key often returns a graphic (Code Point 189) before returning the minus sign (Code Point 045). Outside of that program, it displays the two characters “=-”.
6. **BkSp** = **LftA** = Arrow Pad Left Arrow, **NoCh** ==> no character is displayed. Both **SHFT-BkSp** and **SHFT-LftA** on the Arrow Pad erase the current line. The **BkSp** and Arrow Pad **LftA** keys do exactly the same thing. The Number Pad **BkSp** key, on the other hand, does a plain backspace whether shifted or unshifted.
7. **CLEAR** clears the screen.
8. The “\|” key is not a valid CoCo key. But it nevertheless is a key on the PC keyboard which has an effect on the VCC screen. Unshifted, it returns the Code Points 189 + 047 in **KEYCHECK.BAS** and displays the two characters “= /” on the screen when pressed outside of that program. Shifted, it returns the Code Points 049 + 189 in **KEYCHECK.BAS** and displays the two

characters “**1=**” on the screen when pressed outside of that program: Frequently, it fails to return the 049 and just returns the 189 alone.

9. The Number Pad Up Arrow Key (8Up), unshifted or shifted, displays an Up Arrow (Code Point 094) on the screen.
10. **CRLF** = Carriage Return + Line Feed.
11. **LSHFT** = Left Shift key. Also see Note 2 above.
12. **RSHFT** = Right Shift key. Also see Note 2 above.
13. **UpA** = Up Arrow, **LftA** = Left Arrow.
14. The “=” key (The Left Control Key on the VCC CoCo and VCC Natural keyboards) displays an equals sign on the screen if unshifted. If shifted, it displays nothing.
15. **SHFT-@** pauses a running program until the spacebar is pressed. This works with either **SHFT** key and with either @ key in any combination.
16. **sp** = Space.
17. **DnA** = Down Arrow. The PC keyboard’s Number Pad (2Dn) key also returns Code Point 10; whether the key is shifted or unshifted..
18. **RgtA** = Right Arrow. The PC keyboard’s Number Pad (6Rt) key also returns Code Point 10; whether the key is shifted or unshifted

=====

Appendix P: VCC Compact (OS-9) Keyboard Key Code Coverage

This chart shows, for each Key Code (000-127), which key press on a VCC Emulator PC Keyboard using the VCC's Compact (OS-9) Keyboard will return that code.

The key presses are as from the keyboard after initial startup, followed by the **SHIFT-0** case toggle.

They also reflect line "245 POKE 282,0" in the **KEYCHECK.BAS** program (See Appendix T).

Keys in this list which are enclosed in brackets are unshifted keys. Those without brackets are shifted keys.

As is clear from the following chart, the Key Code Coverage is not complete, i.e. there are several Key Codes which cannot be generated by a simple key press from the keyboard.

KEY CODE	KEY PRESS	KEY CODE	KEY PRESS
000		064	[@]
001	CTRL	065	A
002		066	B
003	BRK	067	C
004	[F2]	068	D
005		069	E
006		070	F
007		071	G
008	[LA]	072	H
009	[RA]	073	I
010	[DA]	074	J
011		075	K
012	[CLR]	076	L
013	[ENT] , ENT	077	M
014		078	N
015		079	O
016		080	P
017		081	Q
018		082	R
019		083	S
020		084	T
021	RA	085	U

022		086	V
023		087	W
024		088	X
025		089	Y
026		090	Z
027		091	DA
028		092	CLR
029		093	RA
030		094	[UA]
031		095	UA
032	[SP],SP	096	
033	1!	097	[A]
034	2"	098	[B]
035	3#	099	[C]
036	4\$	100	[D]
037	5%	101	[E]
038	6&	102	[F]
039	7'	103	[G]
040	8(104	[H]
041	9)	105	[I]
042	:*	106	[J]
043	;+	107	[K]
044	[,<]	108	[L]
045	[-=]	109	[M]
046	[.>]	110	[N]
047	[/?]	111	[O]
048	[0]	112	[P]
049	[1!]	113	[Q]
050	[2"]	114	[R]
051	[3#]	115	[S]
052	[4\$]	116	[T]
053	[5%]	117	[U]
054	[6&]	118	[V]
055	[7']	119	[W]
056	[8(]	120	[X]
057	[9)]	121	[Y]
058	[:*]	122	[Z]
059	[;+]	123	
060	,<	124	
061	-=	125	
062	.>	126	
063	/?	127	

In the above chart:

BRK = BREAK

CLR = CLEAR

ENT = ENTER

SP = SPACE

UA = Up Arrow

RA = Right Arrow

DA = Down Arrow

LA = Left Arrow

=====

Appendix Q: Various PC Keyboards (For Reference Only)

PC Keyboards all look similar to one another at first glance. But they do often have subtle differences. And, so, their use with the VCC Emulator may vary slightly as well. I present four representative examples here.

The point of this Appendix is to warn the user that their experience with the material of this paper may vary a bit, depending upon which specific PC keyboard they are using. Some minor investigation on their part may be necessary to explore how their particular PC keyboard works with the VCC Emulator.

Since I'm using the Victsing throughout this paper, I'll use it as my reference keyboard here as well. Note the `Clc` key to the right of the `F12` key. This key just acts to pop up the system calculator. The `Clc` key is not found on most PC keyboards.

Note what appears to be two `+` keys and two `ENT` keys on the right side of the Number Pad. On the actual keyboard, the two `+` keys are actually one double-sized vertical key. And, the two `ENT` keys are actually one double-sized vertical key. This arrangement is common on most PC keyboards.

Victsing PC176B USB Wireless PC Keyboard:

```
[Esc][F1][F2][F3][F4][F5][F6][F7][F8][F9][F10][F11][F12][Clc] [Prnt][Scr][Paus]
[~`] [1!][2@][3#][4$][5%][6^][7&][8*][9()][0][_][=+] [BkSpc] [Inst][Hom][PgUp]
[Tab ] [ Q][ W][ E][ R][ T][ Y][ U][ I][ O][ P][[{}][\] [Dlet][End][PgDn]
[Caps ] [ A][ S][ D][ F][ G][ H][ J][ K][ L][;:]['" ] [Enter]
[ Shift ] [ Z][ X][ C][ V][ B][ N][ M][,<][.>][/?][ Shift ] [UpA]
[Cntl][Win][LAlt][ Space ] [RAlt][Fn][Prp][Cntl] [LftA][DnA][RgtA]
```

Victsing PC176B USB Wireless PC Keyboard Number Pad Extension:

```
[Prnt][Scr][Paus]
[Inst][Hom][PgUp] [NLk][ / ][ * ][ - ]
[Dlet][End][PgDn] [7Hm][8Up][9PU][ + ]
[4Lt][ 5 ][6Rt][ + ]
[UpA] [1En][2Dn][3PD][ENT]
[LftA][DnA][RgtA] [ 0Ins ][.Dl][ENT]
```


The following keyboard shows a second **Win** key in place of the **Fn** key. It has no **Clc** key and separates the Function keys into three groups of four each.

The diagram below appears to show two **Enter** keys below the **BkSpc** key. On the actual keyboard, this is actually just one large **Enter** key. Note also that the **\|** key has been displaced from the third row of keys to the fourth row, to make room for the larger **Enter** key.

