

# The Commodore CLCD Project



## INTRODUCTION

Welcome! 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 No REV2 firmware has been archived as of yet.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.
- **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 funtions. 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.

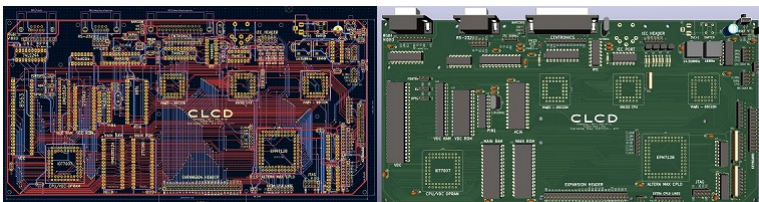
The CLCD's video was designed to drive a custom LCD panel. As such, it's not possible to duplicate it. The LCD panel was quite advanced for it's time however technology has come a long way since then. I am familiar with the old 6545 CRTC controller as found in the PET/CBM line. The C128's VDC chip is 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, however 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 jerry-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

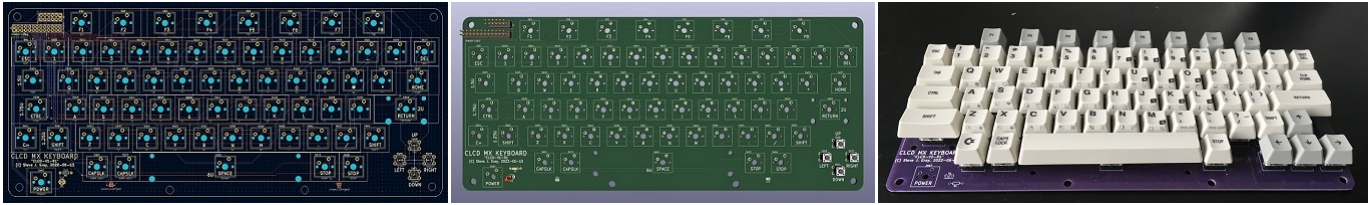
The CLCD keyboard is unique for Commodore. It is compact and has 8 function keys and a 4-way cursor block. In many ways the keyboard is similar to the Plus/4 keyboard but is less wide. The keycaps are low profile. Electrically, the keyboard has an 8x8 main matrix and an additional separate modifier matrix. 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 Printed 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 (ie: 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!

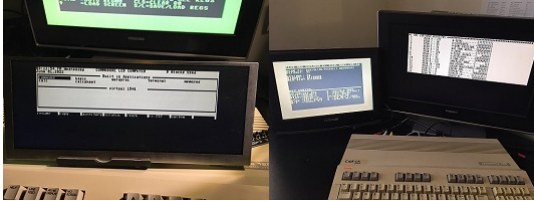
**STATUS:** I had a few keyboard pcb's made of the inverted-T design and they have arrived. I have populated one with standard MX switches. Custom cherry mx keycaps have been designed and have arrived.



## 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 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.

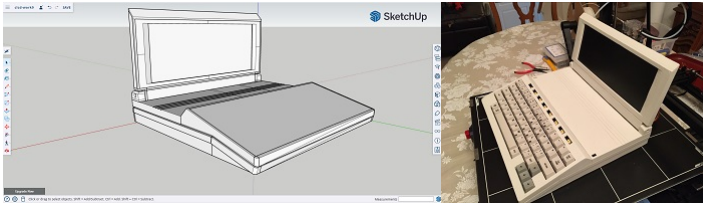


**STATUS:** Purchased. Currently testing for suitability.

## 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's pre-production machine, and dimensions provided by Bil and from the 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

## 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 chunks 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 eeprom.

