THE EDITOR'S GODZILLA!

COMMODORE MAYHEM

-by Lenard R. Roach

Not very long ago I demonstrated "Bible Games for the Commodore 64" to the children's pastoral staff at the Bonner Springs Church of the Nazarene in Bonner Springs, Kansas. After the demonstration, the leader of the children's ministry stepped up to me with a most interesting proposition.

"Lenard," Barbara said, “can you come up with a program that will help me teach my Quizzing Class about the history, geography, and verses of certain sections of scripture? This year we are going through First and Second Samuel."

I instantly had some reservations. "I don't have any notes or papers on 1st or 2nd Samuel, let alone a list of questions to ask the class."

Barb had a quick answer. "I have the notes that the Quizzing Committee sent to all the churches in the district. I can make you a copy of them. There is one question with 3 possible answers each. I think that there are 900 total questions with 1,400 answers in 20 modules. Do you think that you can code that much?"

Again I persisted, "I haven't written code on the Commodore since 2016. It will take me some time to get into the groove. What would you like the program to contain, other than the questions and answers?"

Barb didn't hesitate for a minute. "I was thinking of some graphics, some color, and maybe sound effects. One effect for a right answer and another for a wrong answer."

Graphics? Color? Sound effects? Who did Pastor Barbara think I was? I was not a game designer. Blessed by Heaven, I had created software for budgeting and debt paying that solved the personal traumas that had struck my household. Yet, this new project would be well beyond anything that I had ever done.

Over the course of years, my son Gabriel had shown me that before you embark into unknown territory, you research the terrain. This meant that I’d be pulling out my Commodore 64 User's Guide and my Commodore 64 Programmer's Reference Guide and going through them in search of the proper procedures to get started on the project, let alone finish it.

I acquiesced. "When do you need the project done?"

"We'll be doing a new section of 20 questions each week. So, if you can do a section at a time, it would be appreciated."

Oh (bleep)! I was having enough trouble with other things in my life at the moment. How was I going to work this in? It was September when Pastor Barb asked me to start the project. I was between owning cars, i.e., I was using rentals to get around. I’d have to use a rental, try and go to work, deal with smaller fires trying to burn down my personal forest, AND get 20 questions with 60 answers done once a week. To take on a new Commodore project in the middle of this, I was either stupid, insane, or both; however, my word was my bond. I would work on the project in whatever free time that I had, and I would try to come up with a section a week as time went forward. But first I needed a plan.

After my talk with Pastor Barb, I went home to both my waiting cats and my Commodore 64. The cats nuzzled and purred around my legs, and to avoid stepping on my little fur babies, I had to baby-step towards the computer room. I quickly patted each of their heads, and they broke away. Making it to the computer chair, I sat down and turned on the classic computer and the corresponding 1541 disk drive, the C64 home screen appearing on the Commodore 1902A monitor.

The start plan seemed simple enough at first. Take the first question of Section One with its three multiple choice questions and transpose what was on paper to my Commodore 64. Simple enough, I thought. Lots of PRINT statements with a bunch of IF/THEN subroutines to give the player the choice of either a right or wrong answer. Now, the first question and its answers were:

Name four of the twelve apostles.

1. Peter, James, Andrew, and John

2. Pilate, Caiaphas, Matthias, and Judah

3. Daniel, Shadrach, Meshach, and Abednego

What? How was a ten year old Bible student supposed to figure that out, especially if the boy or girl was new to the scriptures, which most of them were?

This phraseology had to change but to what? Hmm. I mulled around a few ideas until the more comedic side of me decided to have a say. Soon, the corrections were made.

Name four of the twelve apostles.

1. Peter, James, Andrew, and John

2. Batman, Robin, Wonder Woman, and Superman

3. SpongeBob, Patrick, Sandy, and Squidward

Now that would give a player a fighting chance!

The next thing I needed to code were the graphics. Oh, boy! The only graphics I had ever worked with were the PETSCII graphics provided on the Commodore keyboard. The last time I had used PETSCII graphics was on the opening screens for my budgeting software that I presented at the last Commodore Vegas Expo, and it did not receive good reviews from those viewing it at the show. Oh, well. In this day and age (2022) any publicity, good or bad, was considered positive publicity, so I opted to use the PETSCII graphics for this project as well.

To notify the user of a wrong answer, I decided to display a big, bright red, PETSCII-built X on a white background. By using the corresponding Commodore keys and the Commodore key or the SHIFT key, I soon had the needed X drawn on the white background (barring several trial and error attempts, of course).

Next was a sound effect that would accompany the red X. It had to be some kind of beeping sound to let the player know, "You got it wrong!" Thankfully, this was easy enough, because I had recently coded in Bible Games from a book of Bible-based programs. In the main body of the program (the menu if you will) was embedded a series of POKE commands that accessed the Commodore's SID chip to make a low tone, beeping noise. This would work perfectly with the program I was creating. Finding the book of game codes, I located the necessary POKE subroutine and incorporated those commands into my quizzing program. I knew I had hit pay dirt after I typed RUN on the Commodore's home screen; the program passed the quick test. With the wrong answer subroutines in place, it was time to work on the graphics and sounds for the right answer.

Finding a graphic for when the player gets a right answer was easy enough. By using the POKE 53280 and POKE 53281 locations and by adding a FOR/NEXT loop at an interval of 200 jiffies per color, I programmed the Commodore to scroll a few colors across the screen. I chose the first five colors of the Commodore spectrum so as to not to keep the player too bored with images. Again, a quick alpha test on my Commodore showed that the color scroll worked as expected. So far, everything was working as planned. Being away from the BASIC side of the Commodore for a while did not appear to have hampered my work in the language. Now to add some music.

