



THE EDITOR'S GODZILLA

by Lenard R. Roach

GAMES IN TOW – Stock Market

(This is part of an ongoing series that will review some of the games I plan on presenting half way across the country at the Commodore Los Angeles Super Show, provided we get to have a show at all. With this pandemic going on, it will be hard to tell until we actually get closer to show time.)

Part Three

“The Dow is up 245, the NASDAQ is up 112, and the S & P is up 54.” I think that just about everyone has heard similar phrases announced at least once an hour during the news on television or radio, and the Commodore 64 offers its own version of playing the stock market in the game reviewed below.

“Stock Market” is one of those type-in games that was found many moons ago in one of the popular Commodore magazines of the day. No joystick is needed to play the game. Everything is operated by punching in the necessary data on the keyboard.

In setting up the game, the program asks for the number of players and the names of each person. After the player types in this information, it also asks how many “rounds” each session will go.

The player can choose between one to three rounds. This selection of rounds will set the number of transactions that go on between rounds. I've never figured out how many transactions there are in a round, but from observation I do believe it is three. The selection of rounds cannot be changed once the decision has been made at the start and will carry on throughout the entire game. For example, if the player chooses three rounds between trading, then three rounds will be the set format throughout the entire game.

As gameplay starts, you are given five thousand (whether dollars or shares I do not know but by the way the game is set up, one would assume dollars) and six ways to divide it: gold, silver, industry, bonds, grain, and oil. With “Stock Market” you must start off with some stock in your portfolio, but you can only buy them in increments of 500. The program's strategy is to put 500 into each category and leave the rest in its bank while you, however, can choose how much you want to put into each stock. I took the bold approach and put all my money into various stocks and didn't leave anything out. After observing how the program played the first round and after getting to see the transactions, I think the program had the better method of play. I had to wait several rounds before I could buy any more stock while the game went immediately into buying and selling. In “Stock Market,” no matter how many players are playing in the game, your object is to outguess the market and try to beat the program in making the most money while, at the same time, try to beat others who are also playing the game. Translated: Get to first place.

At the left of the screen, just a little towards the bottom, the player gets to see exactly what the stocks are doing at the precise moment the transaction happens during the progression of the game. However the graph display never changes during the rounds. There are only three ways the stocks can proceed: stocks can go up, stocks can

go down, or stocks can pay a dividend of a set increment. The dividends go into the player's bank to build up a cash reserve. The stocks that the player purchased stay in play until the player decides to cash them out. All cash out of stocks go directly into the player's bank account and can be used to purchase different stock or sit in the bank until (what the player may consider) a more opportune time.

After the set rounds are played out, the game asks if you want to play another set of rounds. If so, you press Y, and the game prepares to start the round with the transactions collected from the last round. The Y sends the game to the trading screen where you can buy, sell, or do nothing and let others have their chance to trade and move onto the next round. Press N and the machine goes into BASIC, ready to load the next project.

When I went into the game, I didn't have enough money to buy anything after the first set of rounds so I didn't trade. I didn't follow the market closely enough and made poor choices. I know which stocks had a dividend and which stocks went up or down. I really should have sold half my stock and redirected it into my bank; I should have just waited to see what the market did in the next round. Since this game was written in the 1980s, oil would have been the big money maker in those days, but it never crossed my mind at the time.

Like the real Wall Street stock market, be sure to buy low and sell high; however, sadly you don't get to see any kind of graph display until trading actually starts at the beginning of a new round. If you are not following the market closely while gameplay is under way, then you will start making mistakes when it comes to your turn to actually trade, i.e., which stocks to buy, which stocks to sell, or whether to do nothing and let the game have its turn at the market. The aftermath is that you start losing money by shear trading in bad, unyielding stocks. This is where the game

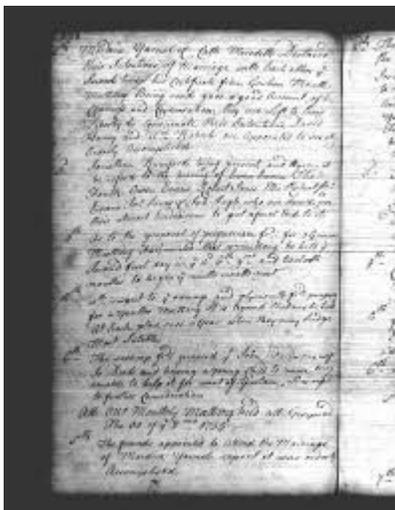
has the advantage since its programming can "see" the graph during gameplay. I also suspect that the game knows exactly what will be happening in the next set of rounds since it is the master controller of everything that is going on in the market. This also could be my paranoia of always constantly losing to the Commodore 64 whenever I sit down and play the game. That paranoia hasn't changed at all even after rebooting the game for this review and losing my tail to the computer ... again.