Unidentified (Made in China) USB Wired PC Keyboard:

```
[Esc] [F1][F2][F3][F4] [F5][F6][F7][F8] [F9][F10][F11][F12] [Prnt][Scr][Paus]
[~`] [1!][2@][3#][4$][5%][6^][7&][8*][9()][0][_][=+] [BkSpC] [Inst][Hom][PgUp]
[Tab][ Q][ W][ E][ R][ T][ Y][ U][ I][ O][ P][[{}]] [Enter] [Dlet][End][PgDn]
[Caps][ A][ S][ D][ F][ G][ H][ J][ K][ L][;:]['"] [Enter]
[ Shift ] [ Z][ X][ C][ V][ B][ N][ M][,<][.>][/?] [ Shift ] [UpA]
[Cntl][Win][LAlt][ Space ][RAlt][Win][Prp][Cntl] [LftA][DnA][RgtA]
```

Unidentified (Made in China) USB Wired PC Keyboard Number Pad Extension:

```
[Prnt][Scr][Paus]
[Inst][Hom][PgUp] [NLk][ / ][ * ][ - ]
[Dlet][End][PgDn] [7Hm][8Up][9PU][ + ]
[4Lt][ 5 ][6Rt][ + ]
[UpA] [1En][2Dn][3PD][ENT]
[LftA][DnA][RgtA] [ 0Ins ][.Dl][ENT]
```

This keyboard shows the three sets of four Function keys and no **Clc** key, It has the single-sized **Enter** key and the **Fn** key instead of the second **Win** key.

Unidentified (Made in China) PC230A USB Wireless PC Keyboard:

```
[Esc] [F1][F2][F3][F4] [F5][F6][F7][F8] [F9][F10][F11][F12] [Prnt][Scr][Paus]
[~`] [1!][2@][3#][4$][5%][6^][7&][8*][9()][0][_][=+] [BkSpC] [Inst][Hom][PgUp]
[Tab ] [ Q][ W][ E][ R][ T][ Y][ U][ I][ O][ P][[{}]] [ \| ] [Dlet][End][PgDn]
[ Caps ] [ A][ S][ D][ F][ G][ H][ J][ K][ L][;:]['"] [Enter]
[ Shift ] [ Z][ X][ C][ V][ B][ N][ M][,<][.>][/?] [ Shift ] [UpA]
[Cntl][Win][LAlt][ Space ][RAlt][Fn][Prp][Cntl] [LftA][DnA][RgtA]
```

Unidentified (Made in China) PC230A USB Wireless PC Keyboard Number Pad Extension:

```
[Prnt][Scr][Paus]
[Inst][Hom][PgUp] [NLk][ / ][ * ][ - ]
[Dlet][End][PgDn] [7Hm][8Up][9PU][ + ]
[4Lt][ 5 ][6Rt][ + ]
[UpA] [1En][2Dn][3PD][ENT]
[LftA][DnA][RgtA] [ 0Ins ][.Dl][ENT]
```

This Dell keyboard shows the three sets of four Function keys and no Clc key, It has the single-sized Enter key and no Fn or right side Win key at all.

It also shows the Arrows keypad shifted to the left, and the six Hom etc. Keys in two rows of three instead of in three rows of two.

Dell SK-8125 USB Wired PC Keyboard:

```
[Esc] [F1][F2][F3][F4] [F5][F6][F7][F8] [F9][F10][F11][F12]
[Prnt][Scr][Paus]
[~`] [1!][2@][3#[4$][5%][6^][7&][8*][9()][0][_][=+] [BkSpC] [Hom][PgUp]
[Tab ] [ Q][ W][ E][ R][ T][ Y][ U][ I][ O][ P][[{}][\|] [End][PgDn]
[ Caps ] [ A][ S][ D][ F][ G][ H][ J][ K][ L][;:'"] [Enter] [Dlet][Inst]
[ Shift ] [ Z][ X][ C][ V][ B][ N][ M][,<][.>][/?] [ Shift ] [UpA]
[Cntl][Win][LAlt][ Space ][RAlt][Prp][Cntl] [LftA][DnA][RgtA]
```

Dell SK-8125 USB Wired PC Keyboard Number Pad Extension:

```
[Prnt][Scr][Paus]
[Hom][PgUp] [NLk][ / ][ * ][ - ]
[End][PgDn] [7Hm][8Up][9PU][ + ]
[Dlet][Inst] [4Lt][ 5 ][6Rt][ + ]
[UpA] [1En][2Dn][3PD][ENT]
tA][DnA][RgtA] [ 0Ins ][.Dl][ENT]
```

=====

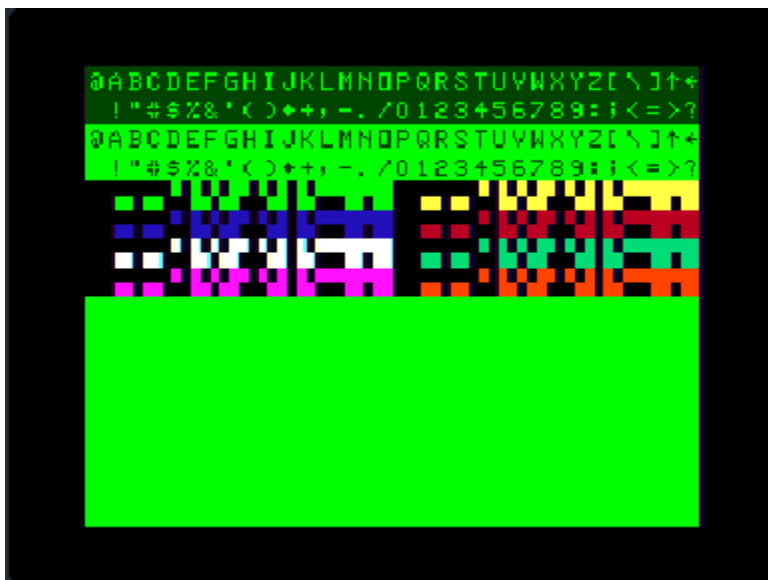
Appendix R: Extended ASCII (128-255)

Although this paper is primarily concerned with the ASCII code points (0-127); for interest and reference, code points 128-255 are also presented and discussed in this Appendix.

The so-called “Extended ASCII” covers Code Points 128-255. There is no Extended ASCII standard, and the Symbols associated with these Code Points are system dependent. They can, and do, vary widely from system to system, and are generally mutually incompatible from system-to-system.

The CoCo provides a set of block-graphics (aka “Semigraphics”) character symbols to cover these Code Points. The block character displayed is the same whether it is displayed via the POKE mechanism or via the PRINT mechanism (cf. Appendix A).

Nine colors are used to form these Semigraphics characters. Colors are perceived differently by different people — and also apparently by different computers and different monitors. Here are the nine colors as shown on the XRoar CoCo Emulator:



And here are the same nine colors as they appear on the VCC Emulator.



With both of these emulators, these are the colors as they would appear on the original CoCo composite monitor.

The nine “Official” CoCo2 Semigraphic colors are:

CLS0 = Black	(Black)
CLS1 = Green	(Green)
CLS2 = Yellow	(Red)
CLS3 = Blue	(Light Blue)
CLS4 = Red	(Gray)
CLS5 = Buff	(White)
CLS6 = Cyan	(Aqua)
CLS7 = Magenta	(Dark Blue)
CLS8 = Orange	(Orange)

The official colors are on the left in this list, and they represent the colors as they would appear on a composite monitor. The colors on the right (in parentheses) are the colors that would appear on a more “modern” RGB monitor. Here are the nine colors as they appear on the VCC Emulator’s RGB monitor.



For the remainder of this appendix, I will use the RGB colors since I consider them to be more crisp and distinct; because I suspect RGB monitors are easier to find than composite monitors these days, and because RGB is the default in VCC 2.1.0d.