When I went to search for the Commodore User's Guide, I discovered my copy of the manual had gone AWOL. Luckily, I had a couple of Commodore SX64 User's Guides that served just as well. Along with the Commodore 64 Programmer's Reference Guide, I searched out the Sound and Music section of both books and skimmed through the pages, trying to comprehend the method of how a programmer inputted code to create music and sound. The reference guide revealed a quick and simple answer to my dilemma; there were some sample programs that presented short musical tones. Since each sample only contained a slight change from the other, it was quick work to code all the variants of the brief musical interlude into the Commodore. After careful deliberation, I selected a sawtooth version of the music. The entire tune played for no more than five total seconds. I put the musical code, which consisted of POKE commands and DATA statements, into the sample program with music playing first before the color scroll presented itself on screen. Running the sample program as a whole, I was blessed that the entire sample ran smoothly; in fact, everything that I copied and coded went without a hitch. Thankful that I still had my knack in BASIC programming, I saved the sample to a 5.25" disk. By now, it was evening. I had worked approximately five hours on the program. It was time to eat, shower, and get to bed to get ready for the work week.

Time was short. I had to get Pastor Barb to examine the prototype program and get her approval or disapproval of what I had done thus far. The next Sunday I loaded up my Commodore SX64 into the back of my rental car and made a bee line west from my Kansas City, Kansas digs towards Bonner Springs, a short distance of approximately twelve to fifteen miles. I got to the church in record time, and with software and the SX64 in hand, I practically busted into the Fellowship Hall where Children's Church was already in progress. Out of respect to Pastor Barb and her class, I put a curb on my enthusiasm and took a chair in the room quite a distance away from the lesson in progress. Jon, Pastor Barbara's husband, assistant, and my best friend, walked over to me while I was setting down the SX64 and getting myself adjusted. Jon bent down to whisper in my ear.

"What's all the hubbub, bub?" Jon asked.

As quickly and quietly as possible, I told Jon about the events that led up to the moment, that I had a sample prototype of what they asked for, and that I had brought the proper machine to run it on.

"That was fast," Jon commented with raised eyebrows. "How did get it done so quickly?"

I reiterated that this was a sample program and not a complete module of twenty questions and sixty clues. It was just a single question and three multiple-choice answers.

"Great," Jon said. "Right after the lesson is over, you can set up your computer, and we'll look at what you got."

I gave Jon the high sign. He stepped back and joined his wife in helping teach the lesson. I quietly listened to Pastor Barb finish the lesson about David and Goliath – little kid whoops on big bully. My kind of story.

When Children's Church was over and Jon and Pastor Barb said goodbye to the last of the kids, they turned their attentions to me.

"Okay, Lenard," Pastor Barb said, "let's see what you got in such a short time."

"Hurried work is worried work and a job not well done is not a well done job" - Alan Hale, Jr.

I didn't have much but hopefully enough to whet her appetite and allow me to continue with the project. I set up the SX64 and loaded the sample program. When the computer showed READY, I typed RUN and pressed the RETURN key, the sample program displaying on the five-inch screen. Pastor Barb played the game; in a scant few seconds, she had won. We went back into the program, and she checked out what the computer did if the player got a wrong answer. Pastor Barb stood up from her stooped position over the SX64 (the computer was on a low-standing table) and turned to address me.

"I like the way the colors blitz across the screen when a player is correct," she commented, "but I was looking for more design. Sound effects are pretty cool, too. The program needs an opening screen so a player can choose which question they want to answer." She glanced at me. "You've got something going on here," she told me, "but it really could use some outstanding graphics. Other than that, you've got a good start."

I explained humbly that this was my first attempt at writing a game of any kind. Other than my typing in games from a book or magazine of collected games, this is really the best I could do in such a short time or any other time for that matter. Pastor Barbara nodded in understanding.

"With that in mind, I must admit this is pretty good for a first attempt," she confessed. "I would like to see what an entire lesson looks like when it is done. How soon can you have a lesson done so we can run it for the kids?"

My ego overcame my logic at that moment, and without thinking, I told her that I could have a module done in about two weeks.

"Great," she answered. "We'll be on lesson three by then. I'll send you the questions and answers to your email."

As I packed up the SX64 and started to head for the exit to the parking lot, Pastor Barb stopped me. I turned to face her.

"Can you also keep adding the jokes to the lesson answers like you did in the sample? I think the kids might get a kick out of them."

I nodded and assured her that I would do my best to add some humor to the mix. As I stepped out the door and headed towards the rental, I stopped in mid-stride and looked up towards the sky.

"Dear Lord in heaven, what has my mouth gotten me into now?"

After getting home and putting my SX64 away in its cubby, I grabbed the disk that had the sample program which had been presented to Pastor Barb earlier. I put that disk in my 1541 disk drive of my desktop Commodore set-up. I loaded it, and upon the READY prompt displayed on the screen, I typed LIST to again glance over the BASIC code that made up the sample program. As I looked at it, I thought to myself, "Repeat this pattern twenty times, and I'll have a module."

Working in Commodore BASIC, I enjoyed its screen editor; it gave the programmer the ability to copy lines of code by simply changing the line numbers of that particular piece of the code to a different line numbers, thus "copying" the lines of code. This little feature would come in handy when I got serious in copying the graphics and sound effects which announced each right and wrong answer in the program. The challenge now was to write out forty one-line jokes to give the player an easier chance to win.

As a washed up comedian from the 1970s, I had quite the trouble writing one-liners that would make the young player of this modern era giggle. However in about four days, I had all forty wrong answers written out in PRINT statements. Pastor Barbara wanted an introduction screen so the player could choose which question they wanted to answer. Utilizing the INPUT command, I set up a series of IF/THEN subroutines for the player to respond to a numerical selection and to press RETURN. Naturally, this main menu idea had to have a smattering of humor with it as well. With the added gag lines, the player had no idea what the selected question would be. I got this program module written and alpha-tested in a week, just in time for Pastor Barb to beta test it on Sunday.