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

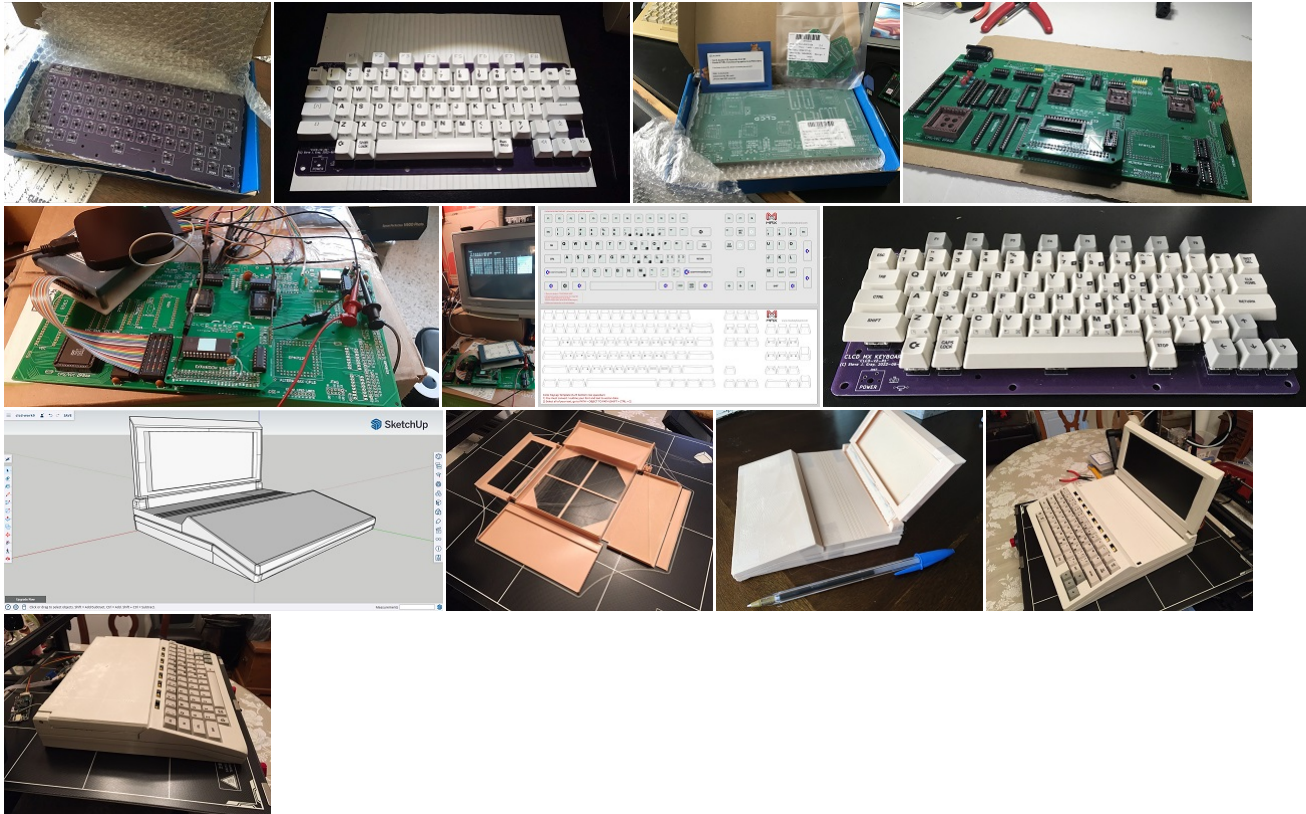
## RESOURCES

The following are resources used in this project:

- [Mike Naberezny's CLCD page.](#)
- [LGB Online CLCD Emulator and info](#)
- [Kernal disassembly](#)
- [CLCD Schematics](#)
- [My MX keyboard designs on Github.](#)

## PICTURES

Some pictures of the project development and results:



## HISTORY

- 2022-06-21: Initial Web Page.
- 2022-07-01: Sent gerbers to JLCPCB.
- 2022-07-10: PCB's arrived.
- 2022-07-26: Add progress info and pics.
- 2022-07-29: Add pics of working system.
- 2022-07-30: IEC working. Custom Keycaps ordered!
- 2022-08-22: Custom keycaps arrived.
- 2022-10-15: Update case info and pics.
- 2022-11-13: More case pictures.

Created: June 21/2022. Last updated: Nov 13/2022, 9:30pm EST

Send comments or feedback to [Steve Gray](mailto:sjgray@rogers.com)([sjgray@rogers.com](mailto:sjgray@rogers.com))