



The

Broadcasters' Desktop Resource

www.theBDR.net

... edited by Barry Mishkind – the Eclectic Engineer

Bits & Bytes

Back to BASICS Can Be A Good Idea

By Barry Mishkind

[October 2021] There are many of choices of programming languages out there. BASIC (Beginners' All-purpose Symbolic Instruction Code) for any number of programs in the past. Today, it is now as well known, but still have uses, including on the new Raspberry Pi devices.

As computers began to be common in the 60s and 70s, it was obvious a general-purpose, high-level programming language whose design philosophy emphasized ease of use would help a lot of folks.

It was late 1950's when John Kemeny, math department chair at Dartmouth College, and Thomas Kurtz embarked on their effort to provide programming literacy for all students – even those not in the “normal” technical curricula. DARSIMCO (Dartmouth Simplified Code) and DOPE (Dartmouth Oversimplified Programming Experiment) were among the first attempts. These led to Fortran and ALGOL – usually noted as being overly complicated for the intended audience, which did not have the capability to deal with punch-cards and long run times, only to find out there were errors.

With a \$300,000 grant from the National Science Foundation, the first variations of BASIC smoothed out some commands from Fortran, and a free-to-use compiler, coupled with a GE-225 computer and Teletype Model 33, was released May 1, 1964. The goal was for BASIC to become widespread, and in that they were very successful.

Among the many variations were IBM's VS-BASIC (1973) and Micro-soft's MS-BASIC (1974). Micro-computers like the Commodore, Altair, and the TRS-80 were very popular on the consumer market.

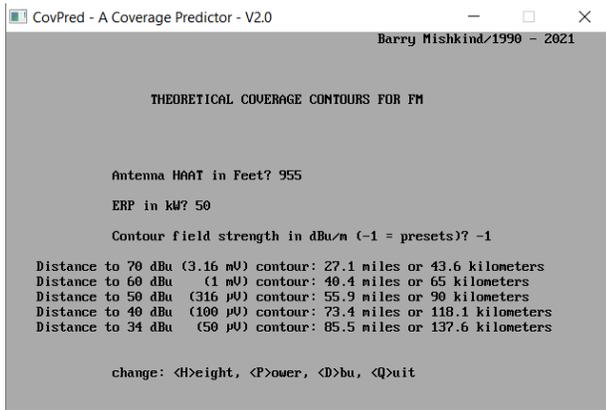
LEARNING BASIC

Since it was designed to take the machine code and let users instead use high-level language, people of all sorts started making programs.

BASIC as especially useful for the early machines which often featured a total of – are you ready? – 4 kB of memory. In other words, you had to be careful not to waste the available memory. Another minor impediment was that you stored the code on cassettes. (Yes, some of us are old enough to remember!)

I wrote a little program in 1978 for the purpose of taking the parameters of a directional antenna and comparing them to licensed values. Mostly mathematics, the little program saved a *lot* of time in analyzing the data from a six tower array. We fully used the 4 kB, and soon bought some more. With more memory, we were able to compile the programs with QBASIC, Power-BASIC, etc, so others could run them easily.

Then, I needed to determine distance and azimuth from two points, for STL or RPU issues, and wrote a program to do that. Estimating the coverage of an FM station was next and CovPred was born.



Due to the interest, I wrote an article about it and gave it away to anyone asking.

MORE AND MORE COMPLEX

It was not long before all sorts of folks were sharing little programs, with listing in all sorts of magazines.

There was even an automation program – in the late 70s! – that was essentially free for folks to adapt and use. Once there was an easier way to store programs using diskettes, more and more applications became available, in forms of free-ware, shareware, and paid software. Early word processors and even spreadsheets became popular.

In those days, it took a bit to build your own computer. Memory was as much as \$700 for 16 kB, and hard drives were “big” at 5 MB and “huge” at 10 MB. It was not hard to spend \$4,000 (or more – in 1970’s money) for a modest setup. One of the early hard drive setups for audio was shown at NAB; 60 MB for a cool \$60,000

But I digress. BASIC was the language of choice for everything from technical to financial to inventory applications – yes, and games, including one based on *Star Trek* by Mike Mayfield. By 1973 you could buy a book (!) of listings for *101 BASIC Computer Games*.

AUTOMATION AND MORE

Moving through the 1980s, the IBM PC and the Apple II became wildly popular and real computers were now in a price range pretty much anyone could afford.

Stations now had computers in several departments, and more sophisticated automation programs started to appear. Transmitter remote control and logging became possible.

Once the big companies found the market, large programs with more and more features were produced, first in BASIC, but then in other languages, including C, C+, and others. Users were just buying Word, Excel (or 1-2-3), and other applications to get the job done, happy they did not have to program. For those still wanting to program from the ground up, Visual Basic and similar programs allowed more easily produced graphic screens.

BACK TO BASIC

So, why am I talking about BASIC again?

A few months ago I was looking over the programs I wrote back in the late 70’s and 80’s. Unfortunately, not all computer CPUs and Operating Systems maintained compatibility from the days of the Z80 and 8086 systems. In Windows10, I was unable to run CovPred, for example.

Looking around, I was made aware of QB64, a new version of BASIC with a modern compiler for 32 and 64-bit machines.



Dusting out some cobwebs from my personal memory banks, I was able to take the code and compile it – and it still works! That was the best part of BASIC for me, while it just was a bit “wordy” at times, you can follow the logic and see what the program is doing – and why. Compiled, it can be passed along to others, so they do not have to figure out how to “run” it or compile themselves.

If you would like a copy of CovPred, even after 30 years, [I offer it to you freely \(click here to download Covpred.exe – it is safe!\)](#) It was based on the FCC algorithms of the time, which really have not changed much, but is useful for general results over the HAAT (Height of Antenna over Average Terrain) in most situations. Please note that some “stores” like the Microsoft Store or anti-virus programs may try to tell you there is danger because you did not get it from them. The program is safe.

QB64

If you are interested in learning, relearning, or just playing a bit, QB64 is available for Windows, MacOS, or Linux. Fellippe Heitor and his team are responsive and do not laugh at us relearners.



Just go visit www.QB64.org where QB64 Version 2.0.1 was just released. Its latest feature is an included debug routine which allows you to cycle through a program one line at a time. Also available is a graphical-interface generator, called InForm.

They are free downloads. And there are several videos and several community groups available to answer questions and help you along. There is option to donate for those working to upgrade the program, but it is not demanded.

It is up to you what applications, small or large, will do something that you need to do. I already have an idea on something that will save me at least 6-8 hours a month on something repetitive.

Feel free to let us know what successes you have.

Are articles like this helpful to you? If so, you are invited to sign up for the one-time-a-week BDR Newsletter. It [takes only 30 seconds by clicking here.](#)

[Return to The BDR Menu](#)