By now, I had downloaded and started using the WinVICE emulation program for PC and had started converting all the Commodore Christian programs off of the Commodore disks and onto a .D64 image by using the latest version of DraCopy, the swapping program between PC and Commodore. This way Pastor Barb could use a Commodore emulator instead of having me drag my big SX64 around every Sunday. Pastor Barb had enough equipment on the small table, so to add a SX64 to the mix would leave next to no room on the table. This way, with the use of the WinVICE emulator and several .D64 images containing any files that I had written on my desktop C64, the programs could be run on the church's PC and be projected onto the big screen on the wall. Now, the new dilemma was to find the time to code all the necessary questions and answers in two weeks time while I still dealt with my work dilemma. Of course, it would seem that the smiling heavens had that planned already.

I had a tough time using rental cars for my job, paying fifty percent of my daily salary to the rental company so I could work. During this tough financial stretch, at some points of the month I had to take days off from time to time in order to save what little money that I could.

As I sat around twiddling my thumbs, wondering what to do with all this lull time, I decided that I would work on the programs Pastor Barbara wanted. Starting at nine o'clock in the morning, I sat down at the Commodore 64 and started working on the first module of the series, which was the prayer of Hannah and the birth of Samuel in 1st Samuel, chapter one. By one o'clock in the afternoon, I had the aforementioned module done. Still in the groove, I started on module two which was Samuel's call to be a prophet and Eli's death. All I had to do with this module was to come up with new questions and some one-liners for possible answers. Also, with the C64 screen editor’s wonderful ability to copy lines of code, the process was sped up, and it only took me about two hours to type up the second module.

As each module came up to be coded, it seemed that it took a lot shorter time to complete each one. By ten o'clock that night I had five modules completed. Since I was getting into a groove, I decided to call it a night and take the day off the next day so I could code even more modules into existence, especially now that I could get each module coded in about two hours. I did this for three solid days, doing nothing but coding, eating, and sleeping. By the end of one week I had all twenty modules coded and saved to two 5.25" disks, utilizing both sides of each disk. Before I turned in anything, I went ahead and did a final alpha test on a random module of the set.

Things during the alpha test were going well until the program seemed to lock up when it came to playing the "correct answer" music. I went back into the Programmer's Reference Guide and to where I had gotten the sound code, comparing the text in the guide to the code I put on the screen. Everything matched, but after each boot and reboot of the module, it still locked up. Time was short. It was time to call in the big guns.

Robert Bernardo of the Fresno (California) Commodore User Group had been my "go to" guy when it came to confusing code problems, even if it was something that I myself coded and couldn't find the elusive bug that infested the work. I emailed Robert, told him my dilemma, sent him a .D64 image of one of the disks containing about five modules, and asked him to look it over. His skills in deciphering confusing code would be most beneficial at this time.

"Hurried work is worried work and a job not well done is not a well done job" - Alan Hale, Jr.

Robert got back to me almost the same day, "Lenard, I sent your programs off to Atariage and told them your problems and sent them the .D64 image that you created. At first glance, I couldn't find anything wrong with what you have done, and after I ran your programs on my own C64, the problem still existed. This test eliminated the possibility of faulty hardware on either of our ends. I should hear something from Atariage no later than the end of the week."

End of the week nothing! Atariage got back with Robert the next day. In an abbreviated email, Robert told me that, despite the programs needing a complete rewrite, the errors in the codes were very simple to fix. I needed to add one line of code to the DATA statements that produced the sounds:

3625 RESTORE

I knew that RESTORE was a key on the Commodore keyboard, but I didn't know that it was a command as well. Looking up RESTORE in the Programmer's Reference Guide, the manual stated that adding RESTORE to the reading of a series of DATA statements resets the read code back to zero so that the DATA statements can be read over and over again, just as if the computer were reading the DATA statements for the first time. I added the RESTORE command to the offending lines of text and again alpha-tested the game. Everything ran perfectly, and the modules never locked up but went right to the opening selection screen and waited for more input. After adding the RESTORE command to each of the twenty modules, everything was done. Now I needed to take the programs off the C64 by using DraCopy to turn the programs into .D64 images. Then I waited until Sunday to come around so I could demonstrate the finished product to Pastor Barb and see if the programs would meet her approval for general church use.

On the following Sunday at the Children's Church, I was there with a flash drive containing all the programs that I had written, along with a copy of WinVICE for the PC. The church board had been nice enough to buy Pastor Barb a new Windows 11 PC to replace the old Windows 7 machine that she had been running.

(I don't know why many churches expect a children's ministry to run smoothly on hand-me-down products and devices. It's shameful that churches do that to the children and expect them to learn about the Bible and the Christian life when they don't have updated equipment.)

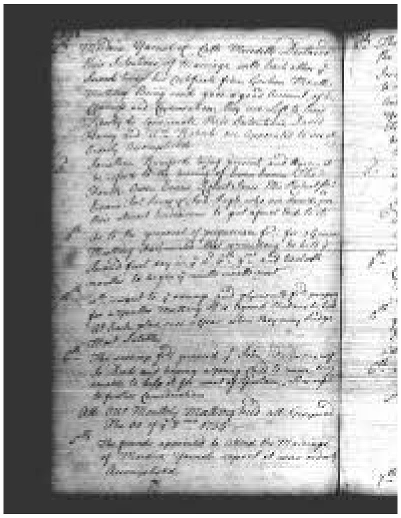
But I digress. When class was over, Pastor Barb allowed me to sit down at the PC and load WinVICE and the newly written programs off of the flash drive. Pastor Barb and Jon watched the big screen as I worked the program from the monitor on the desk. They both laughed at some of the one liners, they thought some jokes were "distasteful" or "questionable," and they wondered whether kids could tell if a line was a joke or was the real answer. Pastor Barb looked away from the screen and addressed me.

