

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

Towards a
VCC Bundle

by M. David Johnson

2021/09/29

Abstract

A Plan is presented for developing a VCC Bundle, i.e. a collection of tutorials, manuals, and examples, built around the VCC Emulator, and suited to guiding even the most newbie of newbies into the wonders of computer programming and coding.

Table of Contents

Abstract	2
Introduction	4
Design Plan	5

Appendix A: My CoCo Philosophy	9

References	11

Introduction

Children and teenagers (and even adults), who show no interest in computers beyond using them to prepare homework and to play computer games, may yet be awakened to the challenges and pleasures of computer programming (i.e. “coding”) if they’re introduced to those challenges and pleasures in very simple, fun, and small, doses.

In this paper, in accordance with my CoCo philosophy (See Appendix A), I propose to build a bundle of such small doses around the VCC Color Computer 3 emulator, as originally developed by Joseph Forgeone, and as updated and published by Bill Pierce in version 2.1.0d on 2021/07/03, under the 2007/06/29 Version 3 of the GNU General Public License. I’ll, thereafter, endeavor to update the bundle as needed to incorporate changes in the VCC version.

Although I’ll be using the VCC Color Computer 3 emulator, I’ll initially be using it as if it were a 64K CoCo 2. This will further simplify the small doses. When we were babies, we began our foray into the world of walking by ... ahem ... taking baby steps. This is the way the VCC Bundle will begin: with baby steps.

This paper describes my plan for the first set of doses, to be entitled “Twelve Doses of VCC”. Over the coming months, I hope to be able to expand this plan into a completed first element of a bundle, to be available both interactively on my website, and also as a .pdf file with associated .DSK files.

Design Plan

This is my initial plan for “Twelve Doses of VCC”. Any suggestions for additions or revisions to this initial plan will be gratefully received. Also, if you would be interested in authoring a specific section or sections of these Doses, please contact me at info@bds-soft.com .

The opening First Dose will introduce the VCC Bundle itself, and answer questions like:

1. Why code?
2. Why and how is coding fun?
3. Why use a computer that is a single unit instead of building one with an Arduino or a Raspberry Pi?
4. Why use a 30 year old computer instead of a more modern one?
5. Why use a CoCo instead of the more ubiquitous IBM-PC?
6. Why use an emulator instead of a “real” computer?
7. Why use VCC instead of some other emulator?
8. What does the VCC Bundle require?
9. What does the VCC Bundle include?
10. How to begin?
11. Resources

The Second Dose:

1. Downloading VCC.
2. Downloading the Bundle
3. Installing VCC.
4. Setting VCC Options
5. “Hello World!”
6. Installing the Bundle
7. The Floppy Disk
8. Loading Programs from Disk
9. Running BASIC Programs.
10. Text, Graphics, Sound, and Hardcopy
11. Some Simple Programs

The Third Dose:

1. Why BASIC?
2. Writing BASIC Programs
3. Saving BASIC Programs
4. Data Input to BASIC Programs
5. Screen Output from BASIC Programs

6. Keyboard Input to BASIC Programs
7. Sample Programs
8. I/O for Sample Programs
9. Exercises
10. I/O for Exercises

The Fourth Dose:

1. Variables and Constants
2. Decision Making
3. Structured Programming
4. Determining True and False
5. Determining Equality and Inequality
6. Relative Values
7. Error Trapping and Handling
8. Sample Programs
9. I/O for Sample Programs
10. Exercises
11. I/O for Exercises

The Fifth Dose:

1. Arrays
2. Looping
3. Structured Programming Again
4. FOR...NEXT Loops
5. Simulating a DO Loop
6. Simulating a WHILE Loop
7. Sample Programs
8. I/O for Sample Programs
9. Exercises
10. I/O for Exercises

The Sixth Dose:

1. Functions and Subroutines
2. A Little Touch of Math
3. Random Numbers
4. Number and String Conversions
5. Joysticks and Printers
6. Some (Gasp!) Advanced Math
7. Sample Programs
8. I/O for Sample Programs
9. Exercises
10. I/O for Exercises

The Seventh Dose:

1. Text Processing
2. Files
3. Cassette Files
4. Disk Files in General
5. Sequential Access Disk Files
6. Random Access Disk Files
7. Sample Programs
8. I/O for Sample Programs
9. Exercises
10. I/O for Exercises

The Eighth Dose:

1. Logic
2. Screen Organization
3. Little Programming Puzzles and Gems
4. Algorithms
5. System Organization
6. Some Thoughts on Game Design
7. Sample Programs
8. I/O for Sample Programs
9. Exercises
10. I/O for Exercises

The Ninth Dose:

1. Graphics
2. The Ubiquitous Semigraphics Mode
3. PMODE Graphics
4. Screens and Screen Design
5. Graphic Elements (Lines, Circles, etc.)
6. Graphic Files
7. Sample Programs
8. I/O for Sample Programs
9. Exercises
10. I/O for Exercises

The Tenth Dose:

1. Sound
2. Sound Effects
3. Music
4. Integrating Sound and Graphics
5. User Sound Control

6. Sound Files
7. Sample Programs
8. I/O for Sample Programs
9. Exercises
10. I/O for Exercises

The Eleventh Dose:

1. Machine Language
2. Peeking and Poking
3. Exploring Memory
4. Altering Memory
5. Reserving and Allocating Memory
6. Interfacing with BASIC Programs
7. Documentation
8. Sample Programs
9. I/O for Sample Programs
10. Exercises
11. I/O for Exercises

The Twelfth Dose:

1. Assembly Language
2. Structured Programming One More Time
3. Using Assembly for Speed
4. Using Assembly for Stuff BASIC Can't Do
5. Replacing BASIC with Assembly
6. Coordinating Assembly With BASIC
7. Sample Programs
8. I/O for Sample Programs
9. Exercises
10. I/O for Exercises

A Bonus Dose:

1. Advancing to the CoCo 3
2. Other Platforms
3. Other Programming Languages
4. Operating Systems
5. Some Final Thoughts

Resources and References:

Yet to be compiled

Appendix A: 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

References

Color Computer System Programming Manual. Fort Worth, TX: Radio Shack, 1987. Print.

Getting Started With Color BASIC. Fort Worth, TX: Radio Shack, 1980. Print.

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

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

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