The original IBM PC used a set of 255 symbols which was then known as DOS Latin. This set is now known as Code Page 437.


For each of Code Points 128-255, the following entries show:

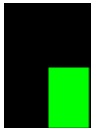
1. The first box is a representation of the block graphic character symbol itself. It's not an exact representation, because the actual characters do not include the black outline and sector borders. These borders are useful, however, for clarity in this context.
2. The second box is provided as an aid for those who have difficulty discerning the different colors, or for when this paper is printed in black-and-white. Each sector color is represented by a letter, as follows:
 - B = Black
 - G = Green
 - R = Red
 - L = Light Blue
 - Y = Gray
 - W = White
 - A = Aqua
 - D = Dark Blue
 - O = Orange
3. For reference, how Code Page 437 handled that Code Point.

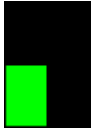
All Character Codes (128-255) are presented in decimal format. See Appendix S for decimal to hexadecimal conversions.

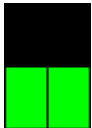
The CP437 (Code Page 437) Codes and Symbols are taken from:

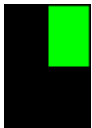
https://en.wikipedia.org/wiki/Code_page_437 ,
and are as produced by PC BASIC.

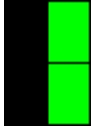
Code 128		<table border="1"><tr><td>B</td><td>B</td></tr><tr><td>B</td><td>B</td></tr></table>	B	B	B	B	CP437 = Ç
B	B						
B	B						

Code 129		<table border="1"><tr><td>B</td><td>B</td></tr><tr><td>B</td><td>G</td></tr></table>	B	B	B	G	CP437 = ü
B	B						
B	G						

Code 130		<table border="1"><tr><td>B</td><td>B</td></tr><tr><td>G</td><td>B</td></tr></table>	B	B	G	B	CP437 = é
B	B						
G	B						

Code 131		<table border="1"><tr><td>B</td><td>B</td></tr><tr><td>G</td><td>G</td></tr></table>	B	B	G	G	CP437 = â
B	B						
G	G						

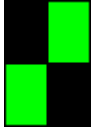
Code 132		<table border="1"><tr><td>B</td><td>G</td></tr><tr><td>B</td><td>B</td></tr></table>	B	G	B	B	CP437 = ä
B	G						
B	B						

