Commodore 64: A Look at a BASIC language

By Javier Rodriguez and Amar Jagrup

Current Software and design Implementation: Since basic is a different language than what we're used to and we don't have the actual hardware to go one. Software is going to be one of the last things we work on if we have the time. Since we plan on making small programs for most of the sections we feel that should be sufficient coding from us. If we end up finishing the paper early, then we plan on synthesizing code from the graphics and sound section and make a program from that timeline.

Timeline:

Week of 11/11/19: Try to write at least one page of a topic this week if not finish a topic.

Week 2: Each member will finish another topic of the paper this week and start the the last 2 topic for week 3.

Week 3: Paper should be finished this week assuming none of the sections in this paper took longer than expected. Work should start on the combined program at this point if we have the time.

Week 4: This week will be spent on finishing up any loose ends either the program or the paper if the paper ended up being longer than expected.

Overview and Hardware

This section will discuss the hardware and have a basic overview of the Commodore 64 and the purpose of this paper. Will most likely not be very long. The section is hear more so so that readers can understand the limitations that the language has to deal with.

The Commodore 64(C64) at the time it was developed was designed around the philosophy that the sound and graphics of the system were to be state of the art for the world's next big video game(Matthews). The purpose of this paper is to explore the now almost forgotten language used in the C64 on hardware it was built on. First let's look at the hardware in detail. The C64 uses MOS Technologies 6510 CPU with a clock speed of 1Mhz, a VIC-II 6567 video chip which produces 16 colors and the ability to display a resolution of 320x200, a SID 6581 sound chip which at the time was truly state of the art, it could produce a recognizable human voice without additional hardware. While the Commodore 64 does have 64K of ram available only 38,911 bytes are actually free because the rest of the memory goes towards the internal function of the language Commodore Basic 2.0(Matthews). Being developed with video games in mind the C64 has several parts of it's hardware devoted to it. It has two game parts for controllers on the side panel. On the rear of the

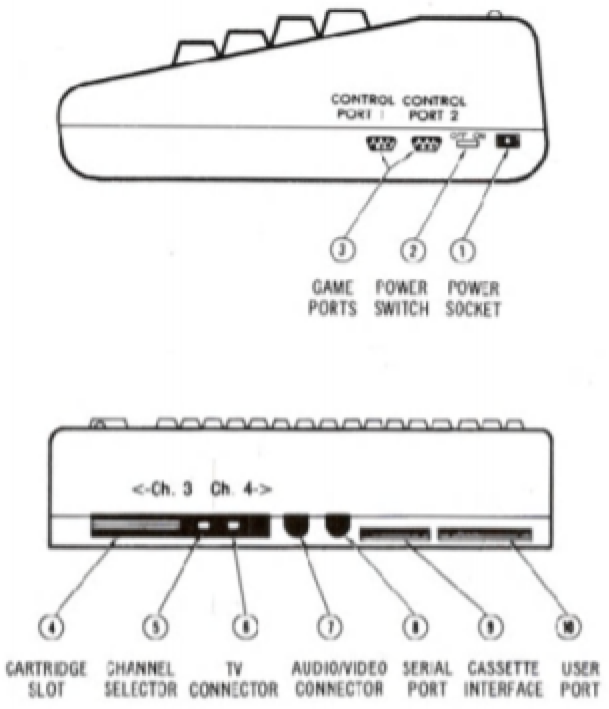


Figure 1:From the C64 User's Guide

system it has a cartridge slot specifically for game cartridges, There is also a cassette interface which connects to an eternal cassette reader which is used to load programs from cassettes(Commodore Business Machines) This can be seen in Figure 1(For more information on other aspects of the hardware not covered here see the C64 reference manual).