"Even though the program seems to be simple, it does do the job that I wanted. There are a few lines that I would like to have changed, so the kids can understand the clues a little better. A big plus is that you were able to reduce all of the Commodore's programs and functions down to one simple flash drive which will work on the PC. I'll implement this in Quizzing Class next week to start having them learn more about the Bible books that they are studying. Good job, Lenard."

Though I didn't get to use an actual Commodore 64 in the class due to the bulkiness of the machine and the difficulty of hooking the Commodore up to the wall projector, it was great to have a form of the Commodore, the WinVICE emulator, on a PC and make it easier for Pastor Barbara and her class to use. But at what cost?

Due to the lack of work with my car still down, my paycheck was well below the mark that was needed to keep the house up and running at peak financial performance. Many bills got pushed back to the next pay cycle, which put many of them into late status which meant late fees and fines in some instances. Even though this happened, I did get the satisfaction of coding a massive work on the Commodore.

Now I am looking forward to next year (2023) and a whole new set of questions to come down from the church's national headquarters. I am also going to take the Atariage's constructive criticism and go over the code and see what lines I can modify to make the program run much more efficiently.



MONTHLY MEETIING REPORTS

By Robert Bernardo & Dick Estel

November 2022

Thankful for each other and that the club and Commodore computers have continued to live more than two decades into the 21st century, Robert, Roger, Dave, Dick, Bruce, and a guest, Roger's brother Aaron, were at November's meeting. Young Michael arrived late but was overflowing with enthusiasm.

There were some interesting topics during our November pre-lunch discussion. We definitely discussed the availability of pizza with peanut butter and speculated on whether or not it included tomato sauce. Somehow, we got onto the subject of explorers, how cats accompanied them on their ships, and how cat litter was invented in 1947. Before 1947, who knows what methods were used for cats going to the bathroom? All the more reason to stay out of cat sandboxes in those days!

On Monday, Robert will be heading to Washington with several pieces of equipment to leave for Ray Carlsen to repair. Ray is cutting back on his repair activities but will take care of loyal, long-term customers, like Robert. After dropping off the equipment, Robert will visit the Seattle Commodore Computer Club. http://www.sea-ccc.org/ Robert has a Commodore PC20-III DOS machine that is not working. Ray cannot work on this product, but a member of the Seattle club has the identical machine, and Robert is hoping he will be able to repair it.

Robert heard from Stephen Jones of the Living Computer Museum in Seattle. Stephen offered to host the 2023 Pacific Commodore Expo NW (PaCommEx) at the "Interim Computer Museum" in the Old Rainier Brewery, a landmark building which had been transformed into business offices and into an event venue.

Robert learned that he will have to wait 'til early December to learn if we can hold the 2023 Commodore Los Angeles Super Show at the Burbank VFW hall, and what the cost will be. He had received a recent donation that reduced the 2022 show deficit to $171. In addition, the club voted to donate $50 which would reduce the deficit to $121.

"Hurried work is worried work and a job not well done is not a well done job" - Alan Hale, Jr.

We make an annual charitable donation each November, this year the club voting to donate $50 to the Central California Food Bank (Fresno). This agency is able to turn a dollar donation into about eight dollars worth of food which is badly needed this year.

Robert had visited Europe in October, and he passed around British newspapers that mourned the death of Queen Elizabeth II and praised ascendance of King Charles III. Our demonstrations got under way with Robert presenting a slide show of equipment at the National Museum of Computing in Bletchley Park, England. We saw familiar Commodore machines as well as products made and sold only in Great Britain.

He presented a new hardware item, the Orange Cartridge, which was plugged into the expansion port of the club C128. The cartridge had what it called "Super-REU" mode, a mode that gave it 16 megs of RAM. It also had a movie player installed. Unfortunately, Robert did not bring a microSD card with the particular movie files, so the movie player could not be demonstrated. At the next meeting, Robert promised to bring the required items.

The big demonstration of the meeting was TheC64 (Maxi). Due to copyright concerns, this new computer could not be called a Commodore, though it looked almost exactly like a brown, full-sized Commodore 64 breadbox. The computer was really Linux computer board emulating the C64 under VICE 2.4. Roger and Robert compared its case with that of a real, brown C64. The badge now said TheC64. The C= key was now labeled TheC64. TheC64 didn't have the traditional ports – user, cassette, RF, serial, expansion, a/v, coaxial power in, nor 9-pin sub-D's for joysticks. It did have 4 USB ports, one momentary power switch, HDMI out, and microUSB-in for a 5V USB power adapter. At the computer's rear edge, it sat a millimeter or two higher than the original brown C64. The keys had almost the same feel as the original C64.

Several months earlier, FCUG member David had given a presentation on TheC64 Maxi, but this presentation was meant to be more thorough. Robert's Dell U2410 monitor could be forced into 4:3 mode, and so, when TheC64 screen was displayed, it had widescreen mode for its "game carousel" but 4:3 mode for the games were played. TheC64 Maxi ordinarily would power up in its game carousel mode where the games were graphically represented with a short description, and the games would be chosen by moving the included Competition Pro-style USB joystick either left or right and then by pushing the joystick button to select. However, in the Tool part of the Maxi (chosen by pushing the fourth accessory button on the joystick, moving the joystick to the Tool graphic, and pushing the joystick button), you could set whether you wanted the game carousel screen or the BASIC screen at start-up, set PAL or NTSC, set C64 or VIC-20 mode, and adjust other settings.