Code 133 

B	G
B	G

 CP437 = à

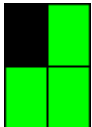
—

Code 134 

B	G
G	B

 CP437 = å

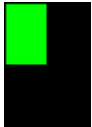
—

Code 135 

B	G
G	G

 CP437 = ç

—

Code 136 

G	B
B	B

 CP437 = ê

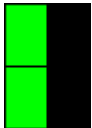
—

Code 137 

G	B
B	G

 CP437 = ë

—

Code 138 

G	B
G	B

 CP437 = è

—

Code 139 

G	B
G	G

 CP437 = ï

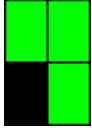
—

Code 140 

G	G
B	B

 CP437 = $\hat{\mathbf{i}}$

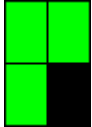
—

Code 141 

G	G
B	G

 CP437 = $\hat{\mathbf{i}}$

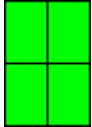
—

Code 142 

G	G
G	B

 CP437 = $\ddot{\mathbf{A}}$


—

Code 143 

G	G
G	G

 CP437 = $\mathring{\mathbf{A}}$


—

Code 144 

B	B
B	B

 CP437 = $\acute{\mathbf{E}}$


—

Code 145 

B	B
B	R

 CP437 = $\mathfrak{a}\mathfrak{e}$

—

Code 146 

B	B
R	B

 CP437 = Æ

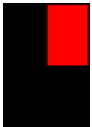
—

Code 147 

B	B
R	R

 CP437 = Ô

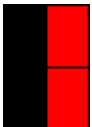
—

Code 148 

B	R
B	B

 CP437 = Ö

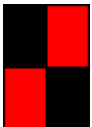
—

Code 149 

B	R
B	R

 CP437 = Ò

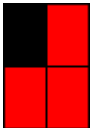
—

Code 150 

B	R
R	B

 CP437 = û


—

Code 151 

B	R
R	R

 CP437 = ù

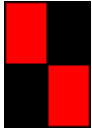
—

Code 152 

R	B
B	B

 CP437 = ŷ

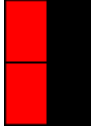
—

Code 153 

R	B
B	R

 CP437 = Ö

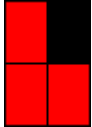
—

Code 154 

R	B
R	B

 CP437 = Ü

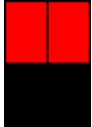
—

Code 155 

R	B
R	R

 CP437 = Ç

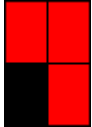
—

Code 156 

R	R
B	B

 CP437 = £

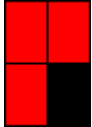
—

Code 157 

R	R
B	R

 CP437 = ¥

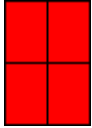
—

Code 158 

R	R
R	B

 CP437 = ₺


—

Code 159 

R	R
R	R

 CP437 = *f*

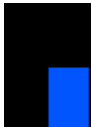
—

Code 160 

B	B
B	B

 CP437 = *á*

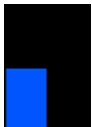
—

Code 161 

B	B
B	L

 CP437 = *í*

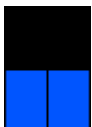
—

Code 162 

B	B
L	B

 CP437 = *ó*


—

Code 163 

B	B
L	L

 CP437 = *ú*

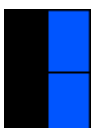
—

Code 164 

B	L
B	B

 CP437 = *ñ*

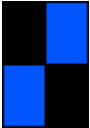
—

Code 165 

B	L
B	L

 CP437 = *Ñ*

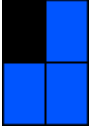
—

Code 166 

B	L
L	B

 CP437 = a


—

Code 167 

B	L
L	L

 CP437 = 0

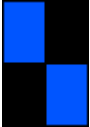
—

Code 168 

L	B
B	B

 CP437 = ¿

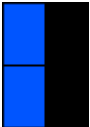
—

Code 169 

L	B
B	L

 CP437 = ㄱ

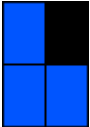
—

Code 170 

L	B
L	B

 CP437 = ㄴ

—

Code 171 

L	B
L	L

 CP437 = 1/2

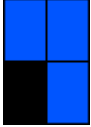
—

Code 172 

L	L
B	B

 CP437 = $\frac{1}{4}$

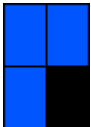
—

Code 173 

L	L
B	L

 CP437 = **i**

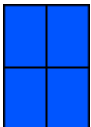
—

Code 174 

L	L
L	B

 CP437 = **<<**


—

Code 175 


L	L
L	L

 CP437 = **>>**


—

Code 176 


B	B
B	B

 CP437 = 


—

Code 177 


B	B
B	Y

 CP437 = 

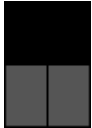
—

Code 178 

B	B
Y	B

 CP437 = 


—

Code 179 

B	B
Y	Y

 CP437 = |


—

Code 180 

B	Y
B	B

 CP437 = †

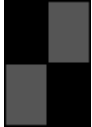
—

Code 181 

B	Y
B	Y

 CP437 = ‡

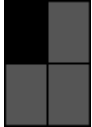
—

Code 182 

B	Y
Y	B

 CP437 = ††


—

Code 183 

B	Y
Y	Y

 CP437 = †††


—

Code 184 

Y	B
B	B

 CP437 = ‡


—

Code 185 

Y	B
B	Y

 CP437 = $\begin{matrix} \vdash \\ \vdash \end{matrix}$

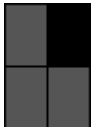
—

Code 186 

Y	B
Y	B

 CP437 = $\begin{matrix} \parallel \\ \parallel \end{matrix}$


—

Code 187 

Y	B
Y	Y

 CP437 = $\begin{matrix} \vdash \\ \vdash \end{matrix}$

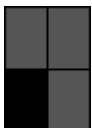
—

Code 188 

Y	Y
B	B

 CP437 = $\begin{matrix} \vdash \\ \vdash \end{matrix}$

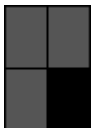
—

Code 189 

Y	Y
B	Y

 CP437 = $\begin{matrix} \vdash \\ \vdash \end{matrix}$

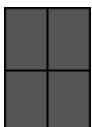
—

Code 190 

Y	Y
Y	B

 CP437 = $\begin{matrix} \vdash \\ \vdash \end{matrix}$


—

Code 191 

Y	Y
Y	Y

 CP437 = $\begin{matrix} \vdash \\ \vdash \end{matrix}$


—

Code 192 

B	B
B	B

 CP437 = L


—

Code 193 

B	B
B	W

 CP437 = ⊥

—

Code 194 

B	B
W	B

 CP437 = T


—

Code 195 

B	B
W	W

 CP437 = †

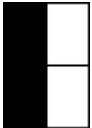
—

Code 196 

B	W
B	B

 CP437 = —

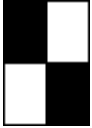
—

Code 197 


B	W
B	W

 CP437 = †

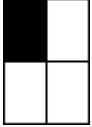
—

Code 198 


B	W
W	B

 CP437 = 


—

Code 199 


B	W
W	W

 CP437 = 

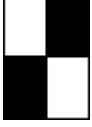
—

Code 200 


W	B
B	B

 CP437 = 

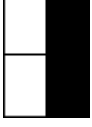
—

Code 201 


W	B
B	W

 CP437 = 

—

Code 202 

W	B
W	B

 CP437 = 

—

Code 203 

W	B
W	W

 CP437 = 

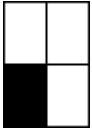

—

Code 204 

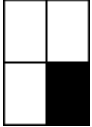

W	W
B	B

 CP437 = 

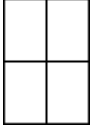
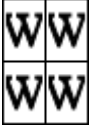
—

Code 205   CP437 = =


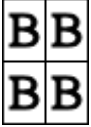
—

Code 206   CP437 = \perp

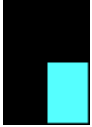
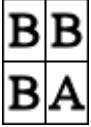
—

Code 207   CP437 = \perp

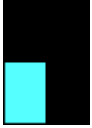
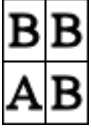
—