"Stock Market" can be a never-ending game of outguessing the market and trying to make the most before you either get sick of how the game is progressing and press N to quit, or just get tired and press N for the same results. The lack of a visual graph during gameplay is a negative. However, if you'd like to play a strategy game that allows you some time to think before you actually have to make a move, then "Stock Market" is the game for you. Most video games require quick thinking or a quick hand to make things work to the player's advantage (sometimes a little of both). When I'm at my sharpest, I could usually play the faster games, but if you need a warm-up game to get you to that point to play those faster games, then consider "Stock Market."

"Stock Market" is one of the games I will present at CLASS. Please stop by my table and have a look at this game on display.

○ NEXT
ISSUE

Despite enemies from without and problems from within, the next issue of The Interface should be slated for publication sometime around Saint Patrick's Day, March 17th, 2021. Please keep in contact with FCUG staff for any possible changes.



MONTHLY MEETING REPORTS

by Robert Bernardo & Dick Estel

November 2020

Where was the electrical outlet? In our first FCUG meeting since February, Roger and I had arrived early to set up the computer equipment in the patio of the Panera Bread Restaurant. The Covid pandemic had canceled all of our other meetings for the year. However, for October and November there was a relaxation in the restrictions, and we decided to chance a meeting.

In October we had had our club “picnic” lunch in the patio of a local Mexican restaurant. Even though indoor dining had opened up at 25% capacity at the time, we decided that eating outdoors was safer. We decided that the November meeting would follow the same tack, i.e., having a meeting outdoors would be safer than having it indoors.

Yet, to the surprise of Roger and me, the Panera Bread patio electrical outlet was gone... sealed off with a metal plate. I figured that the homeless were getting into the patio, and that was the reason the outlet was closed off. Thinking quickly, we decided to hold the meeting indoors.

I found some socially-distant tables near the front entrance, and that’s where we placed the minimal set-up of an Ultimate 64, HDMI monitor, and 1571 disk drive. Why the minimal set-up? Because our time was limited to 90 minutes, the new rule at Panera Bread during this time of Covid. We had to get in quickly and then leave quickly.

Member David came in a few minutes later, and we had to wave him into the restaurant instead of into the patio.

After we ordered and received our food, we settled down to small talk. But not too much small talk, because of our time limit. As we finished the last of our food, I called the meeting to order. There was no old business, but for new business we had two items on the agenda. First off was our annual donation to a charity. In absentia, treasurer Dick had recommended that we donate \$50 to the Central Valley Food Bank. Motion made, seconded, and passed. Then I made the motion that we have the same roster of officers for 2021 as we had for 2020. Seconded and passed.

I mentioned to the group that in all the months we hadn’t had meetings, I was still collecting software. Here was a chance to show some of it in the brief meeting.

Not really software but a hardware update, I showed the new Turbo Mode setting of the Ultimate 64. In Turbo Mode, the user can pick CPU speeds from 1 MHz. up to a blistering 48 MHz., making the U64 the fastest implementation of a C64 ever.

To show off the 48 MHz. speed, I tried to run the latest 3D Construction Kit updated game, A Chance in Hell. However, it was taking too long to load, and I gave up. It would have to be presented at another time.

While we had been eating, Breadamp v.05, a music player for the U64, had been playing in the background. Now we had a chance to look more in-depth into the features of Breadamp. First, on the demo playlist was the THX audio opening for movies, played at 22 KHz in 16-bit audio. The other 3 music samples that were played topped out at 15 Khz in 8-bit audio. Very clear sounding and not rough like the usual C64 1-bit or 4-bit audio.

Because the U64 had been configured to display PAL video, we were able to watch 3 of the 170+ novies collected on SD card. Having been around since 2012, Commodore novies were composed of up to approximately 760 still-frame NuFLI (Flexible Line Interlace) images strung together to make a video. The collected novies were either silent or had non-synced audio. (Well, there was one with synced audio, but it seemed that for those creating novies, syncing audio was difficult.) We watched the novies: C64-WebungRemastered 2019 (two Australian Commodore commercials put together), A-Team (opening credits of the iconic t.v. show), and TMS-JCN (video captured of a demo created on an Amiga and other computers).

Finally, we had to play a game. It was “Wreck It, Ralph,” a new C64 game just released a few months ago. It was a marvelous implementation with bright colors, beautiful characters, smooth movement, and good music. Roger immediately sat down to play the game and knew what to do, commenting that he had seen the animated motion pictures based on the game.