Robert told the story of how he bought TheC64 from the Menkind store in Crawley, England, after seeing it at the Menkind store in Brighton, England. Knowing that it was unavailable on Amazon.com, he had this last chance to buy it directly for 129 pounds.

The rest of the meeting was devoted to playing several of the 64 games built into the machine. Michael eagerly played them, and Robert and Roger tried out the one VIC-20 game, a text adventure, that came with the computer.

DECEMBER 2022

-by Robert Bernardo

At the December meeting, Robert B., David S., and Roger V. attended with Michael C. showing up at the end as the equipment was being disassembled. Robert had brought back hardware repaired by Ray Carlsen; specifically, the hardware was Roger's VIC-20. The computer had a screen color problem which no amount of tweaking on the computer's internal potentiometers could cure. To Ray's surprise – something he had not seen in over his 30 years of Commodore repairs – the problem was a bad 6502 CPU and not the VIC-I display chip. Roger also bought the medium-priced Ray Carlsen power supply for the machine.

Under old business, Robert reminded everyone of the April 15-16 Commodore Los Angeles Super Show 2023. Just as he suspected, the rental price of the Burbank VFW hall had gone up. Instead of the 2021-2022 price of $1,000 for 2 days, it was now $1,200 – still a bargain in the Los Angeles area – and Friday set-up was still free. He had sent in a $400.00 deposit to hold the room. Robert also reported that the red ink of CLASS 2022 was nearly erased, thanks to donations from various benefactors.

In new business, Dave suggested that the club send a thank you card to Ray Carlsen for his years of repairing the computers and peripherals for the club. The motion was made, seconded, and passed unanimously.

Also Dave suggested that Robert contact Dick Estel so that the future of the club's treasury was assured (the monies were held in an account at a local credit union). Robert was a co-signer on the account, but he didn't even know the account number.

Then Robert carried on with the election of FCUG officers for 2023. Roger motioned that the current officers be carried over into the new year. The motion was seconded, and it was passed unanimously.

Finally under new business, Robert reported that he had signed up for two tables at the April 1 Los Angeles Maker Faire. The tables would exhibit vintage computers, not just Commodore and Amiga computers.

With all business accounted for, the members went on with the presentations of the day. On his Ultimate 64, Robert showed two C64 video formats he had recently discovered. One was Koala Video, which ran 160 x 200 multi-color video with synced sound. Under Koala Video, he ran the "Commodore Ads," a compilation of 4 Commodore commercials. The picture was grainy, and the sound was minimally better than other C64 video formats. Then Robert showed an older Commodore video format called BR-TV. Under BR-TV, he ran "King Kong," the original 1933 movie which had been abridged to just the important scenes, coming in at about 6 minutes. Again the picture was grainy, the synced sound scratchy, but the black-and-white scenes had been colorized in C64 colors.

On the U64, the members tried out the following games: Santa's Workout, Xmas Demo 2, Captain Ishtar, Terrestrial, Petscii Invaders and Usagi Yjimbo (a game based on the comic book series of the same name).

On the Amiga 600, the members briefly checked out the Arcade Game Selector II, a graphical file browser which showed a description and thumbnail picture of each WHDLoad game on the hard drive. The AGS II presented a very colorful screen, and it was easy to use.

On the C128, the members tried out the CBM's Commodore 128 Christmas Demo disk, a disk that had 4 programs which advertised the capabilities of the machine. The first program, the C128 Christmas Demo, seemed to be almost the same as the well-known C64 Christmas Demo in that it had the same 40-column music and PETSCII animation, the only difference being in the ending text that urged the user to buy the C128.

To finish the meeting, there was more C128 goodness – the new Eye of the Beholder (a C64 graphical adventure which had better speed on the C128) the classic Harmonizer 128 from Loadstar (but it didn't work), and the classic Story Writer 128 from Loadstar (a multi-window story outliner which helped a writer create stories)

Commodore LCD resurrected in 2022

by Guest Contributor Dave McMurtrie

The Commodore LCD computer was a brilliant idea and came very close to production, but it was unfortunately one of the casualties of Commodore's cash crunch, [Commodore] losing $113,900,000 in fiscal year 1985. Commodore cancelled the project, and only a few of these machines exist today in the hands of the Commodore folks who worked on the project. Thanks to some awesome work that is being done by Steve Gray, the LCD may see the light of day after all! Steve is working on a project to recreate the LCD.

The Commodore CLCD Project

by Steve Gray

INTRODUCTION

This is a project to recreate a "Commodore LCD" computer (aka CLCD). The CLCD was a prototype laptop computer that was in development in 1984/85 and reached pre-production stage. Unfortunately, it was cancelled in favour of the C128 computer. Only a handful of these computers were made and few survived. Luckily, system specifications, firmware, and schematics were saved. Firmware from Bil Herd's early REV1 prototype were dumped; however, the schematics are from the later REV 2, and there were some changes. No REV2 firmware has been archived as of yet.

For a long time I have been interested in this machine, as it had the potential to be a game-changer. Bil had managed to save the schematics, but until recently one page was missing. Finally, the missing page was found, and the schematics scanned and finally released in 2022, which motivated me to start this project. This is a multi-faceted project to create a CLCD work-alike machine, not in an exact duplicate, but in a form that is more up-to-date, using newer still-available chips. Since the LCD panel, custom gate arrays, keyboard, and case are not available we need to find alternate solutions. This project is also a way for me to learn CPLD programming, as well as using PLCC chips, and modern implementations of the CPU and VIA chips.

All pcbs will be designed with Kicad. Full design files will be made available when they are usable. I will probably use Sketchup to design the 3D case as I am most comfortable with that.
This will be a team project with myself and Mike Naberezny. Mike has much more electronics experience than I do and also has experience with designing and building Single-Board Computers. Mike has also done extensive disassembly of the CLCD KERNAL, which means he can write test firmware to test each component of the pcb.