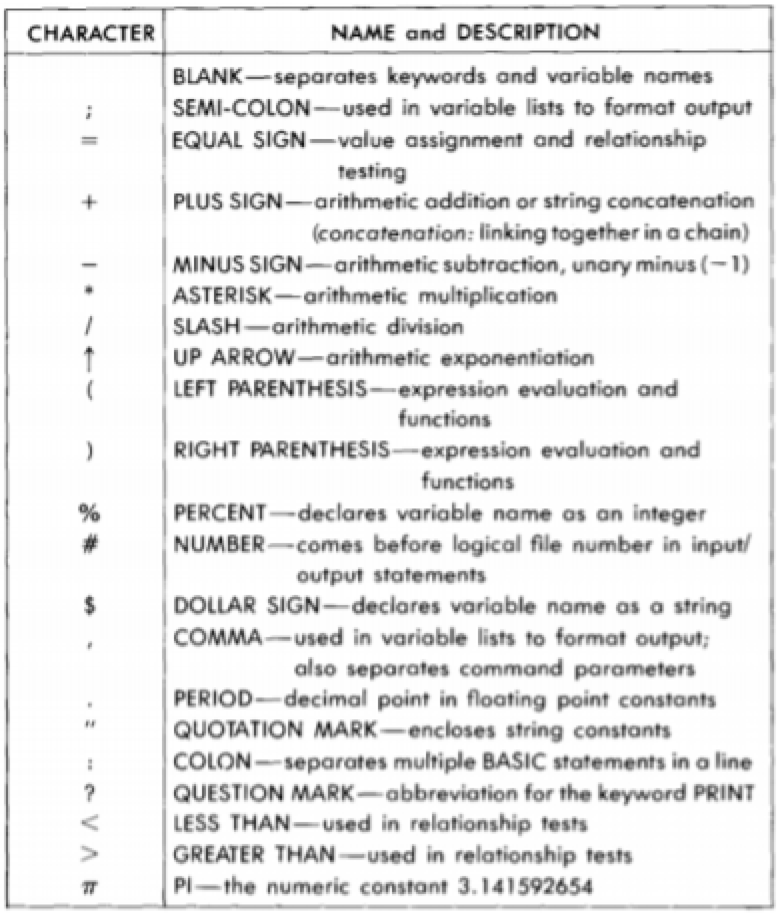
At the time these specifications were extraordinary. Now they pale in comparison to the weakest of smartphones. However, wasn't just it's power that made the system popular it was the also the language it used. In order to create software and in order to use software on the C64. The system used a language called Basic. The language gave the user essentially full access to the machine's now laughable hardware capabilities and allowed for games and programs that at the time were hard to replicate elsewhere. Basic, while versatile is now almost an alien language to the modern programmer.

We will be examining Basic on the C64 and describe its functions in four topics Graphics, Sound, Input/Output and comparing Basic to the functionality provided by modern programming languages. The following section **Basic Commands** will cover the keywords used in the programs that will be shown in the sections described and will contain keywords that most Basic programmers should know.

Basic Commands

This Section will attempt to cover all the commands in the language and provide a description along with an image for each of them. This section is here so if a reader is confused on a command we use they can look it up easily here.

Before we discuss the the keywords used by the language it's imperative to understand what the Basic Interpreter is and what it does. The Basic interpreter is what analyzes the statement syntax the programmer types in and performs the required calculations and data manipulations. There are a total of 65 keywords that the interpreter recognized. There are also several characters that contain special meaning to the interpreter(Commodore Business Machines) . This can be seen in Figure 2.