While we carried on with the meeting, a young woman on the other side of the nearly empty dining room had been watching us. Finally, she spoke up and asked what we were doing. Roger explained about our club, though she didn't seem to know what Commodore was. It was only later that I realized

that we should have invited her to our next meeting (whenever that was going to be).

We had come to the end of our limited time, though no one really came to push us out. We disassembled the equipment, and before we carried it out to the car, I opened up a box of C= stuff that Ray Carlsen gave when I visited him in October. Almost all of it was VIC-20 software and documentation neatly packed in plastic bags. David got a Super Graphix Jr. printer interface. I handed over seven VIC-20 cartridges to Roger for him to examine. His expertise would come in handy for these rare carts which looked as if they were meant to be used by ham radio operators. All of the carts had typewritten labels from the company, “Kustom Computer Response, Rt. 1 Box 57A, Helotes, Texas 78023,” with the programmer identified as Harry M. Crouse. The carts came with names, such as ARRLDXTEST and CQDXTEST (a few days later, Roger told me that ARRL stood for Amateur Radio Relay League).

The carts could be a subject of a future meeting. However, because of the pandemic, a few days after our meeting, indoor dining in Fresno was closed, and so, our next meeting is delayed until indoor dining comes back. That probably won't be until some time in 2021.

December 2020

No meetings were held in December due to Los Angeles county closing up indoor eating facilities until further notice. Please contact any FCUG staff member for any concerns or for the latest updates.



Fond Computer Memories

By Terril Yue Jones
Los Angeles Times Staff Writer

June 27, 2006

(From Robert Bernardo, “I vaguely remember a LA Times reporter visiting us years ago, but for the life of me, I can’t remember that he ever published any article about FCUG. Then in a recent e-mail, I was provided a link to this article that proves that it was published! Here is the unedited article plus an addendum (but no photo)).

FRESNO —

Robert Bernardo spent a week this spring traveling the Pacific Northwest, trying to save part of yesterday’s future.

The high school English teacher swung through Portland and Astoria, Ore., and then on to Ethel, Wash., to drop off a collection of antiquated computers -- a PET8032, three VIC-20s, an SX-64 portable and a Commodore 128D.

Then on his way home to the Central Valley town of Visalia, Bernardo packed his white Crown Victoria with three more SX-64s, boxes of software and a couple of printers.

With any luck, this agglomeration of decades-old circuit boards and dusty disk drives will allow Bernardo to reboot a handful of computers made by the long-defunct Commodore Business Machines.

In an era when a home computer’s power is measured in gigabytes, Bernardo still counts kilobytes as a devoted Commodore user 12 years after the last machine was assembled.

Once the largest personal computer maker in America, the company behind the VIC-20 and the Commodore 64 introduced millions of people like Bernardo to the digital age. The company went out of business in 1994, but its legacy survives in dozens of Commodore clubs around the country.

Bernardo presides over the Fresno chapter.

Never mind that the VIC-20 has so little usable memory -- just 3.5 kilobytes -- that it can store only a couple of pages of text in its buffers. Or that Commodore hardware was notoriously clunky and buggy. Bernardo still manages all his e-mail on a 1980s-vintage Commodore 64.

“I’ve never considered the Commodore obsolete,” Bernardo said. “I can still do many things with it -- e-mail, browse the Web, word processing, desktop publishing and newsletters. I still do games on it: new games that are copyright 2006, ordered from Germany.”

Like classic car fans, Bernardo and other Central Valley Commodore devotees lug their gear every month to the Pizza Pit restaurant and put the hoods up, so to speak. For many, a Commodore machine was their first computer. They cherish their machines the way some guys pamper their high school hot rod.

The tinker mentality pervades American culture, from guys who fix their lawn mowers to computer

geeks who build the next big thing in their garages. Commodore clubs are “about preserving a particular era in computing -- just showing that you can make it serviceable takes ingenuity,” said Robert Cole, a professor emeritus of technology management at UC Berkeley’s Haas School of Business.

Commodore computers are rudimentary enough that enthusiasts with a little technical know-how can repair them themselves. They also can be programmed with relative ease using the BASIC (Beginner’s All-purpose Symbolic Instruction Code) computer language. Linus Torvalds, the creator of the popular Linux computer language, cut his teeth writing code on a VIC-20 in the 1980s.

“It wasn’t just an appliance. I liked it because it was open and it invited you to play with it,” said Mike McDermott, a Commodore fan who co-founded a website that ranks building contractors. “You didn’t just do what it told you. It invited you to tinker with it. They really did encourage you to go write programs for it.”

And that, in turn, made people passionate about the quirky machines.