STATUS OVERVIEW

-
- MAINBOARD: Boots and runs KERNAL.

"Hurried work is worried work and a job not well done is not a well done job" - Alan Hale, Jr.

- WORKING: Reset, CPU, RAM, ROM, DPRAM, VDC, Font ROM, Centronics, ACIA, IEC
- UNTESTED: RTC, Keyboard Interface
- KEYBOARD: Inverted-T cursor version built. Untested.
- KEYCAPS: Design complete. Keycaps arrived.
- FIRMWARE: Patched for VDC. Working.
- LCD PANEL: Testing for suitability.
- CASE: A full case has been designed and printed, but still needs some work.
- CPLD: Not started. Using EPROM as PLA temporarily.

MAINBOARD

The main PCB schematics were studied, and I determined that we could make a mostly compatible design with modern versions of the chips such as the CPU and VIA's. However, the Custom Gate arrays, which include the MMU and the LCD controller, would need to be re-created. Since the LCD panel is not available, it didn't make sense to try re-creating it. Instead, the video subsystem has been replaced with a Commodore 8563 VDC chip from the C128 computer, and a dual-port SRAM that also doubles as the system's main 32K ram. The VDC chip should be able to emulate the 6x8 font format as well as the 128-bytes per screen line that are unique to the CLCD machine.

The CLCD used a 65C102 DIP cpu. These are hard to find now. Instead I have replaced it with a 65C02 which is fully code compatible and comes in PLCC format in speeds up to 14MHz. It may be possible to clock the machine at higher speeds with the use of more modern IO, RAM, and ROM chips.

I have included 512K of RAM on top of the 32K DPRAM which will act as extended RAM memory. I have also included 1MB of ROM that can hold all of the KERNAL, MENU, and APPS that were in the CLCD prototype. There is more RAM and ROM than on the original machine, but perhaps the additional memory can be utilized in the future.

There are two VIA chips just like the original. These control Keyboard, IEC, RS-232, centronics, RTC, and barcode scanner functions. The CLCD had an internal modem, but it doesn't make sense to include it. The barcode port will not be included; however, there is a header. I include a video port, serial port, centronics port, IEC port and internal IEC header, power jack and switch, and a reset button along the back of the pcb in the initial design. Future designs may change.

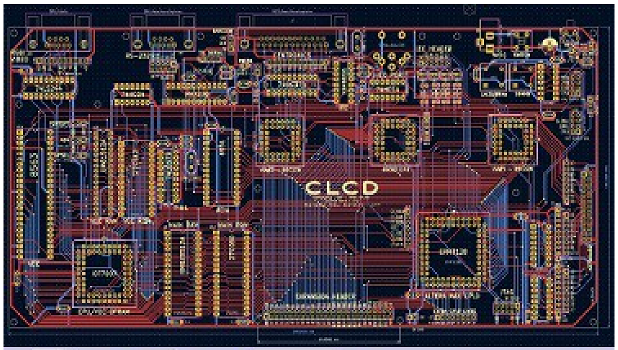
The CLCD's video was designed to drive a custom LCD panel. As such, it's not possible to duplicate it. The LCD panel was quite advanced for it's time; however, technology has come a long way since then. I am familiar with the old 6545 CRTC controller as found in the PET/CBM line. The C128's VDC chip is like a CRTC chip on steroids. I looked into it and discovered it is capable of handling 6x8 pixel fonts and 128 byte per row screen formats. The VDC chip poses some challenges as this chip was designed to use local video RAM that is accessible only through the VDC registers. This would not work with the CLCD, [but] the solution I came up with is to use a 32K dual-port SRAM that can be accessed by the VDC and CPU independently. As the VDC has a 64K address space, I have mapped the FONT ROM and some RAM into the upper 32K. This means the font pointer just needs to point to it during initialization and will be hidden from the CPU as in the real machine. You can also select RAM to be there so that you can program your own fonts, or do graphic screens. Lastly, the VDC supports colour, and so we could theoretically add that capability in the future.

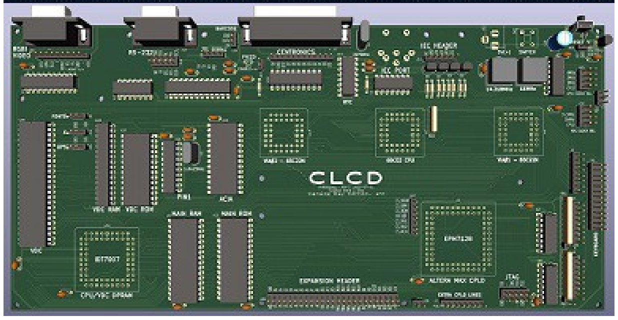
The original CLCD machine was battery powered. This function will not be implemented for the first pcb.

STATUS: PCBs have arrived and I am populating the board and gathering parts. Mike also made a set of boards and has been populating each section and testing each section to make sure it works before moving to the next section. Mike has been identifying mistakes and possible issues with my pcb design and jury-rigging the fixes on the board.

WORKING: Reset, CPU, RAM, ROM, DPRAM, VDC, FontROM, Centronics, ACIA, IEC

UNTESTED: RTC, VDC Upper RAM, Keyboard Interface





KEYBOARD

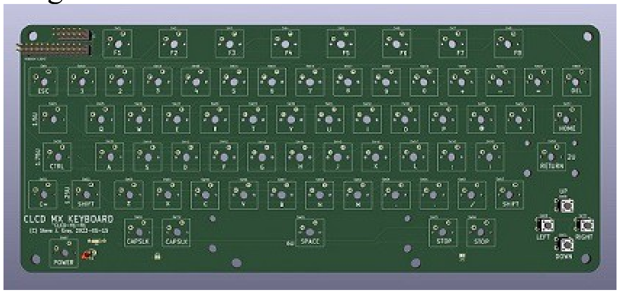
The CLCD keyboard is unique for Commodore. It is compact and has 8 function keys and a 4-way cursor block. In many ways, the keyboard is similar to the Plus/4 keyboard but is less wide. The keycaps are low profile. Electrically, the keyboard has an 8x8 main matrix and an additional separate modifier matrix. Also unique to this keyboard is the way it's connected to the computer. It uses two shift registers to read the keys serially. This allows the keyboard to be read with less lines and also reduces the key ghosting that is common when pressing multiple keys at the same time.