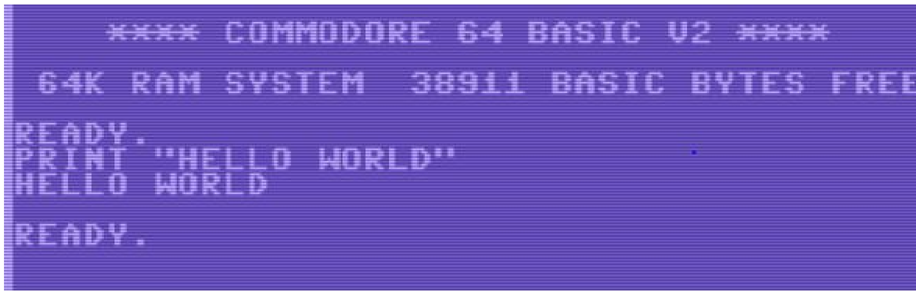
CHARACTER	NAME and DESCRIPTION
	BLANK—separates keywords and variable names
;	SEMI-COLON—used in variable lists to format output
=	EQUAL SIGN—value assignment and relationship testing
+	PLUS SIGN—arithmetic addition or string concatenation (concatenation: linking together in a chain)
−	MINUS SIGN—arithmetic subtraction, unary minus (−1)
*	ASTERISK—arithmetic multiplication
/	SLASH—arithmetic division
↑	UP ARROW—arithmetic exponentiation
(	LEFT PARENTHESIS—expression evaluation and functions
)	RIGHT PARENTHESIS—expression evaluation and functions
%	PERCENT—declares variable name as an integer
#	NUMBER—comes before logical file number in input/output statements
$	DOLLAR SIGN—declares variable name as a string
,	COMMA—used in variable lists to format output; also separates command parameters
.	PERIOD—decimal point in floating point constants
"	QUOTATION MARK—encloses string constants
:	COLON—separates multiple BASIC statements in a line
?	QUESTION MARK—abbreviation for the keyword PRINT
<	LESS THAN—used in relationship tests
>	GREATER THAN—used in relationship tests
π	PI—the numeric constant 3.141592654

Figure 2: from the C64 Programmers Reference Manual

There are two modes of operation in Basic direct mode and programming mode. Direct mode doesn't use line numbers and the statement is executed whenever the "Return" Key is pressed. Programming mode is the one used for actually used for making programs. Each line has a number associated with it. Multiple statements can be used one one line but the system has an 80 character limit for each line. So, if that's reached than the next statement has to be placed on the next line(Commodore Business Machines) With all of that we can now discuss the key words that Basic uses. They are as follows.

PRINT:

Used to output strings, numbers and the value of variables to the screen. Can be used in Direct mode but more meant for programming mode where it can be used to print the result of a program.



```
    **** COMMODORE 64 BASIC V2 ****
 64K RAM SYSTEM  38911 BASIC BYTES FREE
READY.
PRINT "HELLO WORLD"
HELLO WORLD

READY.
```

INPUT:

.

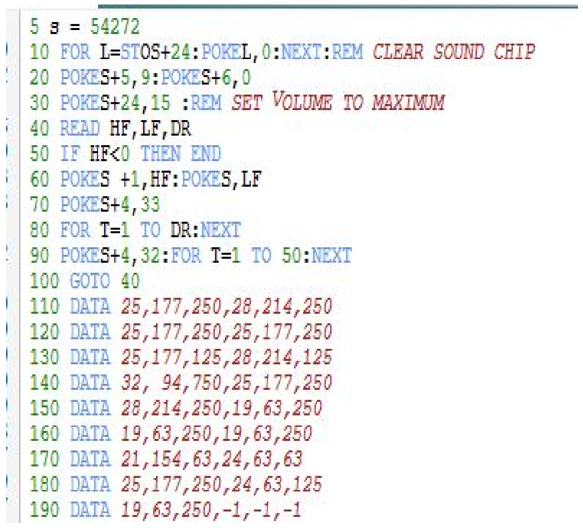
Graphics

This section will be focused on discussing the graphics capabilities that Basic provides. Sections like these will most likely include a few demonstrations in the form of images of emulator and code snippets. Since we’re not programming on the emulator itself we should be able to include the code with little to no issue.

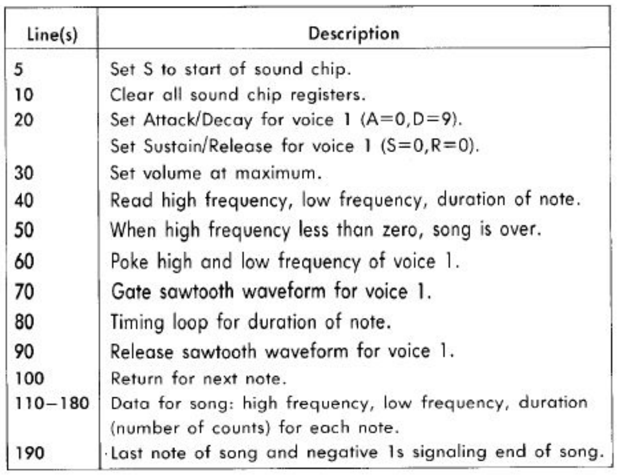
Sound

Very similar structure to the graphics section. The only real difference is that since we can't show sound through images, we might include videos of showing what sound our code produces.