Bernardo, who sometimes sports a button that reads “I Adore My 64,” says that every room but one in his three-bedroom house contains Commodore. In the other is his “Star Trek” collection. But there is crossover between his dual passions. His prized possessions include six pieces of Commodore hardware and software signed by “Star Trek” star and former Commodore pitchman William Shatner.

If Bernardo and his ilk keep the memory of Commodore alive, they also may hold the key to its future. The Dutch company that owns the Commodore name is planning to resurrect the brand in the United States with devices that act as digital entertainment centers.

“The Commodore 64 was the biggest-selling computer in the world,” said Patrick Olenczak, vice president of global sales for the company now called Commodore International.

But that fan base can have drawbacks.

“It’s going to be difficult to fulfill their expectations of being a computer company because we’re not,” Olenczak said. “What we’re doing is bringing new forms of computing into the living room.... We are not into computing the way we used to be.”

And Commodore used to be in computing in the biggest way.

Few companies illustrate the ruthless evolutionary efficiency of the high-tech economy better than Commodore. Founded in 1959 as a typewriter company by Polish immigrant Jack Tramiel, it later moved into adding machines and then calculators.

Commodore purchased a small chip foundry and built computers around the processors it manufactured itself, the first being the PET, Commodore’s first desktop, introduced in 1977. In 1981 came the VIC-20 that could do color graphics and generate simple music.

The company’s biggest hit was the Commodore 64, introduced in 1982 with 64 kilobytes of memory, high-resolution graphics and an impressive sound synthesizer. It was followed in 1985 by the Commodore 128 upgrade and the Commodore Amiga, a desktop with phenomenal graphics at the time.

But in the late 1980s and early 1990s, IBM Corp.’s PC clones gained supremacy.

Tramiel was known for aggressive advertising. But he also took manufacturing shortcuts that sometimes put dud computers on the market.

“Jack encouraged the environment where shortcuts were overlooked and rewarded,” said Bil Herd, the chief engineer of the Commodore 128. “The attitude was get it under the Christmas tree -- there is always time for them to return it for service in January.”

Tramiel was ousted from Commodore in 1984.

Commodore found itself expanding in too many unprofitable directions without Tramiel’s iron-fisted stewardship, and although the company had a few subsequent hits, such as the graphics whiz Amiga, it also had a number of costly flops that forced cuts in the workforce and closure of plants.

Reached at his home in Monte Sereno, near San Jose, Tramiel, now 78, would not comment on the business while he was running it or afterward, but allowed in a brief conversation that he was “very happy” that enthusiasts kept the Commodore name and machines alive.

“Today’s computers are definitely more advanced than Commodores,” Tramiel said. “But at the time it was the best computer for the money, because I was building a computer for the people at a price everybody could afford.”

Commodore dissolved in 1994, and its name went through a succession of owners. In its place rose Apple Computer Inc., Dell Inc., and Hewlett-Packard Co., and industry experts don’t think the resurrected Commodore has a future in the U.S. PC market.

“History’s passed them by,” said Tim Bjarin, who runs Silicon Valley consultancy Creative Strategies and has monitored the PC market for more than 20 years. “It’s a blast from the past with no future. You’ve got HP and Dell, and Apple’s picking up tremendous steam. It’s basically pushing a boulder uphill.”

Commodore enthusiasts don’t deny the Sisyphean nature of their hobby. It’s not clear how many Commodore clubs there are, but they are scattered around the country, with devotees collecting and restoring the computers for old time’s sake.

“User groups are there for the camaraderie and friendship,” said Herd, who now runs a handful of small companies including an Internet service provider in New Jersey. “They remember the times of this really cool computer, but it’s more about the people.”

At Pizza Pit in Fresno, club members say they enjoy trading stories about keeping their machines running.

“Fifty percent of the time when we set things up, the hardware fails,” said Bill Terry, a former math teacher in Tulare. Interrupted by a group of children singing “Happy Birthday” at the next table, Terry said the school district paid for him to take computer lessons to upgrade his teaching credentials.

Hobbyist developers continue to build applications for Commodore machines. Most write games, but enthusiasts also have developed ways to read Windows spreadsheets on Commodore or access modern CD-ROM drives. The efforts recall the earliest days of computing, when most applications were home-built.

Jeri Ellsworth, a self-taught computer engineer in Portland, developed a joystick that sells for \$30 and contains about 30 old Commodore games. The joystick’s popularity lies with “all the 30-, 40-year-olds like myself who are trying to relive a little bit of their past,” Ellsworth said. “The toy is priced low enough that people don’t hesitate purchasing it just to play a few games they remember from their childhood.”