Code 208   CP437 = \perp

—

Code 209   CP437 = $\overline{\perp}$

—

Code 210   CP437 = $\overline{\perp}$

—

Code 211 

B	B
A	A

 CP437 = \mathbb{L}


—

Code 212 

B	A
B	B

 CP437 = \mathbb{L}

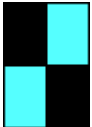
—

Code 213 

B	A
B	A

 CP437 = \mathbb{F}

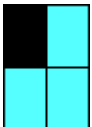
—

Code 214 

B	A
A	B

 CP437 = $\mathbb{\Pi}$

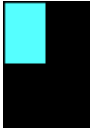
—

Code 215 

B	A
A	A

 CP437 = $\mathbb{\parallel}$

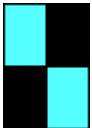
—

Code 216 

A	B
B	B

 CP437 = $\mathbb{\neq}$

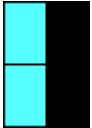
—

Code 217 


A	B
B	A

 CP437 = $\mathbb{\lrcorner}$

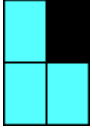
—

Code 218 


A	B
A	B

 CP437 = 

—

Code 219 


A	B
A	A

 CP437 = 

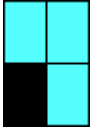
—

Code 220 


A	A
B	B

 CP437 = 

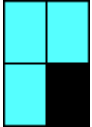
—

Code 221 


A	A
B	A

 CP437 = 

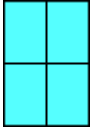
—

Code 222 


A	A
A	B

 CP437 = 


—

Code 223 

A	A
A	A

 CP437 = 

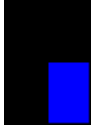
—

Code 224 

B	B
B	B

 CP437 = α

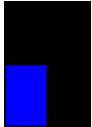
—

Code 225 

B	B
B	D

 CP437 = ß

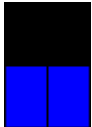
—

Code 226 

B	B
D	B

 CP437 = Γ

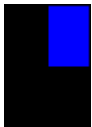
—

Code 227 

B	B
D	D

 CP437 = Π

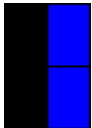
—

Code 228 

B	D
B	B

 CP437 = Σ

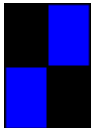
—

Code 229 

B	D
B	D

 CP437 = σ

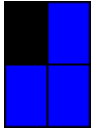
—

Code 230 

B	D
D	B

 CP437 = μ

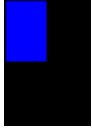
—

Code 231 

B	D
D	D

 CP437 = τ

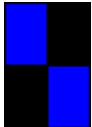
—

Code 232 

D	B
B	B

 CP437 = ϕ

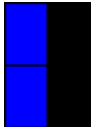
—

Code 233 

D	B
B	D

 CP437 = \ominus

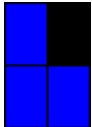
—

Code 234 

D	B
D	B

 CP437 = Ω

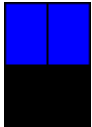
—

Code 235 

D	B
D	D

 CP437 = δ

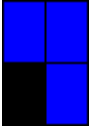
—

Code 236 

D	D
B	B

 CP437 = ∞

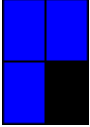
—

Code 237 

D	D
B	D

 CP437 = φ

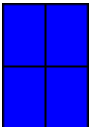
—

Code 238 

D	D
D	B

 CP437 = ϵ


—

Code 239 

D	D
D	D

 CP437 = \cap


—

Code 240 

B	B
B	B

 CP437 = \equiv


—

Code 241 

B	B
B	O

 CP437 = \pm


—

Code 242 

B	B
O	B

 CP437 = \geq


—

Code 243 

B	B
O	O

 CP437 = \leq


—

Code 244 

B	O
B	B

 CP437 = ∫


—

Code 245 

B	O
B	O

 CP437 = ∫


—

Code 246 

B	O
O	B

 CP437 = ÷


—

Code 247 

B	O
O	O

 CP437 = ≈


—

Code 248 

O	B
B	B

 CP437 = °


—

Code 249 

O	B
B	O

 CP437 = •

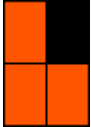
—

Code 250 

O	B
O	B

 CP437 = •


—

Code 251 

O	B
O	O

 CP437 = √


—

Code 252 

O	O
B	B

 CP437 = n


—

Code 253 

O	O
B	O

 CP437 = 2

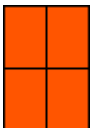
—

Code 254 

O	O
O	B

 CP437 = ■

—

Code 255 

O	O
O	O

 CP437 = space

=====

Appendix S

Decimal to Hexadecimal Conversions

<u>DEC</u>	<u>HEX</u>	<u>DEC</u>	<u>HEX</u>	<u>DEC</u>	<u>HEX</u>	<u>DEC</u>	<u>HEX</u>
000	00	032	20	064	40	096	60
001	01	033	21	065	41	097	61
002	02	034	22	066	42	098	62
003	03	035	23	067	43	099	63
004	04	036	24	068	44	100	64
005	05	037	25	069	45	101	65
006	06	038	26	070	46	102	66
007	07	039	27	071	47	103	67
008	08	040	28	072	48	104	68
009	09	041	29	073	49	105	69
010	0A	042	2A	074	4A	106	6A
011	0B	043	2B	075	4B	107	6B
012	0C	044	2C	076	4C	108	6C
013	0D	045	2D	077	4D	109	6D
014	0E	046	2E	078	4E	110	6E
015	0F	047	2F	079	4F	111	6F
016	10	048	30	080	50	112	70
017	11	049	31	081	51	113	71
018	12	050	32	082	52	114	72
019	13	051	33	083	53	115	73
020	14	052	34	084	54	116	74
021	15	053	35	085	55	117	75
022	16	054	36	086	56	118	76
023	17	055	37	087	57	119	77
024	18	056	38	088	58	120	78
025	19	057	39	089	59	121	79
026	1A	058	3A	090	5A	122	7A
027	1B	059	3B	091	5B	123	7B
028	1C	060	3C	092	5C	124	7C
029	1D	061	3D	093	5D	125	7D
030	1E	062	3E	094	5E	126	7E
031	1F	063	3F	095	5F	127	7F

<u>DEC</u>	<u>HEX</u>	<u>DEC</u>	<u>HEX</u>	<u>DEC</u>	<u>HEX</u>	<u>DEC</u>	<u>HEX</u>
128	80	160	A0	192	C0	224	E0
129	81	161	A1	193	C1	225	E1
130	82	162	A2	194	C2	226	E2
131	83	163	A3	195	C3	227	E3
132	84	164	A4	196	C4	228	E4
133	85	165	A5	197	C5	229	E5
134	86	166	A6	198	C6	230	E6
135	87	167	A7	199	C7	231	E7
136	88	168	A8	200	C8	232	E8
137	89	169	A9	201	C9	233	E9
138	8A	170	AA	202	CA	234	EA
139	8B	171	AB	203	CB	235	EB
140	8C	172	AC	204	CC	236	EC
141	8D	173	AD	205	CD	237	ED
142	8E	174	AE	206	CE	238	EE
143	8F	175	AF	207	CF	239	EF
144	90	176	B0	208	D0	240	F0
145	91	177	B1	209	D1	241	F1
146	92	178	B2	210	D2	242	F2
147	93	179	B3	211	D3	243	F3
148	94	180	B4	212	D4	244	F4
149	95	181	B5	213	D5	245	F5
150	96	182	B6	214	D6	246	F6
151	97	183	B7	215	D7	247	F7
152	98	184	B8	216	D8	248	F8
153	99	185	B9	217	D9	249	F9
154	9A	186	BA	218	DA	250	FA
155	9B	187	BB	219	DB	251	FB
156	9C	188	BC	220	DC	252	FC
157	9D	189	BD	221	DD	253	FD
158	9E	190	BE	222	DE	254	FE
159	9F	191	BF	223	DF	255	FF

=====

Appendix T: Programs

PRTCHRCK.BAS Program

```
100 '*****
110 '*
120 '* PRTCHRCK.BAS
130 '* MDJ 2021/08/05
140 '*
150 '* DISPLAYS THE CHARACTERS
160 '* CORRESPONDING TO CHR$
170 '* CODES 0-255
180 '*
190 '*****
200 '
210 CLS
220 FOR I = 0 TO 255
230 A$ = CHR$(I)
240 PRINT A$;
250 NEXT I
32767 END
```

Results:



POKECHEK.BAS Program

```
100 '*****
110 '*'
120 '* POKECHEK.BAS
130 '* MDJ 2021/08/05
140 '*'
150 '* DISPLAYS THE CHARACTERS
160 '* WHICH RESULT
170 '* RESPECTIVELY FROM EACH
180 '* POKE (0-255) INTO
190 '* VIDRAM (&H0400-&H05FF)
200 '*'
210 '*****
220 '
230 CLS
240 FOR I = 0 TO 255
250 L = &H0400 + I
260 POKE L,I
270 NEXT I
280 GOTO 280
32767 END
```

Results:



GPOKECHK.BAS Program

```
100 '*****
110 '*'
120 '* GPOKECHK.BAS
130 '* MDJ 2021/08/05
140 '*'
150 '* DISPLAYS THE SEMIGRAPHIC
160 '* CHARACTER WHICH RESULTS
170 '* RESPECTIVELY FROM EACH
180 '* POKE (128-255) INTO
190 '* VIDRAM (&H0400-&H05FF)
200 '*'
210 '* DISPLAYS ONE CHARACTER
220 '* AND ITS CODE AND THEN
230 '* PROMPTS FOR THE NEXT
240 '*'
250 '*****
260 '
270 FOR I = 128 TO 255
280 CLS
290 'DISPLAY THE CHARACTER
300 POKE &H0400, I
310 'SET THE CURSOR
320 'LOWER ADDRESS = HIGH BYTE
330 POKE &H0088, &H04
340 POKE &H0089, &H02
350 'DISPLAY THE CODE
360 PRINT I
370 'DISPLAY THE PROMPT
380 PRINT
390 PRINT "PRESS ANY KEY FOR NEXT CHAR."
400 A$ = INKEY$
410 IF A$ = "" GOTO 400
420 NEXT I
32767 END
```

Results:

The same as the Code Point 128-255 results from the POKECHEK.BAS program above, but on a character-by-character basis for greater clarity.

PRTTEST.BAS Program

NOTE: This is NOT a CoCo program. It runs under PC BASIC on Windows.

```
100 '*****
110 '*
120 '* PRTTEST.BAS
130 '* MDJ 2021/08/10
140 '*
150 '* OUTPUTS THE ORIGINAL
160 '* CODE PAGE 437 IBM PC
170 '* CHARACTER SET
180 '*
190 '* THIS PROGRAM IS WRITTEN IN
200 '* PC BASIC AND RUNS UNDER
210 '* WINDOWS 10
220 '*
230 '*****
235 '
240 OPEN "OUTPUT.TXT" FOR OUTPUT AS #1
250 FOR I = 0 TO 255
260 A$ = CHR$(I)
270 PRINT I, A$
280 PRINT #1, I, A$
290 NEXT I
300 '
310 PRINT
320 PRINT #1,
330 FOR I = 0 TO 255
340 A$ = CHR$(I)
350 PRINT A$;
360 PRINT #1, A$;
370 IF I = 31 THEN PRINT
380 IF I = 63 THEN PRINT #1,
390 IF I = 95 THEN PRINT
400 IF I = 127 THEN PRINT #1,
410 IF I = 159 THEN PRINT
420 IF I = 191 THEN PRINT #1,
430 IF I = 223 THEN PRINT
440 IF I = 255 THEN PRINT #1,
450 IF I = 255 THEN PRINT
460 IF I = 255 THEN PRINT #1,
470 IF I = 255 THEN PRINT
480 IF I = 255 THEN PRINT #1,
490 IF I = 255 THEN PRINT
```

```
500 IF I = 223 THEN PRINT #1,  
510 IF I = 255 THEN PRINT  
520 IF I = 255 THEN PRINT #1,  
530 NEXT I  
540 CLOSE #1  
32767 END
```