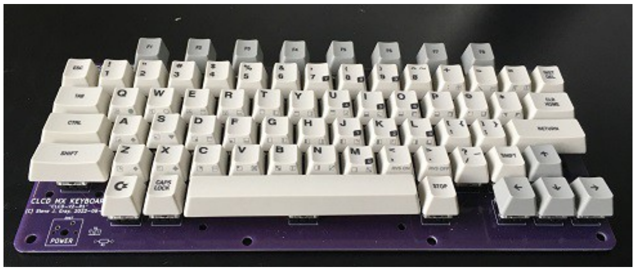
I have designed several different MX Replacement Keyboards for Commodore machines, so doing the CLCD keyboard was pretty straight-forward. I decided to make two designs, one with arrow keys like the Plus/4 and a second with inverted-T cursor using MX switches. I should note that the REV2 keyboard is connected differently than the REV1 keyboard. This has been confirmed by looking at the REV1 disassembly. My keyboard is based on the REV1 design so that it'll work correctly with the existing CLCD firmware.

For keycaps I decided to use MaxKeyboard.com again like I did for the V364 project. They offer a Custom Color Keycaps service that uses a dye-sublimation process. Each key can be a different colour, from a choice of over a dozen colours. You provide the vector artwork for each key using their keyboard templates. You can put basically anything you want on each key. To recreate the CLCD keyboard, I used a high-resolution scan of an actual bare CLCD keyboard provided by Bo Zimmerman. I matched the font and symbols as best I could and created the graphics symbols all in vector format. I selected the 104-key layout in order to get enough keys of the proper size and key row. There were extra keys which I used to make some alternate versions of some keys (i.e., letter keys without the numeric pad) and a few other keys. I used beige keycaps for the CLCD keys, but I also did an extra set of F-Keys in rainbow colours so I could have samples of the different colour keycaps that they offer. I also made some extra C= keys for fun!

"Hurried work is worried work and a job not well done is not a well done job" - Alan Hale, Jr.

STATUS: I had [designed] a few keyboard pcb's made of the inverted-T design and they have arrived. I have populated one with standard MX switches. Custom cherry mx keycaps have been designed and have arrived.



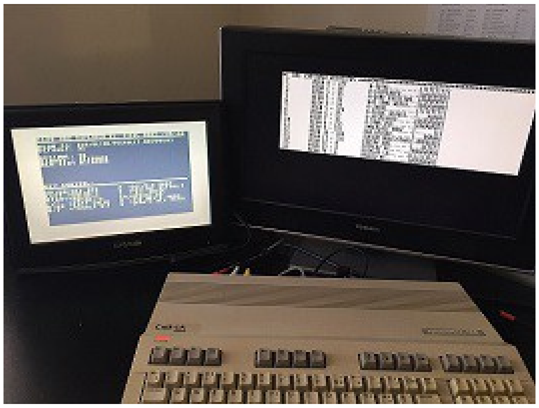


LCD PANEL

Since the original LCD Panel is no longer available, I looked for a panel about the same size. I found a colour panel that is almost the exact size, but not with the same pixel resolution. It came paired with a controller that supports HDMI, VGA, and composite inputs. We will need some type of conversion to get the RGBI output from the VDC chip to a format that will be accepted by the controller. I would prefer to avoid composite [in order] to get the best display possible. I hope to do some experimenting using a C128 and some custom software to simulate the CLCD screen output. At this point I don't know if that will be possible. One other solution is to use a 4:3 screen and make the display taller or less wide, but that would change the look and feel of the machine.

I have written a program called VDC Explorer (VDCX) that lets me play with the VDC registers to generate different screen parameters, which can be used to initialize the VDC depending on the installed panel.

STATUS: Purchased. Currently testing for suitability.