It’s a natural instinct: holding on to the past at a time of great change, said Cole, the Berkeley

professor. If not with Commodores, something else.

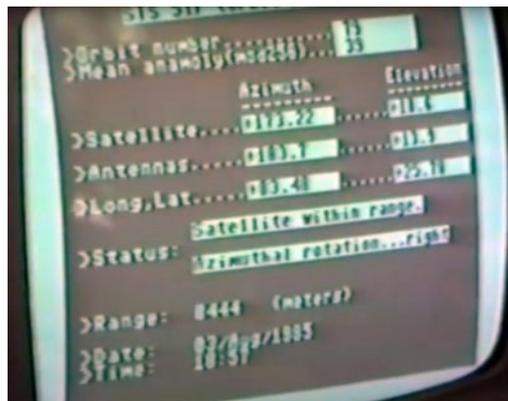
“I suspect the iPod at some point will perform that role,” he said. “Thirty years from now, old people still using old iPods – it’s absolutely plausible. People are so in love with their iPods that a lot of them will resist change.”

Jul. 02, 2006: For The Record

Los Angeles Times, Sunday July 02, 2006, Home Edition, Main News, Part A, Page 2, National Desk, 1 inches; 37 words. Type of Material: Correction

Commodore computers: A photo caption accompanying Tuesday’s Column One about enthusiasts of computers made by now-defunct Commodore Business Machines misidentified the photo’s subjects. Robert Bernardo was at left and Bill Terry was at right, not vice versa.

(Robert Bernardo writes, “Except for the error that said I was a high school teacher (I was really a middle school teacher), the article was accurate at the time. In the 14 years since the article was published, some things have changed; the dozens of clubs are now down to 7-8 Commodore and Amiga clubs, FCUG no longer meets at the Pizza Pit, Commodore International is gone, and sadly Jack Tramiel has passed away. However, some things have stayed the same; I still go to Ethel, WA to have Commodores repaired, I still live in Visalia and drive the white Crown Victoria, I still wear the ‘I Adore My 64’ button at times, and I have had more Commodore items autographed by William Shatner.)



Tracking Satellites with a Commodore PET

by Guest Contributor Chris Lott

A recent writeup by Tom Nardi about using the 6502-based NES to track satellites brought back memories of my senior project at Georgia Tech back in the early 80s. At our club station W4AQL, I had become interested in Amateur Radio satellites. It was quite a thrill to hear your signal returning from space, adjusting for Doppler as it speeds overhead, keeping the antennas pointed, all while carrying on a brief conversation with other Earth stations or copying spacecraft telemetry, usually in Morse code.

The Manual Way

Pulling this off was quite a production in the pre-Internet days. To begin with, you needed data about each satellite’s orbit. This was obtained by signing up to a real-paper mailing list from NASA’s Goddard Space Flight Center. An envelope arrived about twice a week by mail containing a stack of NASA Prediction Bulletins for the satellites you had requested. These contained the orbital parameters in what’s known as two-line element (TLE) set format, which you had to then type into your computer for processing. When we communicated with Space Shuttles, we would get the preliminary and later

actual TLE data from someone within NASA by phone or FAX.

Unlike geostationary communications satellites, Amateur radio satellites are typically launched in polar orbits which take between 90 to 120 minutes to complete one revolution of the Earth. To communicate through one of these satellite, first we needed to know when a particular satellite would pass over Atlanta.

To solve this problem, I wrote a Fortran program on the campus mainframe computer, a CDC Cyber 74, a real computational beast of the day. This program propagated the state vector given by the TLE data, and generated a list of orbits which passed over Atlanta. These passes were characterized by the maximum elevation, the duration of the pass, the AOS and LOS (Acquisition / Loss of Signal) times, and the distance at closest approach. We humans would filter this list even further, ignoring passes that were very short or that occurred when we would be asleep or attending class — after all, the reason we were at GA Tech was to study hard, learn a lot, and graduate with a degree.

Object - Orbital altitude – Inclination - Orbital period
 Oscar 8 - 900 km circular - 99 degrees - 103 min.
 Radio Sputnik 5,6,7,8 - 1600 km cir. - 83 deg. - 120 min.
 Shuttle STS-9 - 250 km circular - 57 degrees - 90 min.
 Shuttle STS-51F - 315 km circular - 50 deg. - 91 min.

The satellites of the day had two popular modes. Mode A was easiest to use for most Hams, the uplink was in the 2 m band, and the downlink was on 10 m. Mode J also used a 2 m uplink, but the downlink was in the 70 cm band. A typical Ham satellite station would use a stationary horizontal loop antenna for 10 m reception. The 2 m and 70 cm antennas were usually Yagis, mounted on

common rotatable elevation boom which was itself mounted on a vertical mast that could be rotated in azimuth. Ten meter reception was simple, no antenna movement required. Steering the Yagi antennas, however, was quite a task.