Results:

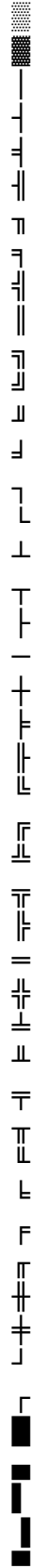
```
0  
1      —  
2      —  
3      —  
4      —  
5      —  
6      —  
7      —  
8      —  
9  
10  
  
11     —  
12     —  
13  
  
14     —  
15     —  
16     —  
17     —  
18     —  
19     —  
20     —  
21     —  
22     —  
23     —  
24     —  
25     —  
26     —  
27     —  
28     —  
29     —  
30     —  
31     —  
32  
33     !  
34     "  
35     #
```

36	\$
37	%
38	&
39	'
40	(
41)
42	*
43	+
44	,
45	-
46	.
47	/
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
58	:
59	;
60	<
61	=
62	>
63	?
64	@
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O
80	P
81	Q
82	R

83	S
84	T
85	U
86	V
87	W
88	X
89	Y
90	Z
91	[
92	\
93]
94	^
95	-
96	`
97	a
98	b
99	c
100	d
101	e
102	f
103	g
104	h
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w
120	x
121	y
122	z
123	{
124	
125	}
126	~
127	◆
128	Ç
129	ü

130	é
131	â
132	ä
133	à
134	å
135	ç
136	è
137	ë
138	è
139	ï
140	î
141	ì
142	Ä
143	Å
144	É
145	æ
146	Æ
147	ô
148	ö
149	ò
150	û
151	ù
152	ÿ
153	Ö
154	Ü
155	ç
156	£
157	¥
158	₠
159	f
160	á
161	í
162	ó
163	ú
164	ñ
165	Ñ
166	a
167	o
168	¿
169	Γ
170	┐
171	½
172	¼
173	i
174	«
175	»
176	⋮

177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223



PRTCHR02.BAS Program

Interestingly, this program is identical to PRTCHRCK.BAS except for the change of program name and date, and except for the addition of lines #241-247. It does exactly the same thing as PRTCHRCK.BAS except that, in addition to printing the character set to the screen, it also prints the character set to the BitBanger (Printer) port. The output there is quite similar (although not identical) to Code Page 437.

```
100 '*****
110 '*'
120 '* PRTCHR02.BAS
130 '* MDJ 2021/08/13
140 '*'
150 '* DISPLAYS THE CHARACTERS
160 '* CORRESPONDING TO CHR$
170 '* CODES 0-255
180 '*'
190 '*****
200 '
210 CLS
220 FOR I = 0 TO 255
230 A$ = CHR$(I)
240 PRINT A$;
241 PRINT#-2,A$;
242 'INTERESTINGLY, LINE 241
243 ' OUTPUTS WHAT APPEARS
244 ' TO BE VERY CLOSE TO THE
245 ' CODE PAGE 437 CHARACTER
246 ' SET TO THE BITBANGER
247 ' FILE.
250 NEXT I
32767 END
```


KEYCHECK.BAS Program

This program was used to check the characters and codes returned by each keypress on

1. My physical 64K CoCo 2 (cf. Appendix U).
2. My physical 128K CoCo 3 (cf. Appendix V).
3. My physical 512K CoCo 3 (cf. Appendix W).
4. The VCC CoCo (DECB) Keyboard (cf. Appendix H).
5. The VCC Natural (OS-9) Keyboard (cf. Appendix K).

```
100 '*****
110 '*
120 '* KEYCHECK.BAS
130 '* MDJ 2021/08/04
140 '*
150 '* KEYBOARD CHECK
160 '*
170 '* DISPLAYS CHARACTER
180 '* AND CHARACTER CODE
190 '* FOR EACH KEYPRESS
200 '*
210 '* NO EXIT MECHANISM
220 '*
230 '*****
240 '
241 ' FOR STANDARD UPPER CASE
242 ' POKE 282,1
243 ' FOR REVERSED LOWER CASE
244 ' POKE 282,0
245 POKE 282,1
250 A$ = INKEY$
255 IF A$ = "" GOTO 250
260 A = ASC(A$)
270 PRINT A$; A
280 GOTO 250
32767 END
```

SMPLCHEK.BAS Program

This program was used to double-verify the KEYCHECK.BAS results on

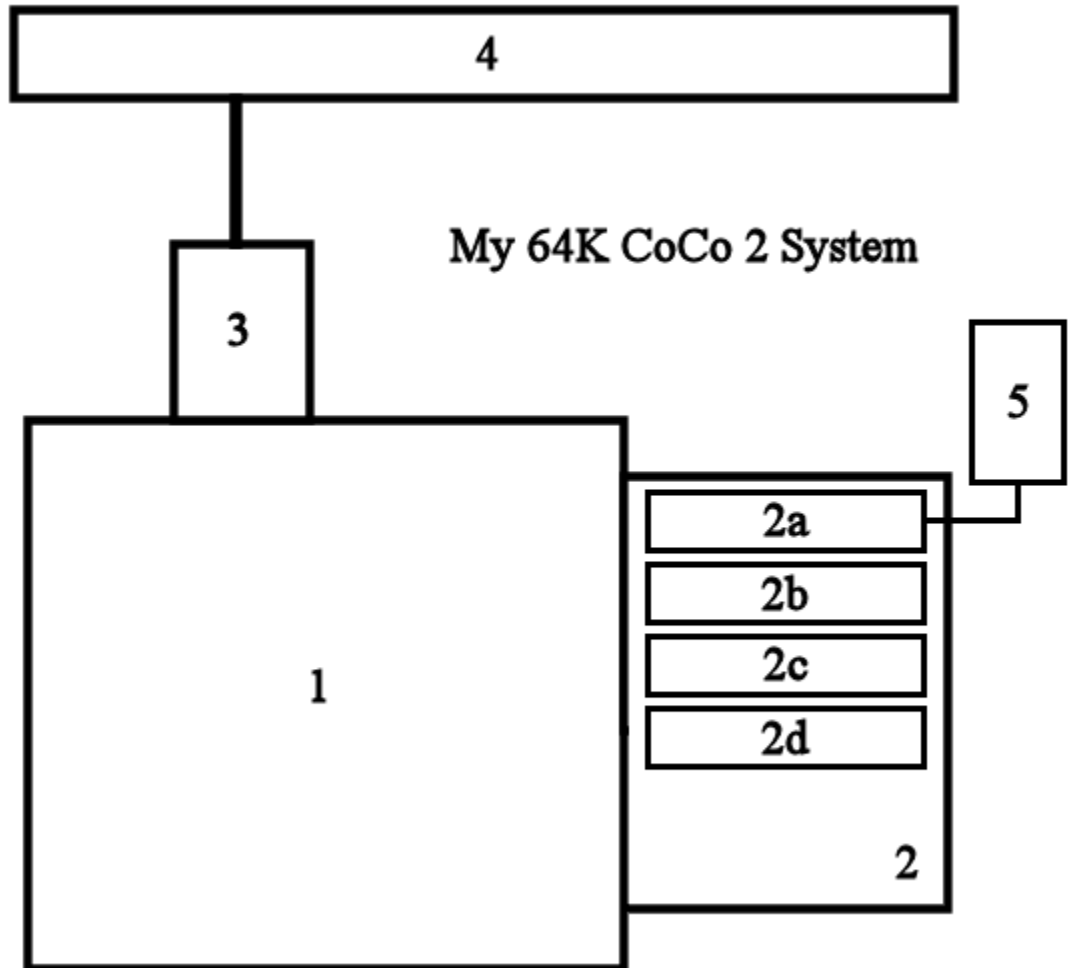
1. My physical 64K CoCo 2 (cf. Appendix U).
2. My physical 128K CoCo 3 (cf. Appendix V).
3. My physical 512K CoCo 3 (cf. Appendix W).
4. The VCC CoCo (DECB) Keyboard (cf. Appendix H).
5. The VCC Natural (OS-9) Keyboard (cf. Appendix K).

```
1000 '*****
1010 '*
1020 '* SMPLCHEK.BAS
1030 '* MDJ 2021/09/07
1040 '*
1050 '* PRINTS ALL CHARACTER
1060 '* CODES AS CHR$(I)
1070 '* BRACKETED BY "****"
1080 '* ON EACH SIDE TO
1090 '* DETECT NON-PRINTING
1100 '* CHARACTERS
1110 '*
1120 '*****
1130 '
1140 I=0
1150 PRINT I;" ****";CHR$(I);"****"
1160 I = I + 1
1170 '
1180 'WAIT FOR KEYPRESS
1190 A$ = INKEY$
1200 IF A$ = "" GOTO 1190
1210 IF I < 256 GOTO 1150
1220 '
1230 'EXIT
32767 END
```