The commodore 64 has the ability to play sounds and using basic you can make complex sounds and songs. Able to play music because the commodore has a SID chip. It comes complete with three voices. To program sounds on the C64 you will use the POKE statement. POKE sets the indicted memory location(MEM) equal to a specific value(NUM)(C64 reference guide). There are specific memory locations for making music on the commodore and they are 54272 to 54296. The numbers you use in the POKE statement must be between 0 and 255. There is also a peek function, which is equal to the current value in the indicated memory location.



```
5 S = 54272
10 FOR L=STOS+24:POKEL,0:NEXT:REM CLEAR SOUND CHIP
20 POKES+5,9:POKES+6,0
30 POKES+24,15 :REM SET VOLUME TO MAXIMUM
40 READ HF,LF,DR
50 IF HF<0 THEN END
60 POKES +1,HF:POKES,LF
70 POKES+4,33
80 FOR T=1 TO DR:NEXT
90 POKES+4,32:FOR T=1 TO 50:NEXT
100 GOTO 40
110 DATA 25,177,250,28,214,250
120 DATA 25,177,250,25,177,250
130 DATA 25,177,125,28,214,125
140 DATA 32, 94,750,25,177,250
150 DATA 28,214,250,19,63,250
160 DATA 19,63,250,19,63,250
170 DATA 21,154,63,24,63,63
180 DATA 25,177,250,24,63,125
190 DATA 19,63,250,-1,-1,-1
```



Line(s)	Description
5	Set S to start of sound chip.
10	Clear all sound chip registers.
20	Set Attack/Decay for voice 1 (A=0,D=9). Set Sustain/Release for voice 1 (S=0,R=0).
30	Set volume at maximum.
40	Read high frequency, low frequency, duration of note.
50	When high frequency less than zero, song is over.
60	Poke high and low frequency of voice 1.
70	Gate sawtooth waveform for voice 1.
80	Timing loop for duration of note.
90	Release sawtooth waveform for voice 1.
100	Return for next note.
110–180	Data for song: high frequency, low frequency, duration (number of counts) for each note.
190	Last note of song and negative 1s signaling end of song.

(Figure 4:Example code from c64 reference manual that plays some music shown on the left and a line by line explanation of the code shown on the right)

Input and Output

This section will discuss input and output peripherals for the C64 and other topics related to it. Will most likely focus on what the programmer has to do get these peripherals to work. The complexity of the section will depend on how well the emulator is able to emulate these functions of the C64.

Modern Languages

This section will look at functions and structures that modern use and say if they can be replicated in Basic. If not we will explain why not to the best of our ability.

Works Cited

Commodore Machines Inc. Commodore 64 Programmer's Reference Guide. First ed., Commodore Machines Inc., 1982, http://www.classiccmp.org/cini/pdf/Commodore/C64 Programmer's Reference Guide.pdf.

This is the programmer's reference guide to the Commodore 64 and it looks like it contains most of the information we need for this project. It covers all of our topics to some extent so this source will the main one we will use.

Zimmers. "The Commodore 64." Zimmer.net, http://www.zimmers.net/cbmpics/c64s.html.

This source just contains the links to the C64 User's guide and reference manual. As well as other official material.

The Commodore 64 Users Guide. Commodore Business Machines (Uk), 1984,

This source is for the C64 users guide which we obtained from the Zimmer site. It includes some information not found on the programmers reference guide. So it should be useful for information on the C64 in general.

Mathhews, Ian. "Commodore 64 – The Best Selling Computer In History." Commodore Computers, Running Technologies Inc, 29 Nov. 2018, https://www.commodore.ca/commodore-products/commodore-64-the-best-selling-computer-in-history/.

This source contains a summary of several other sources on several aspects of the Commodore 64. We'll be using this for mainly to describe the hardware in a bit more detail than what the reference manual and user's guide describes it as.