Having selected a suitable orbit, next we had make those Yagis follow the path of the satellite. In Amateur satellite stations, this is a completely open-loop process. You calculate where the satellite will be at a certain time, and drive the antennas to that point in space. If you remember the old home steerable TV antennas, before cable TV and streaming, the arrangement in a Ham shack is very similar. There is a rotator control box for each rotor, with an analog meter indicating direction, modern ones being digital, of course. These control boxes usually had three buttons for steering: Left, Right, and Brake release.

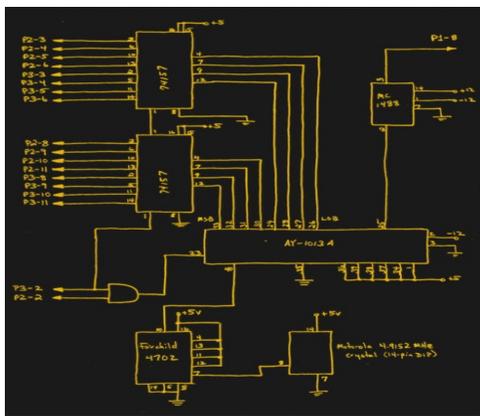
To follow the satellite, we had a couple of techniques available. One method used a graphical overlay of the satellite's ground track and footprint on top of a polar projection map of the Earth. By rotating the ground track overlay to match the orbit's position as given by the orbital parameter with the coolest name, Right Ascension of the Ascending Node, we could visually see the satellite's path over our station. The ground path overlay has time tick marks, and the footprint overlay has azimuth lines and elevation rings. This crude graphical technique provides the basic information needed to track the satellite, within the broad beamwidth of our antennas. We also had to keep our clocks synchronized to WWV or WWVH — no Network Time Protocol back then.

A second technique was more precise, but required a trip across campus. Using another Fortran program, I could print a table of time, azimuth and elevation angles for a desired orbit. The operator wouldn't have to interpolate angles from the graphical overlays: the numbers are clearly shown on the computer printout paper.

Getting this data from the campus computer wasn't trivial, however. Usually I would visit the computer center building across campus, login to my account at a terminal, and type in the data by hand. Upon execution, the program would generate a printout.

Then I had to wait at the output counter for the operator to separate and distribute the printouts to all of us impatient users. We felt so lucky when the club obtained a portable printing terminal with a built-in acoustic modem. This meant we could dial into the mainframe and run the programs remotely, and get our orbit schedules and tracking tables immediately without crossing campus — at the blazingly fast speed of 300 baud!

While all these machinations were certainly within our grasp, it took quite a bit of effort and advance planning. And while a lone operator could communicate with a satellite, two people were better — one to operate the radios, the other to steer the antennas. I recall two events from this period which led me to consider a computer controlled project. First of all, we inherited two Commodore PET computers from one of the EE professors. Secondly, we had an ongoing issue controlling equipment up on the roof of the 5th floor from our shack located at basement level. There just weren't enough wires to do all the things we wanted even for normal HF operations, much less satellite tracking.



Commodore PET to the Rescue

The Commodore PET computers we had were modified from their standard factory condition. They had full-size external keyboards, so you didn't have to use the Chiclet keys. The internal memory had been upgraded to 32K RAM. My report from 1983 says this expansion memory card also had two general purpose 8-bit TTL output registers, which were used to talk to the outside world.

As shown in this diagram, the PET computer talks to two devices — the Rotor Control Interface selects and operates the rotor boxes, and the Digital Serial Transmitter, essentially just a UART that sent data serially to the roof to remotely control a variety of equipment. Rotor control was fun, because we only had one rotor control cable between the shack the the roof. I had to implement 8P3T switches out of relays and multiplexed the cable between rotors. This means that only one rotor could rotate at a time, but this wasn't a severe constraint for our purpose.

Besides the rotator control signal multiplexing, there was quite a bit of auxiliary equipment on the roof which had to be remotely controlled. As shown in the diagram, we had RF coaxial relays to switch VHF/UHF antenna polarization, and to switch between 10 m and 70 cm downlink signal paths. In addition, there were preamps, down-converters, and power amplifiers which had to be turned on and off. Except for rotor multiplexing, most of these were static setups for the duration of a satellite pass, although the PET had full control of everything over this interface. We made a UART interface connected to one of the PET's TTL output registers to send this control data serially to the roof for control. I hand-drew all the schematics and hand-taped the PCB artwork – no KiCad back then.