=====

Appendix U: My 64K CoCo 2 System

As of 2021/09/08, my physical 64K CoCo 2 System is depicted in the block diagram below.



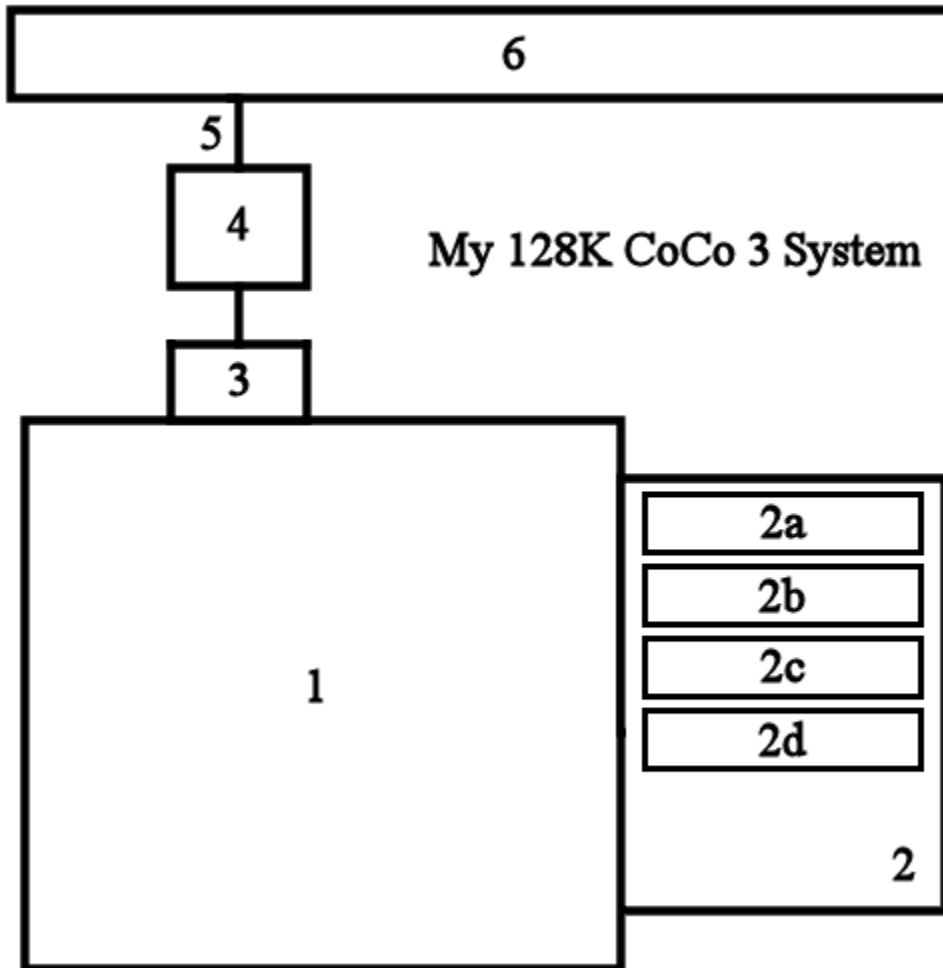
System Legend:

1. 64K CoCo 2, Model Number 26-3127, Serial Number 007601.
2. Multi-Pak Interface, Model Number 26-3124, Serial Number 2005259
 - 2a. Floppy Disk Controller, Model Number 26-3029
 - 2b. CoCo SDC, running SDC-DOS 1.6 CC2.
 - 2c. RS-232 Pak
 - 2d. Empty slot
3. CoCo VGA.
4. RCA 19" VGA/HDMI Monitor, Model RT1970.
5. 5.25" Disk Drive, Model Number 26-3022

=====

Appendix V: My 128K CoCo 3 System

As of 2021/09/08, my physical 128K CoCo 3 System is depicted in the block diagram below.



See following page for System Legend.

System Legend:

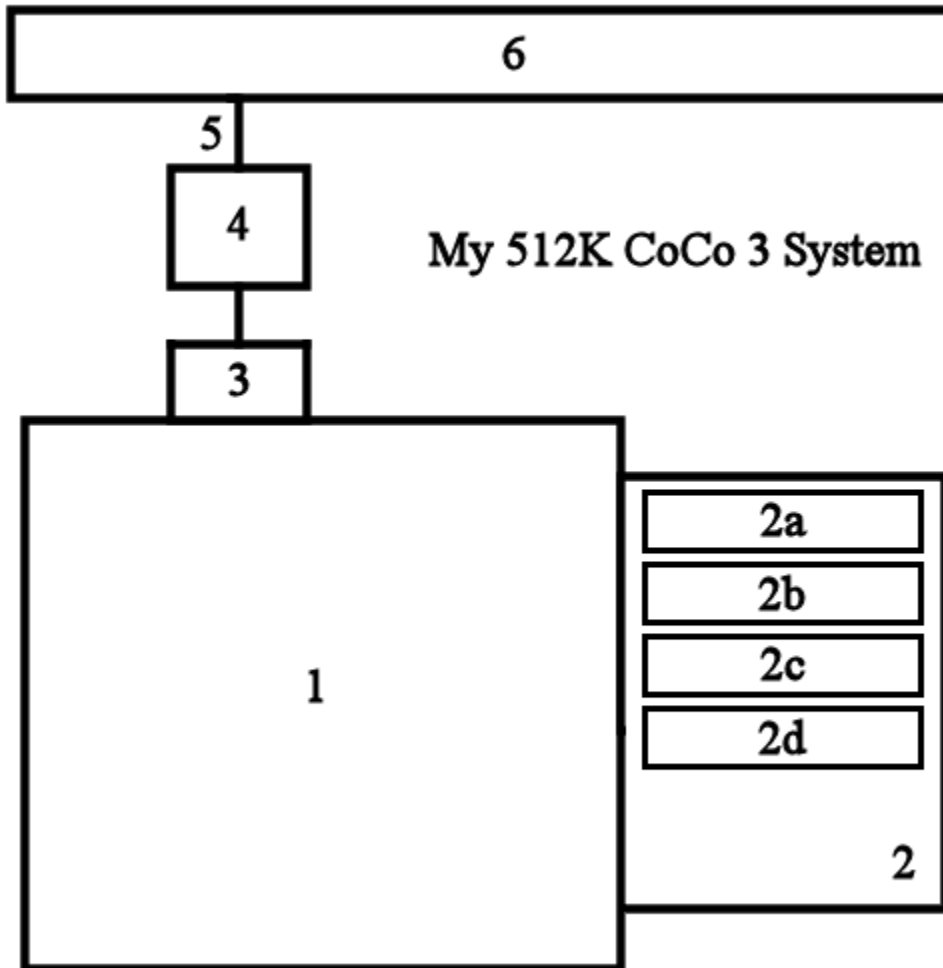
1. 128K CoCo 3, Model Number 26-3334, Serial Number 1110785
2. Mega-Mini MPI
 - 2a. CoCo SDC, running SDC-DOS 1.6 CC3
 - 2b. RS-232 Pak
 - 2c. Empty slot
 - 2d. Empty slot
3. CoCo Switch-a-Roo*
4. TNP SCART to HDMI Converter
5. HDMI Cable
6. RCA 19" VGA/HDMI Monitor, Model RT1970