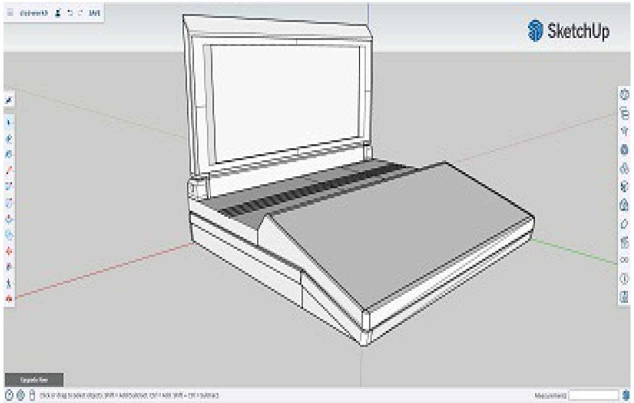


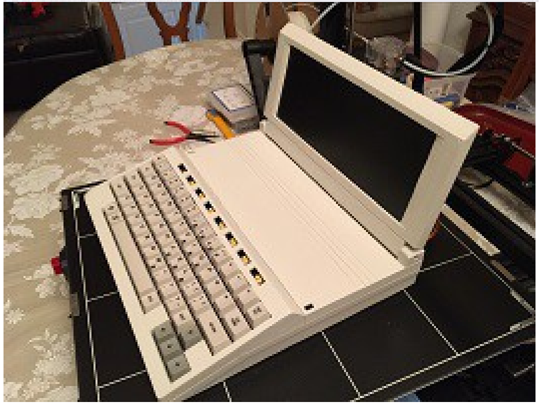
CASE

The CLCD case is also unique. It has a popup screen with keyboard cover attached. It's kinda styled like the 64C and C128 but with the screen added. My goal is to design and 3D print a replica case. The current pcb is also fairly small so it's also possible to house it in a standard C64, Plus/4 or similar case for debugging the design.
I have designed a preliminary case using Sketchup. I used pictures and videos of Bil's prototype, pics of Jeff [Porter’s] pre-production machine, and dimensions provided by Bil and from the [Commodore] brochure as reference. A full-size case has been printed using an Anycubic Kobra Max printer, which is big enough to do each piece full size. This case holds the pcb, keyboard, LCD panel and controller, as well as an RGB2HDMI. Additional work is needed to finalize the mounting and case fastening.

STATUS: In progress

"Hurried work is worried work and a job not well done is not a well done job" - Alan Hale, Jr.





FIRMWARE

The REV1 firmware was disassembled by LGB, and he created a CLCD emulator. Much information can be found on his CLCD pages. The firmware was further disassembled and commented by Mike Naberezny and is currently in a state that it can be re-assembled into a working binary. This source code will be used in this project but will need to be patched to support the VDC video chip. Because of this, the re-created machine may not be entirely compatible with the original but hopefully will be compatible enough to be workable.

The CLCD has an interesting architecture where it maps in large chunks of ROM code as needed. The KERNAL is a full 32K in size. Luckily, there are large empty areas for implementing patches, etc. In fact, there are large chucks of code from the C128 left in there presumably from firmware development.

The CLCD has an 80x16 format display, which is a bit odd for today. The VDC chip can be programmed for the same format, but we have to adjust it to conform to current display devices. It's also possible to program the VDC to use double-height fonts that can stretch the display to fill the screen top to bottom.

Mike has been working on the firmware. He has been creating small test KERNALs using code from the real KERNAL to debug the pcb design and confirm functionality.
STATUS: Working with only a small patch to initialize the VDC.

CPLD DEVELOPMENT

The MMU gate array has been replaced with a modern (well, relatively speaking) Altera MAX7128 5V chip. This will be the main chip that provides all the memory mapping, custom control registers, and chip select lines. I have never done any CPLD programming, so this will be the most challenging part of this project. The current pcb has headers for all the CPLD lines for debugging purposes.

STATUS: Research mode.

EPROM-PLA

I have made a small pcb that mounts on the expansion header and chip-select header that implements an EPROM-based PLA-type circuit for initial testing, with simple debug firmware. Mike has written a python program that generates the binary for the EPROM.

STATUS: EPROM-PLA board is working. A few mods were needed, but it is now generating valid chip select lines for the main pcb.

"Hurried work is worried work and a job not well done is not a well done job" - Alan Hale, Jr.

RESOURCES

The following are resources used in this project:

- Mike Naberezny's CLCD page – http://mikenaberezny.com/2008/10/04/commodore-lcd-firmware/
- LGB Online CLCD emulator and info – http://commodore-lcd.lgb.hu/emulator.html
- Kernal disassembly – https://github.com/mnaberez/clcd
- CLCD schematics – www.zimmers.net/anonftp/pub/cbm/schematics/computers/commodore-lcd-schematics.zip
- My MX keyboard designs on Github – https://github.com/sjgray/CBM-MX-Keyboards

Kudos also to Mike Naberezny who has actually got the motherboard to boot by fixing some of my design errors while I was waiting for parts to arrive. The machine boots to the machine language monitor since the MMU functionality has not been implemented yet, but I hope to work on that soon. This is a personal project of mine using the VDC chip in place of the custom LCD controller and panel so that we can use modern parts to get a functioning system. Mike is going to be working on getting a vintage LCD panel to work, but there won't likely be one exactly like the CLCD panel available.

Club Officers

>>--> Officers and Keypersons <--<<

President ………………………………	Robert Bernardo
Vice-president ………………………………	Roger Van Pelt
Secretary/Treasurer ………………………………	Dick Estel
The Interface Editor ………………………	Lenard Roach
Librarian ………………………………………	Roger Van Pelt
Club equipment ………………………………	Roger Van Pelt
Meeting place reservation ………………	Dick Estel

-The Small Print-

The Fresno Commodore User Group is a club whose members share an interest in Commodore 8-bit and Amiga computers. Our mailing address is 185 W. Pilgrim Lane, Clovis, CA 93612. We meet monthly in the meeting room of Panera Bread, 3590 West Shaw, Fresno, CA. The meetings generally include demonstrations, discussion, and individual help.

Dues are $12 for 12 months. New members receive a "New Member Disk" containing a number of useful Commodore 8-bit utilities. Members receive a subscription to The Interface newsletter, access to the public domain disk library, technical assistance, and reduced prices on selected software/hardware.

Permission to reproduce content of The Interface is granted provided credit is given to the source, and when identified, the author. Club members are encouraged to submit articles, tips, or ideas for articles.

Disclaimer – The club, its officers, members, and authors are not responsible for the accuracy of the

"Hurried work is worried work and a job not well done is not a well done job" - Alan Hale, Jr.

contents of The Interface or the results of actions based on its contents.

Our disk library contains over 3,000 public domain programs for the C64 and C128. Members are entitled to copies of these disks at no cost if a blank disk is provided. We do not deal with pirated, copyrighted, violent, or obscene programs. Please call our attention to any programs found in our library which may violate these standards.

"Hurried work is worried work and a job not well done is not a well done job" - Alan Hale, Jr.