I very quickly identified a problem having the PET control of all this rooftop gear, which was

not related to satellite operations at all. The PET wasn't running 24/7, and it was slow to boot and load up from cassette tape. Operators who just needed to control relays on the roof for HF operations didn't want to wait 10+ minutes to boot the PET. In fact they were quite vocal about it. So I added a manual control panel to bypass the PET. This philosophy has served me well over the years — computer control is great, but there are times when you need manual control, if only for testing.

We Don't Need Closed Loops

There was still one problem with this setup — you might have noticed there is no feedback. There was a method for the PET to read signals, but despite trying a variety of methods we couldn't get a stable enough feedback within our limited budget and limited mechanical skills. The azimuth rotor, which had an integrated feedback signal, would have been the easiest to monitor. But the elevation rotor we used didn't have any such feedback.

We tried a long rod with a lead weight at the bottom, attached to a potentiometer. When the boom rotated up and down in elevation, the rod always pointed down. Good idea on paper, didn't work so well in practice. We briefly played with synchomotors, which were really cool. Turn the shaft of one motor and the other motor across the room also turned the same amount, connected only by a few wires. But even these few wires were too many for our peculiar installation.

In the end, I concluded that for the satellites of interest and the antenna beamwidths involved, running the antenna rotation controls open loop was workable. To be sure, it caused a lot of headaches to procedurally check alignment before a pass, but once the pass began it worked as desired. I was continually tweaking the computer model of the antenna rotation, both the mathematical model and the constants such as

dead-time, rotation speed, acceleration, etc. It was a crude solution, but it got the job done within our limited budget.

Aside: Zenith crossing and three axis mounts

You might think that a satellite passing directly overhead would be perfect — it is as close as possible to your station. Well, that's not necessarily good. The problem is that when the satellite passes overhead, and the elevation angle approaches 90 degrees, the azimuth rotor will suddenly have to spin at an incredibly fast rate to keep pointed. There are ways to mitigate this — one method is to accept some pointing error and steer the antenna in a small circle around zenith — a circle whose diameter is determined by the speed of the satellite and the maximum speed of your rotor.

A more elegant solution, but mechanically complex, is to add a third axis of rotation. Two of the rotors are positioned before the satellite pass so that the antenna boom connected to the third rotor is perpendicular to the orbital plane. This way, only the third rotor is used during the satellite pass, and the other two rotors remain fixed.

Where is OSCAR?

Providing the PET with interfaces to the real world was only half the battle. I still needed a way to calculate the satellite's position. And while it was great to have a computer in the shack, the PET computer was thousands of times less powerful than the campus mainframe. There was no GUI, no tracking map of the Earth, just updating data fields on a text-based status screen.

I remember one major issue in the software development, however. There were several programs floating around the community those

days for tracking satellites, and I hand typed one of the popular ones into the PET. I contacted the author of this program to let him know about my school project and get his permission.

But to my surprise, he replied “No, you can’t use my software”. Just a minor setback for a young college student who doesn’t know his limitations — I went to the GA Tech library, checked out a couple of books on orbital mechanics, and wrote my own algorithms. Surprisingly, the fundamental Keplerian laws and equations for orbital bodies aren’t that difficult to understand and calculate. Alas, when you solve for the antenna pointing angles to the satellite, it results in a transcendental equation. Solving that on the mainframe took milliseconds, but it painfully slow on the PET. Over a month or so that followed, I confined myself to programming at home all weekend, developed an life-long addiction to caffeine-laden diet colas, and emerged with a working program.

Upgrades

The manual controls for rooftop equipment were non-intuitive. You were literally toggling address and data bits into a UART. This could have been improved, of course, but a few of us began to question the original “no more wires can be routed to the roof” constraint. It turns out that it was indeed possible, with a bit of effort. A larger cable for signaling was pulled to the roof, and an improved interface panel was made. Furthermore, the PET was showing its age, too, so we upgraded to the Commodore C64 — with floppy disk storage. I made a new, smaller rotor control interface that connected to the C64’s user port and controlled the rotors by relays.

Successful Ops

The system, while a bit finicky, did serve us well for a few years. Not only were we able to communicate over Amateur Radio satellites, but quite proudly we contacted the first two

astronauts allowed to operate from the Space Shuttle under a new program called SAREX. In Nov 1983 Dr. Owen Garriott (W5LFL) heard us on Columbia’s STS-9 mission, which was piloted by our own GA Tech alumni Dr John Young. We were also proud of our own recent club alumni, Jim Worsham, W4KXY, who was on the team at Motorola that designed and built the custom radio used by Dr. Garriott.