*Not Shown: The Switch-a-Roo is powered from a ByEasy powered 7-port USB 2.0 hub.

=====

Appendix W: My 512K CoCo 3 System

As of 2021/09/08, my physical 512K CoCo 3 System is depicted in the block diagram below.



See following page for System Legend.

System Legend:

1. 512K CoCo 3, Model Number 26-3334, Serial Number 1037654
2. Mega-Mini MPI
 - 2a. CoCo SDC, running SDC-DOS 1.6 CC3
 - 2b. RS-232 Pak
 - 2c. Empty slot
 - 2d. Empty slot
3. CoCo Switch-a-Roo*
4. TNP SCART to HDMI Converter
5. HDMI Cable
6. RCA 19" VGA/HDMI Monitor, Model RT1970

*Not Shown: The Switch-a-Roo is powered from a ByEasy powered 7-port USB 2.0 hub.

=====

Appendix X: My CoCo Philosophy

The CoCo community enjoys a great diversity of interests.

Some choose to concentrate on hardware innovations and modifications such as interfacing with VGA and HDMI monitors, SD Card data storage, and 104-key keyboards. This interest is at least partly born of necessity, since composite monitors, floppy diskettes, and CoCo spare parts are no longer manufactured and are in increasingly short supply.

Others concentrate on expanding the software horizons of the CoCo 3, using NitrOS-9 and other operating systems to make the multitasking CoCo behave ever closer to modern Windows, Mac, and Linux machines.

Still others are devoted to emulating the CoCo on other platforms by developing emulators such as VCC, OVCC, MAME, and XRoar.

And some just love retro gaming.

My personal interest is twofold:

1. To see VCC increasingly used as a learning tool for budding software developers.
2. To see just how much I can cram into a 64K CoCo 2.

First, VCC: Today's Grade School, Junior High, and High School students have a wealth of available learning tools. Micro-bits, Arduinos, and Raspberry Pi supermicro devices provide highly affordable entry-level introductions to computer programming and interfacing. Maker-Spaces and Innovation Centers in our schools and libraries help foster growth and experience.

But these devices do have limitations. Even these simple(?) computers can have rather steep learning curves, and their low initial cost can quickly expand as new peripherals and experimental equipment and supplies are added.

VCC is free, and can be used on any Windows computer: just download it, install it, and it runs. If you don't own a Windows computer, your school, library, or a friend probably does. The included BASIC language is easy to learn and can readily serve as a stepping-stone towards more complex programming languages. (And, no, learning structured programming does not require a language that enforces structure. In fact, I think learning to structure your programs is actually more effective when you do so on your own.)

I prefer VCC to the other emulators for these purposes because its setup is trivial: Again, just download it, install it, and it runs. OVCC, MAME, and XRoar have their advantages, but ease of setup is not one of them. Even with their available Windows binary packages, they require pre-installation of other bits and pieces of software before they can be downloaded, installed, and run. This may not be a

major problem for a reasonably adept aficionado, but it forms a significant barrier for the newbie. And, it's the newbie whom we're trying to reach, interest, and encourage here; the newbie who may not yet recognize even the tiniest awakening of interest in things computational.

But, for these purposes, VCC has one glaring weakness: its instruction manual is woefully terse. I would like to see VCC bundled with a selection of tutorials, manuals, and examples suited to guiding even the most newbie of newbies into the wonders of computing.

Second, The Stuffed CoCo: I'm simply fascinated by the challenge of seeing how much functional capability I can sandwich into the nooks and crannies of the 64K space. Whether it's working in the available RAM left by the 32K ROM and the dedicated RAM that supports that ROM, or whether it's jumping right into ALLRAM mode and just filling the entire 64K to near-overflowing; it's an investigative gauntlet which goes right to the heart of my enchantment with puzzles in general.

It's great fun!

M.D.J. 2021/08/29

=====

Appendix Y: New BDS Software License

This New Software License applies to all software found on the BDS Software site, and supersedes all previous copyright notices and licensing provisions which may appear in the software itself or in any documentation therefor.

All software which has previously been placed in the public domain remains in the public domain.

All other software, programs, experiments and reports, documentation, and any other material on this site (other than that attributed to outside sources) is hereby copyright © 2018 (or later if so marked) by M. David Johnson.

All software, documentation, and other information on the BDS Software site is available for you to freely download without cost.

Whether you downloaded such items directly from this site, or you obtained them by any other means, you are hereby licensed to copy them, to sell or give away such copies, to use them, and to excerpt from them, in any way whatsoever, so long as nothing you do with them would denigrate the name of our Lord and Savior, Jesus Christ.

I make absolutely no warranty whatsoever for any of these items. You use them entirely at your own risk.

If they don't work for you, I commiserate.

If they crash your system, I sympathize.

But I accept no responsibility whatsoever for any such consequences. Under no circumstances will BDS Software or M. David Johnson be liable for any negative results of any kind which you may experience from downloading or using these items.

BDS Software's former mail address at P.O. Box 485 in Glenview, IL is no longer valid. Any mail sent to that address will be rejected by the U.S. Postal Service. See my Contact page.

M.D.J. 2018/06/08

=====

Works Cited

6809.org.uk . “XRoar Online.” *TRS-80 Color Computer Archive*.

<https://colorcomputerarchive.com/xroar-online/> , Web. Last Accessed: 2021/08/11.

“ASCII.” *Wikipedia*. <https://en.wikipedia.org/wiki/ASCII> , Last Edited: 2021/06/08. Web. Last Accessed: 2021/08/10.

“Code page 437.” *Wikipedia*. https://en.wikipedia.org/wiki/Code_page_437 , Last Edited: 2021/07/11. Web. Last Accessed: 2021/08/10.

Getting Started With Extended Color BASIC. Fort Worth, TX: Radio Shack, 1983. Print.

Hagemans, Rob. “PC-BASIC 1.2.14 for Windows.” *PC BASIC*.

<https://robhagemans.github.io/pcbasic/download.html> . Web. Last Accessed: 2021/08/11.

Pierce, Bill. “VCC” *TRS-80 Color Computer Archive*.

<https://colorcomputerarchive.com/repo/Emulators/Coco%203/Vcc/> , Web. Last Accessed: 2021/08/11.

=====