Again in Aug 1985, we contacted Dr Tony England (W0ORE) on Challenger’s STS-51F Spacelab mission. Videos were made of these efforts, which you can see in the links below.

Receiving SSTV from STS-51F

<https://www.youtube.com/watch?v=41y2ITIS8AE>

Attempted Contact with STS-51F

<https://www.youtube.com/watch?v=HtvRJ1HLFqI>

Successful Two-way Contact with STS-51F

<https://www.youtube.com/watch?v=gYGe8jiI-II>

Future?

I challenged myself last year to build a miniature version of this project that fits in an Altoids can. It’s on the back burner for now, until I can track down a copy of the original software. Stay tuned...



Wrapping Up

After couple of years after I graduated, the company I worked with generously let me manufacture and market this C64-based tracking system under the unimaginative name Satellite AutoTracker. That's a story for another day.

I can't overemphasize the impact this project had on me. The act of conversing with another radio station via a satellite pulls together so many engineering and science disciplines, from the physics of orbits and Doppler shifts to the minute details of learning to build PCBs to the overall system engineering efforts to make it all play together. Each one of these were interesting in and of themselves, but it was just plain awesome when it all successfully meshed together and worked.

I learned a lot building this project, but it didn't seem like learning because it was so fun. That, folks, is the best way to learn.

I should note this wasn't done overnight. I'd guess the whole thing stretched over a year or more, with many incremental steps and experiments along the way. About the PET itself, it's funny to say, given how underwhelming a C64 is compared to today's computers, but it was a huge step up in performance and ease-of-use when we migrated away from it to the C64. While the PET was more than adequate for hardware control and so-so for orbital calculations, it was too slow for any decent user interface. The programming experience, even with the full size keyboard, ranged from tolerable to downright unpleasant due to the small screen and cassette tape storage.

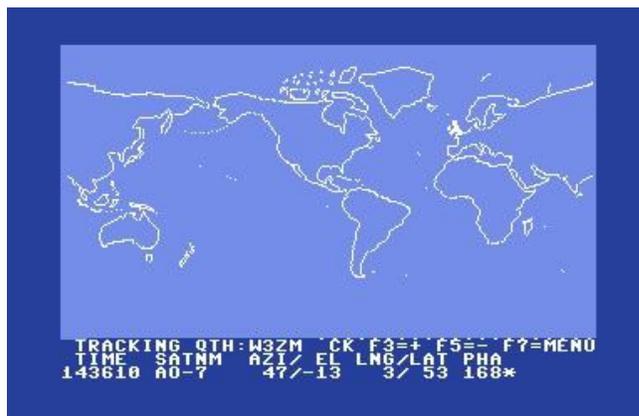
Paul Stoetzer N8HM wrote back to say:

Cool to see this. If anyone wants to try some retro computer satellite tracking, disk images for AMSAT's QUIKTRAK program for the

Commodore 64 and Apple II are available on AMSAT's website at

<https://www.amsat.org/flashback-friday-satellite-tracking-with-quicktrak-for-the-commodore-64-and-apple-ii/>

I also have a tape with the Timex Sinclair 1000 / Sinclair ZX-81 tracking program called AMS-81. I need to digitize that tape and get it posted.



Satellite Tracking with QUIKTRAK for C64/Apple II

by Guest Contributor Paul Stoetzer

<https://www.amsat.org/flashback-friday-satellite-tracking-with-quicktrak-for-the-commodore-64-and-apple-ii/>

For Flashback Friday, AMSAT is releasing disk images of QUIKTRAK for both the Commodore 64 and Apple II. These disk images should work in any emulator for the platforms, such as VICE for the Commodore 64 and AppleWin or LinApple for the Apple II. They can also be imaged to floppy disk or used with a floppy emulator on real vintage hardware. These disk images have AMSAT's club call, W3ZM, and Washington, DC set as the default call sign and QTH. Current Keplerian elements for AO-7 (as of September 13, 2019) are included. These

programs provide a fun look back at the state of the art in satellite tracking in the mid-1980s.

QUIKTRAK.DSK (Apple II)

QUIKTRAK.D64 (Commodore 64/128)

QUIKTRAK was written by Bob McGwier, N4HY, utilizing tracking routines originally developed by Tom Clark, W3IWI (now K3IO). While the Commodore 64 version is in compiled BASIC (Blitz), the AppleSoft BASIC source code for the Apple II version is included on the disk image and can be LISTed.

Additionally, Paul Williamson, KB5MU, recently made the InstantTrack program for the IBM PCs and PC compatibles freely available on his website.

If anyone has any copies of additional historic tracking programs, please email n8hm at amsat.org to arrange preserving this portion of AMSAT's history.

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 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.